Archives of Photogrammetry, Cartography and Remote Sensing, vol. 33, pp. 23-43
ISSN 2083-2214, eISSN 2391-9477
DOI: http://doi.org/10.14681/apcrs-2021-002

RESEARCH ARTICLE

USING CONVOLUTIONAL NEURAL NETWORKS TO IMPROVE THE SPATIAL RESOLUTION OF AERIAL IMAGES

WYKORZYSTANIE KONWOLUCYJNYCH SIECI NEURONOWYCH DO POPRAWY ROZDZIELCZOŚCI PRZESTRZENNEJ ZDJĘĆ LOTNICZYCH

Kamil Bartosik, Zdzisław Kurczyński

Warsaw University of Technology, Faculty of Geodesy and Cartography,
Department of Photogrammetry, Remote Sensing and Spatial Information Science

KEY WORDS: convolutional neural networks, high-resolution convolutional neural networks, image spatial resolution, image resolution enhancement

ABSTRACT: Artificial intelligence is rapidly finding increasing popularity and applications. In the field of photogrammetry and remote sensing, convolutional neural networks (CNNs) are applied, especially for object detection and classification in aerial and satellite images. Some variations of such networks are super-resolution convolutional neural networks (SRCNNs). Proposed only a few years ago, they are already finding applications for increasing the spatial resolution of aerial and satellite images. This problem is addressed in the paper. The authors introduce the issue of convolutional neural networks and, on this background, the specifics of super-resolution convolutional neural networks. In the experimental part, they undertook the task of improving the resolution of high-resolution aerial images acquired with a large-format camera for a city area with original resolution (ground sampling distance) GSD=5 cm. These images were degraded to varying degrees and then subjected to resolution enhancement with the help of a constructed network. The resulting (sharpened) images were evaluated quantitatively using defined measures of resolution improvement and visual comparison with the original images. The results, surprisingly good especially for images with twice the resolution degradation, confirm the great potential of convolutional neural networks to improve the spatial resolution of aerial images is confirmed.

1. INTRODUCTION

Creating an algorithm subject to mathematical rules that performs tasks more efficiently than humans is easy. The real challenge, however, is to create information systems that would perform simple tasks from the human point of view but are difficult to present through formalised mathematical rules. These tasks involve intuitive decision-making based on human consciousness and knowledge gained through experience and knowledge of how the world works.

Humanity has long looked toward creating a self-aware artificial intelligence (AI) to solve such indeterminate problems. Until a few years ago, artificial intelligence was

associated more with the imagination of science fiction writers than with science. Although there is still a long way to go to implement the vision of AI known from pop culture, the technology is already rapidly developing, and the first promising results can be observed. More and more tasks are already being automated and optimised using AI algorithms.

As in other fields, advanced IT tools are widely used to automate and optimise many processes in photogrammetry and remote sensing. Many research centres are looking for applications for this technology, where one of the leading research directions is object detection and classification in photogrammetric images, for which convolutional neural networks (CNNs) are used.

In 2015, a paper was published by Dong *et al.* (2015), Image Super-Resolution Using Deep Learning Convolutional Neural Networks, proposing a completely new use of CNNs. The proposed method obtained significantly better results than other known ways to increase image resolution. The newly created type of algorithm was called super-resolution convolutional neural networks (SRCNN). The publication of these results contributed to the rapid spread of this technology and the beginning of research work in many research centres around the world, which confirm the effectiveness of SRCNN.

The increasing resolution of Sentinel-2 satellite imagery has been reported in the literature (Galar *et al.*, 2019). Very high-resolution satellite image distributor Maxar has recently offered images with a resolution (ground sampling distance) of 15 cm, generated by upscaling images with an original resolution of 30 cm (https://maxar.com). Despite the great interest in such algorithms, the number of conducted studies using this solution in photogrammetry or remote sensing is still limited.

The authors set out to investigate the possibility of improving the spatial resolution of aerial images using a developed deep-learning algorithm. The algorithm takes an image with a given spatial resolution and returns an image with increased spatial resolution. This problem is of great practical importance. The spatial (or geometric) resolution of images determines their measurement and interpretation potential. However, an increase in resolution also means a significant increase in the acquisition cost and subsequent processing. Just a twofold increase of resolution, e.g. from GSD=10 cm to GSD=5 cm, means a fourfold increase in the number of images covering a given area, as well as a twofold increase in the cost of their acquisition (twice as long time of plane operation over the object). The cost of processing will also increase (four times as many images to be processed).

2. EVALUATION OF THE IMPROVEMENT OF IMAGE RESOLUTION

To assess the quality of resolution improvement, the most commonly used is the peak signal-to-noise ratio (PSNR). This ratio is determined as the ratio between the maximum possible pixel value (signal) and the mean squared error (MSE) determined from the difference between the pixel values of the original and transformed image (noise). The PSNR value is expressed in decibels (dB) (Galar *et al.*, 2019). The value of PSNR is given by the formula 1:

$$PSNR = 10 * \log_{10}\left(\frac{{v_{max}}^2}{MSE}\right) \quad (1)$$

where:

v_{max} - maximum possible pixel value (for 8-bit images v_{max} = 255),
MSE - the mean squared error.

The value of MSE is given by the formula:

$$MSE = \frac{1}{N*M*C}\sum_{x=1}^{N}\sum_{y=1}^{M}\sum_{k=1}^{C}[f(x,y,k) - f'(x,y,k)]^2 \quad (2)$$

where:

N, M, C - image dimensions (N - height, M - width, C - number of channels) [px],

f(x, y, k) - value of a pixel with coordinates (x, y, k) of the original image [px],

f '(x, y, k) - value of a pixel with coordinates (x, y, k) of the transformed image [px].

The second widely used index is the structural similarity index (SSIM) of the images, taking values from -1 to 1. When creating resolution enhancement algorithms, SSIM = 1 is aimed for, because it means that the images are identical. The index depends on three components: luminance, contrast and structure, which are given by the formulas (Wang *et al.*, 2003):

$$l(x,y) = \frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1} \quad (3)$$

$$c(x,y) = \frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_2} \quad (4)$$

$$s(x,y) = \frac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3} \quad (5)$$

where:

μ_x, μ_y - the mean value of x or y,

σ_x, σ_y - variance of x or y,

σ_{xy} - covariance of x and y,

$C_1 = (K_1*L)^2$, $C_2 = (K_2*L)^2$, $C_3 = C_2/2$,

K_1, K_2 - constant values much smaller than 1,

L - maximum possible pixel value (for 8-bit images L = 255).

SSIM is given by the formula:

$$SSIM(x,y) = [l(x,y)]^{\alpha} * [c(x,y)]^{\beta} * [s(x,y)]^{\gamma} \quad (6)$$

where:

α, β, γ - parameters that determine the weights of the component data. Most commonly, all are assumed to be equal to 1.

3. CONVOLUTIONAL NEURAL NETWORKS

Artificial intelligence (AI) is the ability of a computer, or a machine controlled by a computer, to perform intellectual activities typical of humans, such as reasoning, discovering meaning, reading information, generalising, or learning through experience. In such an understanding of artificial intelligence, one can separate the following subsets: machine learning, deep learning, computer vision, and natural language processing.

Convolutional neural networks (CNNs) are one of the most widely used deep learning methodologies. They are used to process images to detect and categorise objects in them. The operation of such an algorithm involves creating a training set on which the network can learn. The programmed CNN "learns" to read features from the images by matching them with appropriate objects to categorise them. The key is the amount of training data and its representativeness.

3.1. Construction of convolutional neural networks

The network is a set of many interconnected layers - Fig. 1. The first layer of the network reads pixel matrices of input images, performs calculations and appropriate transformations on them, and then passes the results to the next layer. Each subsequent layer reads the results of the previous layer and, like the first layer, elaborates on them and passes them to the next layer until the results of the entire network, the so-called output, are obtained. A CNN has several different types of layers and operations that serve a specific purpose.

The convolution operation consists of filtering the input images using a filter. In this way, output pixel values are obtained, which are called feature maps. As the name suggests, the result of convolution is localised characteristic features of objects in the image being filtered. In the case of colour images, the filter passes through each RGB channel separately, and then all 3 results are summed together to create only one feature map for the input image.

The objects are at different locations in the images, are at various angles, and may be illuminated differently. For a convolutional neural network to correctly detect and categorise objects, it must be insensitive to this problem, i.e., it must have spatial invariance. A pooling operation is used to ensure this, which works very similarly to convolution. The results of these computations form an updated feature map, which serves as input to the next layer.

The final piece of CNN is the artificial neural network (ANN). Such networks take input in the form of single numerical values. A convolutional network, on the other hand, operates on images, so the input and output of each layer are pixels stored in an array fashion. To combine these two networks, we use a flattening operation. It transforms the matrix data into a single long vector.

During training, the components of the layers are assigned weights that reflect their importance in obtaining the desired result. These weights are arguments supplied to the activation function to check their relevance to the solution. If the weight exceeds the activation threshold, the layer component is considered important for the network training process and is used in its further part. Otherwise, the component is deactivated and thus no

longer involved in the training. The most commonly used activation functions in convolutional neural networks are ReLU, sigmoid, and softmax.

3.2. The network architecture and its training process

Convolutional neural networks are built from two significant parts. Fig. 1. The first is for feature extraction from input images built from convolution layers. Results of each layer (feature maps) are passed through the ReLU activation function and then subjected to pooling operation. Final feature maps are flattened, creating a flatten layer. This layer is the input to the second part of CNN, i.e., the artificial neural network (ANN) responsible for classifying objects in the input image. The network results are passed through a sigmoid or softmax activation function depending on the number of possible outcomes. These functions have the task of determining the probability with which an object is categorised.

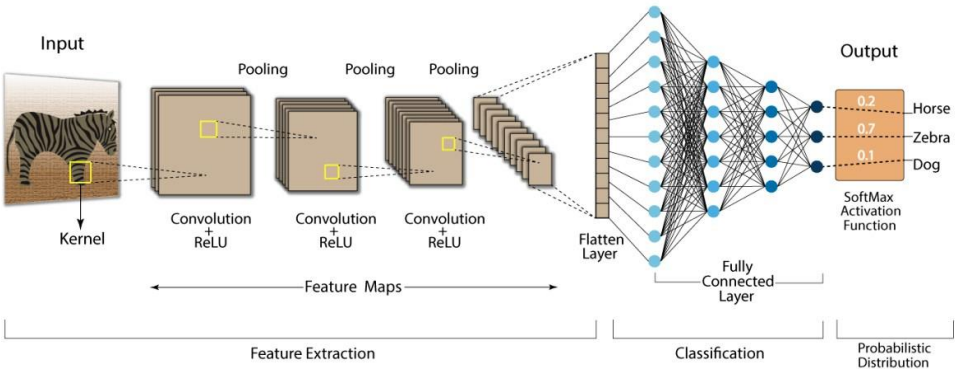

Figure 1: Architecture diagram of an example convolutional neural network (source: developersbreach.com).

A CNN cannot initially perform the tasks for which it is designed. The network learns from the data it receives and, thus, from a prepared training set. The training data should be large and diverse for the network to achieve the highest possible accuracy. The backpropagation process learns the network, which is divided into 4 operations: forward pass, loss function, backward pass, and weight update. During forward pass, the training data is passed through the entire network according to its direction, producing reference results. Then, a cost function is computed to determine the error between the obtained and expected results. In the next step, a backward pass operation is performed, so calculations are performed in the opposite direction of the network. In this way, the gradients between adjacent layers are calculated. The last step of the backpropagation process is to update the weights.

The backpropagation process is performed iteratively. The training data is divided into smaller sets of images (batches). When all the images are used, the process is repeated, and one such loop is called an epoch. CNN training is often time-consuming, as it processes

massive data in several hundred epochs. If it takes too long, it runs the risk of learning dependencies that are not true. This results in excessive matching of the obtained weights to the training data - the so-called network overfitting. To prevent this problem from occurring, the accuracy of the network should be evaluated on an independent set of validation data during training. An optimal CNN has very high precision on both datasets, keeping the accuracy difference between them as small as possible. This indicates that the network can operate correctly on data other than what was learned.

3.3. Super-resolution convolutional neural networks

Super-resolution convolutional neural networks (SRCNNs) solve the task of increasing spatial resolution by comparing small portions of images (patches). High-resolution patches are paired with their lower resolution scaled counterparts. The created patches form a training set divided into subsets of input patches and target patches. CNNs are trained to obtain best-fit weights that determine the accuracy of the network. In contrast, filters are the most important in SRCNNs, as they are responsible for the change in resolution between the input and output images. SRCNNs are trained to get the most accurate filters, not the overall accuracy of the network, which does not play an important role in this case. For this reason, the validation dataset is also not used. SRCNNs have convolution layers, and the activation function ReLU is applied after each layer. The pooling operation is not applicable in high-resolution networks, as it significantly reduces the image and is used to obtain spatial invariance, which is irrelevant to the image-rescaling process. Similarly, the subsequent operations that are responsible for classification are not used. This makes SRCNNs simpler to build than CNNs, containing only convolution layers and ReLU activation functions (Rosebrock, 2017).

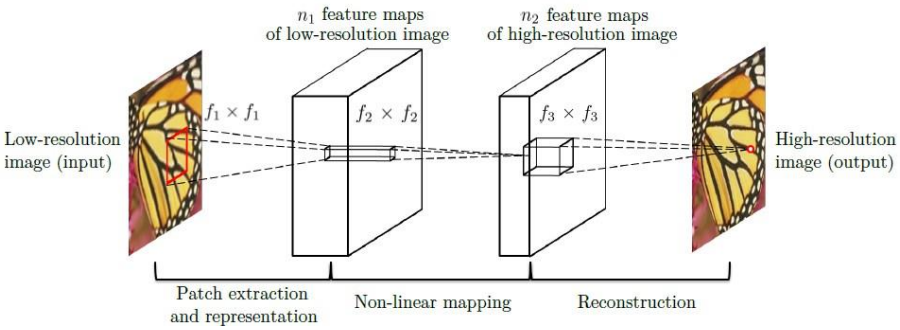

Fig. 2 Schematic of the super-resolution construction of convolutional neural networks (source: Dong et al., 2015).

Training CNNs that detect and categorise objects in images requires hundreds and sometimes thousands of iterations (epochs). In the case of SRCNNs, this is 10 to 12 epochs. A more significant number of iterations adversely affects the performance of super-resolution networks (Dong *et al.*, 2015).

4. EXPERIMENT

4.1. Initial data

The experiment required creating a deep learning algorithm in the Python programming language. The language has many libraries. The libraries used were: OS, random, regular expression (re), time, Numerical Python (numpy), h5py, OpenCV-Python (cv2), Matplotlib, TensorFlow, Keras, and Scikit-image (scimage). OpenCV-Python solves computer vision problems by reading images as pixel arrays or saving such data as images. The TensorFlow and Keras libraries run in parallel, providing tools for working with deep learning problems, such as ready-made convolutional neural network layers.

The data used in the experiment are aerial images from a flight mission conducted over the city of Lodz in April 2015. The photos with GSD=5 cm were acquired using a digital photogrammetric camera. 50 such images with dimensions of 17310x11310px were used. The data were divided into a training set and a test set. The training set was assigned 45 images, which were used to train the SRCNN in later stages. The remaining 5 images found their place in the test set. As independent data, they were used to analyse trained network results consisting of visual evaluation of transformation and numerical verification of its accuracy with PSNR and SSIM indices.

The large format of the images (17310x11310 px) provides a huge amount of data that was difficult to process on the computer used in the experiment. To solve this problem, an algorithm was created that randomly cuts out 600x450px fragments, treated as new smaller images. It is possible to cut out about 725 unique fragments (600x450px) from each original photo. The SRCNN was trained on 5 different datasets, differing in the number of photo fragments: 315 (7*45), 450 (10*45), 1350 (30*45), 2700 (60*45) and 4500 (100*45). A maximum of almost 1/7 of the original data was used.

In the next step, lower resolution (degraded) images had to be acquired, which the algorithm will compare with the original images. For this purpose, the training set is divided into two subsets. The first is the subset of training target images (training target set), which consists of previously acquired small fragments of large format images. The second subset is the training input set, created by degrading the images belonging to the first subset. The degradation was performed by bicubic interpolation. In this way two sets were prepared, forming pairs of original and degraded photos between each other. The SRCNN algorithm was used in the experiment mainly to improve the resolution of the degraded images twice to GSD=10 cm. However, the tests were also performed for resolution changes 3, 4 and 6 times.

4.2. Construction of the designed network and its training process

The designed SRCNN has 3 convolution layers. Each layer goes through a ReLU (Rectified Linear Unit) activation function. The cost function used is the mean squared error, which allows the performance of the algorithm to be compared on the training and test sets

by calculating the PSNR index. The network learned on 4500 images achieved the best results from all the training performed. Networks trained on sets of 1350 and 2700 images performed slightly worse, confirming the logarithmic relationship between the amount of training data and the training accuracies found in each CNN type (Fig. 3).

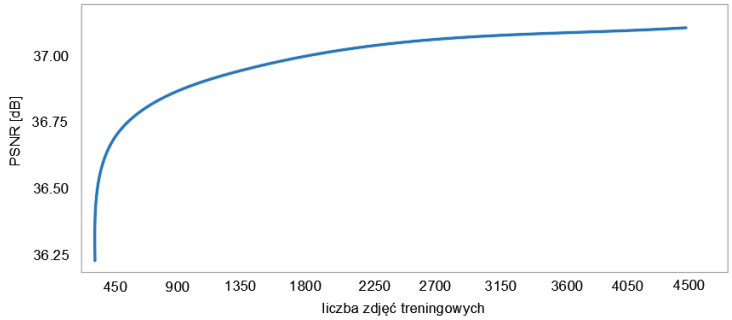

Fig. 3. Graph showing training accuracy (PSNR index) as a function of training set size (Bartosik, 2020).

This means a much more extensive training set must be used to achieve better results marginally. The 4500 image set used was associated with the generation of 5.535 million patch pairs, where each patch saved in HDF5 format occupied 6 KB of computer memory, which was related to the generation of a total of over 63 GB of training data. Loading such data was very demanding on the processor, and including network training (the training itself was already performed on the GPU), the whole process took almost 8 hours.

4.3. The analysis of results

Networks trained on sets of 1350 and 2700 images achieved training results similar to the network trained on 4500 images. Most of the training was performed on a sample of 450 training images. The remaining network training was performed only for a twofold improvement in resolution. However, they produced significantly more artefacts (noise) after examining them on the test set. The numerical analysis of the results by PSNR and SSIM indices is insufficient and should be complemented by a visual inspection of the results. The accuracy of the algorithm performance for all rescaling with PSNR and SSIM indices was analysed. The results are presented in Table 1.

Tab. 1 Resolution improvement indices of PSNR and SSIM images for each scale. All values are averaged over the affected image sets.

Scale	**PSNR [dB] input**	**SSIM input**	**PSNR [dB] result**	**SSIM result**
2	34.41	0.8151	36.43	0.8439
3	30.45	0.5972	32.02	0.6307
4	28.61	0.4997	30.65	0.5100
6	26.30	0.3213	28.72	0.3313

The obtained indices show very good performance of the SRCNN network for dual scaling of spatial resolution of aerial images. For the other scales the indices are not so high, however, one should remember that they are the result of training the network on only 10% of the data used for training the scale equal to 2.

The best overview for the network performance is given by visual confrontation of corresponding fragments of the images: original, input (degraded) and resultant. Such examples are illustrated in figures 4 and 5, each presenting a series of enlargements of photo fragments with varying degrees of rescaling. In the figure captions, the corresponding quantitative PSNR and SSIM indices are additionally given, allowing us to relate the visual effect to quantitative quality measures.

The images scaled 2 times achieved relatively better results than the rest due to using a 10 times more extensive training set. The results are still good for the remaining rescaling, raising the value of the input image indices noticeably. The performed observations of the resulting images also confirm this.

Even though data used in the experiment came from a photogrammetric flight mission over the city, land cover in the images was diverse. It included not only urban areas but also green areas. It was decided to complement the analysis of the results by cutting out parts of the test images so that homogeneous land cover prevailed (Table 2).).

Tab. 2. Resolution improvement indices of images with land cover distinction. 15 images from the test set of 600x450px were selected for analysis. Resolution improvement scale equal to 2.

	Green areas		**Urban area**	
	PSNR [dB]	**SSIM**	**PSNR [dB]**	**SSIM**
Maximum	37.58	0.8906	40.02	0.8127
Average	36.45	0.8703	36.36	0.8027
Minimum	35.78	0.8586	32.90	0.7953

The results of the study presented in Table 2 show interesting relationships. The results were further supported by visual analysis. The training data contained a predominance of urbanized areas, resulting in smaller fluctuations in the SSIM index and a much higher maximum PSNR score than green areas. Green areas represented by less data, are more uncertain, as demonstrated by large fluctuations in SSIM index and much lower maximum PSNR. The average PSNR index values are similar for both land types; however, this is due to the greater amount of artifacts present in urbanized areas, effectively understating the average index score. Despite the smaller amount of data for vegetation-covered areas, they achieved better mapping fidelity than urban areas. The presence of heterogeneous land cover in the aerial images used may have affected the performance of the overall algorithm.

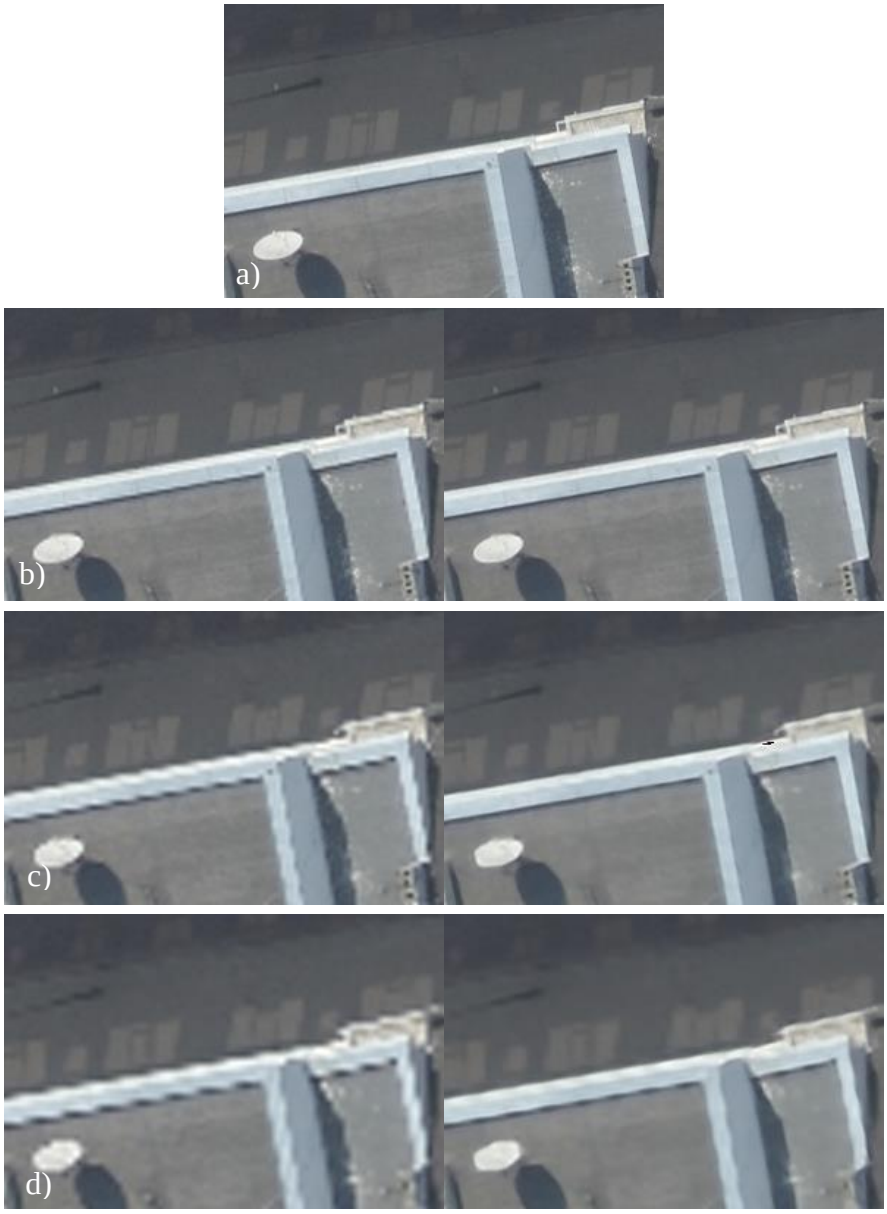

Fig. 4. a) original image GSD=5cm,
b) 2-fold rescaling (GSD=10cm), from left: input, degraded image (indices: PSNR=31.33 dB, SSIM=0.6633), resultant image (indices: PSNR=39.66 dB, SSIM=0.8588),
c) 3-fold rescaling (GSD=15cm), from left: input, degraded image (indices: PSNR=31.33 dB, SSIM=0.6633), resultant image (indices: PSNR=32.90 dB, SSIM=0.6694),
d) 4-fold rescaling (GSD=20cm), from left: input, degraded image (indices: PSNR=29.16 dB, SSIM=0.5559), resultant image (indices: PSNR=31.89 dB, SSIM=0.5754) (source: Bartosik, 2020).

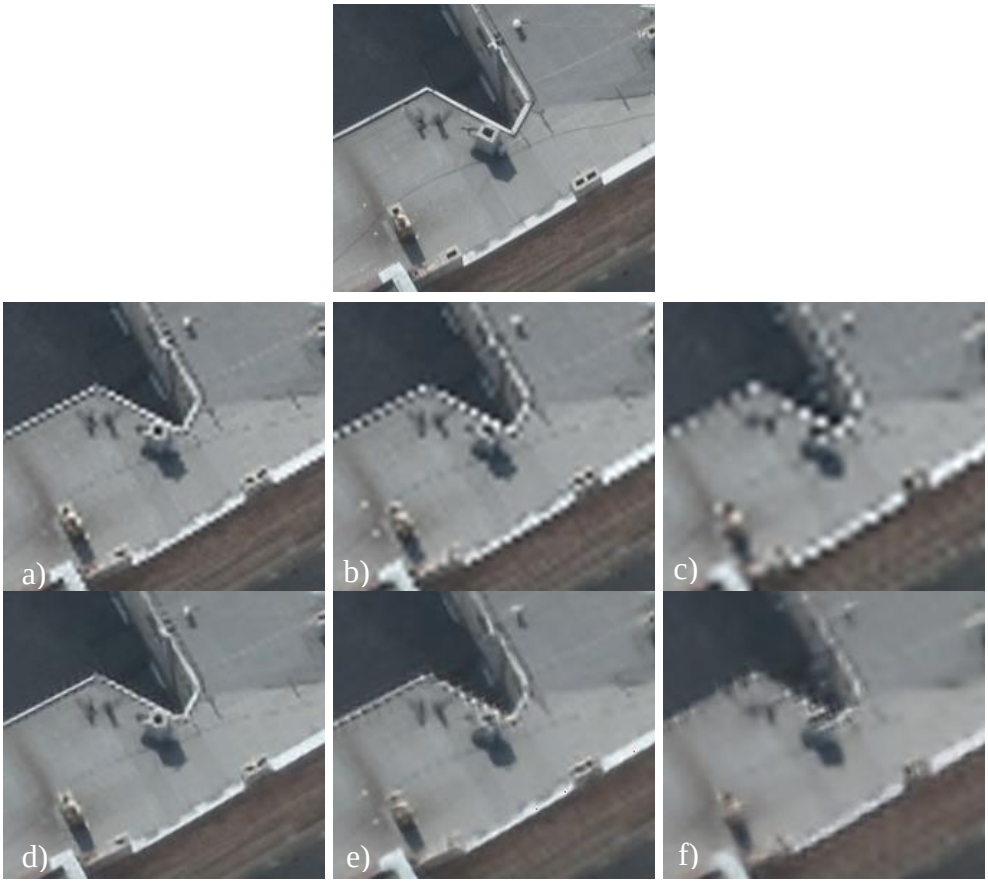

Fig. 5. Top: original image GSD=5cm.
Middle: input images (degraded), (a) GSD=15cm, indices: PSNR=29.43 dB, SSIM=0.6968, (b) GSD=20cm, indices: PSNR=26.76 dB, SSIM=0.5743, c) GSD=30cm, indices: PSNR=24.31 dB, SSIM=0.3789.
Bottom: resultant images: d) 3 times rescaling, indices: PSNR=31.13 dB, SSIM=0.6997, (e) 4 times rescaling, indices: PSNR=28.00 dB, SSIM=0.5766, f) 6 times rescaling, indices: PSNR=25.16 dB, SSIM=0.3940 (source: Bartosik, 2020).

5. SUMMARY AND CONCLUSIONS

The numerical indices (PSNR and SSIM) and the observation of the outcomes allow us to conclude that the results achieved by the created SRCNN are very good for all examined rescaling of the spatial resolution. Thus, the effectiveness of the deep learning algorithm on photogrammetric data is confirmed. This technology can be used to create photogrammetric products of better quality and for object detection and classification in aerial images. Combining the algorithm developed in this way with a typical CNN network that detects and categorises objects in images can also increase its effectiveness. The experiment results

confirm that SRCNNs are a technology that can be used to automate and optimise many photogrammetric and remote sensing tasks, taking out the tedious work of humans.

The results presented in this paper illustrate the effectiveness of SRCNN. However, effective use of such algorithms requires using computers with high computing power. This would also allow the network to be trained on larger data sets.

Since the SRCNN achieves different results for the urban and green areas, a supporting algorithm could be additionally developed using standard CNNs. Such an algorithm would be taught to distinguish and sort image data fragments by land cover types and would treat them separately by cutting patches and feeding them into two different SRCNNs. The results of the two networks would be combined again into a single image with even more effective resolution improvement.

LITERATURE

Bartosik K. (2020). Wykorzystanie konwolucyjnych sieci neuronowych do poprawy rozdzielczości przestrzennej zdjęć lotniczych. Praca dyplomowa inżynierska, Zakład Fotogrametrii, Teledetekcji i Systemów Informacji Przestrzennej, Wydział Geodezji i Kartografii, Politechnika Warszawska.

Bielecki M. (2018). Wykrywanie wybranych klas obiektów na danych lotniczych z wykorzystaniem konwolucyjnych sieci neuronowych. Praca dyplomowa, Zakład Fotogrametrii, Teledetekcji i Systemów Informacji Przestrzennej, Wydział Geodezji i Kartografii, Politechnika Warszawska.

Dong C., Change Loy C., He K., & Tang X. (2015). Image Super-Resolution Using Deep Convolutional Networks, arXiv:1501.00092v3.

Galar, M., Sesma, R., Ayala, C., & Aranda, C. (2019). Super-resolution for Sentinel-2 images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, 95-102.

Rosebrock A. (2017). Deep Learning for Computer Vision. Pyimagesearch.

Wang Z., Simoncelli E.P., & Bovik A.C. (2003). Multi-Scale Structural Similarity for image quality assessment. 37th IEEE Asilomar Conference on Signals, Systems and Computers, 2, 1398-1402

WYKORZYSTANIE KONWOLUCYJNYCH SIECI NEURONOWYCH DO POPRAWY ROZDZIELCZOŚCI PRZESTRZENNEJ ZDJĘĆ LOTNICZYCH

SŁOWA KLUCZOWE: konwolucyjne sieci neuronowe, superrozdzielcze konwolucyjne sieci neuronowe, rozdzielczość przestrzenna zdjęć, poprawa rozdzielczości zdjęć

STRESZCZENIE: Sztuczna inteligencja szybko znajduje coraz szersze zainteresowanie i zastosowania. W zakresie fotogrametrii i teledetekcji zastosowanie znajdują konwolucyjne sieci neuronowe (CNN), szczególnie do detekcji i klasyfikacji obiektów na zdjęciach lotniczych i satelitarnych. Pewną odmianą takich sieci są głębokie konwolucyjne sieci neuronowe (super-resolution convolutional neural networks - SRCNN). Zaproponowane zaledwie parę lat temu, już znajdują zastosowanie do zwiększania rozdzielczości przestrzennej zdjęć lotniczych i obrazów satelitarnych. Tego problemu dotyczy artykuł. Autorzy przybliżają problematykę konwolucyjnych sieci neuronowych, a na tym tle specyfikę superrozdzielczych konwolucyjnych sieci neuronowych. W części eksperymentalnej podjęli zadanie poprawy rozdzielczości wysokorozdzielczych zdjęć lotniczych pozyskanych wielkoformatową kamerą dla obszaru miasta o oryginalnej rozdzielczości (piksel terenowy) GSD=5 cm. Zdjęcia te zostały zdegradowane w różnym stopniu, a następnie, z pomocą zbudowanej sieci poddane poprawie rozdzielczości. Wynikowe (wyostrzone) zdjęcia zostały poddane ocenie ilościowej z wykorzystaniem zdefiniowanych miar poprawy rozdzielczości oraz wizualnie, poprzez porównanie ze zdjęciami oryginalnymi. Zaskakująco dobre, szczególnie dla zdjęć o dwukrotnej degradacji rozdzielczości, potwierdzają duży potencjał konwolucyjnych sieci neuronowych do poprawy rozdzielczości przestrzennej zdjęć lotniczych.

1. WSTĘP

Łatwo można stworzyć algorytm podlegający regułom matematycznym wykonujący powierzone mu zadania w sposób bardziej efektywny od człowieka. Prawdziwym wyzwaniem jest natomiast stworzenie systemów informatycznych, które wykonywałyby proste zadania z punktu widzenia człowieka, jednakże trudne do przedstawienia w sposób sformalizowanych reguł matematycznych. Są to zadania polegające na podejmowaniu intuicyjnych decyzji opartych na ludzkiej świadomości oraz wiedzy zdobytej przez doświadczenie i znajomość funkcjonowania świata.

Ludzkość już od dawna patrzyła w kierunku stworzenia samoświadomej sztucznej inteligencji (ang. *artificial intelligence*, AI) rozwiązującej takie niezdeterminowane problemy. Jeszcze kilka lat temu sztuczna inteligencja kojarzyła się bardziej z wytworem wyobraźni twórców fantastyki naukowej niż nauką. Choć jeszcze długa droga przed nami, aby urzeczywistnić wizję AI znaną z popkultury, to już dziś technologia ta jest prężnie rozwijania i można zaobserwować pierwsze obiecujące rezultaty jej działania. Coraz więcej zadań automatyzuje i optymalizuje się już za pomocą algorytmów AI.

W fotogrametrii i teledetekcji podobnie jak w innych dziedzinach szeroko wykorzystuje się zaawansowane narzędzia informatyczne, automatyzujące i optymalizujące wiele procesów. W wielu ośrodkach badawczych poszukuje się zastosowań dla tejże technologii, gdzie jednym z głównych kierunków badań jest detekcja i klasyfikacja obiektów na obrazach fotogrametrycznych, do czego wykorzystuje się konwolucyjne sieci neuronowe (ang. *convolutional neural networks*, CNN).

W 2015 roku została opublikowana praca Dong *et al.* (2015). *Image Super-Resolution Using Deep Learning Convolutional Neural Networks* proponująca całkowicie nowe wykorzystanie CNN. Zaproponowana metoda uzyskała znacząco lepsze wyniki niż inne znane sposoby zwiększania rozdzielczości obrazów. Nowo stworzony typ algorytmów nazwano superrozdzielczymi konwolucyjnymi sieciami neuronowymi (ang. *super-resolution convolutional neural networks*, SRCNN). Opublikowanie tych wyników przyczyniło się do szybkiego rozpowszechnienia się tej technologii i rozpoczęcia prac badawczych w wielu ośrodkach naukowych na świecie, które potwierdzają skuteczność SRCNN.

W literaturze pojawiły się doniesienia o zwiększaniu rozdzielczości obrazów satelitarnych systemu Sentinel-2 (Galar *et al.*, 2019). Dystrybutor obrazów satelitarnych o bardzo dużej rozdzielczości Maxar od niedawna oferuje obrazy o rozdzielczości (piksel terenowy) równej 15 cm, wygenerowane poprzez zwiększenie rozdzielczości obrazów o oryginalnej rozdzielczości równej 30 cm (https://maxar.com). Mimo dużego zainteresowania takimi algorytmami liczba przeprowadzonych badań korzystających z tego rozwiązania w fotogrametrii czy teledetekcji wciąż jest mała.

Autorzy postawili sobie za cel zbadanie możliwości poprawy rozdzielczości przestrzennej zdjęć lotniczych za pomocą stworzonego algorytmu głębokiego uczenia (ang. *deep learning*). Algorytm przyjmuje zdjęcie o danej rozdzielczości przestrzennej i zwraca zdjęcie ze zwiększoną rozdzielczością przestrzenną. Problem ten ma doniosłe znaczenie praktyczne. To od rozdzielczości przestrzennej (inaczej: geometrycznej) zdjęć zależy ich potencjał pomiarowy i interpretacyjny. Ale wzrost rozdzielczości oznacza również istotny wzrost kosztów pozyskania i późniejszego ich opracowania. Tylko dwukrotny wzrost rozdzielczości, np. z GSD=10 cm, do GSD=5 cm, oznacza aż czterokrotny wzrost liczby zdjęć pokrywających dany obszar, oraz dwukrotny wzrost kosztów ich pozyskania (dwukrotnie dłuższy czas operowania samolotu nad obiektem). Wzrasta również koszt opracowania (czterokrotnie więcej zdjęć do opracowania).

2. OCENA POPRAWY ROZDZIELCZOŚCI ZDJĘĆ

Dla oceny jakości poprawy rozdzielczości zdjęć, najczęściej stosuje się wskaźnik szczytowego stosunku sygnału do szumu (ang. *peak signal-to-noise ratio*, PSNR). Wskaźnik ten wyznacza się, jako stosunek między maksymalną możliwą wartością piksela (sygnału) a błędem średniokwadratowym (ang. *mean squared error*, MSE) wyznaczonym na podstawie różnic wartości pikseli zdjęcia oryginalnego i zdjęcia przekształconego (szumu). Wartość PSNR wyrażana jest w decybelach (dB) (Galar *et al.*, 2019). Wartość wskaźnika PSNR jest dana wzorem 1:

$$PSNR = 10 * \log_{10}\left(\frac{{v_{max}}^2}{MSE}\right) \quad (1)$$

gdzie:
v_{max} - maksymalna możliwa wartość piksela (dla obrazów 8-bitowych $v_{max} = 255$),
MSE – błąd średniokwadratowy.
Wartość MSE jest dana wzorem:

$$MSE = \frac{1}{N*M*C}\sum_{x=1}^{N}\sum_{y=1}^{M}\sum_{k=1}^{C}[f(x,y,k) - f'(x,y,k)]^2 \quad (2)$$

gdzie:

N, M, C - wymiary obrazu (N - wysokość, M - szerokość, C - liczba kanałów) [px],

f(x, y, k) - wartość piksela o współrzędnych (x, y, k) obrazu oryginalnego [px],

f '(x, y, k) - wartość piksela o współrzędnych (x, y, k) obrazu przekształconego [px].

Drugim szeroko stosowanym wskaźnikiem jest indeks podobieństwa strukturalnego (ang. *structural similarity index*, SSIM) zdjęć przyjmujący wartości od -1 do 1. Tworząc algorytmy poprawy rozdzielczości dąży się do SSIM = 1, gdyż oznacza to, iż zdjęcia są identyczne. Wskaźnik ten zależny jest od trzech składowych: luminancji, kontrastu i struktury, które zadane są odpowiednio wzorami (Wang *et al.*, 2003):

$$l(x,y) = \frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1} \tag{3}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_2} \tag{4}$$

$$s(x,y) = \frac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3} \tag{5}$$

gdzie:

μ_x, μ_y - średnia wartość x lub y,

σ_x, σ_y - wariancja x lub y,

σ_{xy} - kowariancja x i y,

$C_1 = (K_1*L)^2$, $C_2 = (K_2*L)^2$, $C_3 = C_2/2$,

K_1, K_2 - wartości stałe dużo mniejsze od 1,

L - maksymalna możliwa wartość piksela (dla obrazów 8-bitowych L = 255).

SSIM dany jest wzorem:

$$SSIM(x,y) = [l(x,y)]^\alpha * [c(x,y)]^\beta * [s(x,y)]^\gamma \tag{6}$$

gdzie:

α, β, γ - parametry wyznaczające wagi danych składowych. Najczęściej przyjmuje się, że wszystkie są równe 1.

3. KONWOLUCYJNE SIECI NEURONOWE

Sztuczna inteligencja (AI) jest zdolnością komputera, bądź maszyny sterowanej przez komputer do wykonywania czynności intelektualnych typowych dla ludzi, takich jak rozumowanie, odkrywanie znaczeń, odczytywanie informacji, generalizacja czy nauka poprzez doświadczenie. W tak rozumianej sztucznej inteligencji można wydzielić podzbiory: uczenie maszynowe (ang. *machine learning*), głębokie uczenie (ang. *deep learning*),

widzenie komputerowe (ang. *computer vision*), przetwarzanie języka naturalnego (ang. *natural language processing*).

Konwolucyjne sieci neuronowe (CNN) są jedną z najszerzej wykorzystywanych metodyk głębokiego uczenia. Służą one do przetwarzania obrazów w celu detekcji i kategoryzacji obiektów znajdujących się na nich. Działanie takiego algorytmu polega na stworzeniu zbioru treningowego na którym sieć będzie mogła się uczyć. Zaprogramowana CNN "uczy" się odczytywać cechy z obrazów dopasowując je do odpowiednich obiektów w celu ich kategoryzacji. Kluczowa jest ilość danych treningowych oraz ich reprezentatywność.

3.1. Budowa konwolucyjnych sieci neuronowych

Sieć jest zestawem wielu połączonych ze sobą warstw – Rys. 1. Pierwsza warstwa sieci odczytuje macierze pikseli obrazów wejściowych (ang. *input*), wykonuje na nich obliczenia i odpowiednie przekształcenia, a następnie przekazuje wyniki kolejnej warstwie. Każda kolejna warstwa odczytuje wyniki poprzedniej warstwy i tak jak pierwsza opracowuje je i przekazuje kolejnej, aż do uzyskania wyników całej sieci tzw. wyjścia (ang. *output*). CNN ma kilka różnych typów warstw i operacji służących w konkretnym celu.

Operacja konwolucji (ang. *convolution*) polega na przefiltrowaniu obrazów wejściowych za pomocą filtra (ang. *filter*). Takim sposobem otrzymywane są wartości pikseli produktu wyjściowego (ang. *output*), który nazywany jest mapą cech (ang. *feature map*). Jak sama nazwa wskazuje, wynikiem konwolucji są zlokalizowane charakterystyczne cechy obiektów znajdujących się na obrazie poddawanym filtracji. W przypadku zdjęć barwnych filtr przechodzi przez każdy kanał RGB osobno, a następnie wszystkie 3 wyniki są ze sobą sumowane, tworząc tylko jedną mapę cech dla zdjęcia wejściowego.

Obiekty znajdują się w różnych lokalizacjach na zdjęciach, ustawione są pod różnym kątem oraz mogą być inaczej oświetlone. Aby konwolucyjna sieć neuronowa mogła poprawnie wykrywać i kategoryzować obiekty musi być niewrażliwa na ten problem, a więc posiadać niezmienność przestrzenną (ang. *spatial invariance*). By to zapewnić, korzysta się z operacji tzw. *pooling*, działającej bardzo podobnie do konwolucji. Wyniki tych wyliczeń tworzą zaktualizowaną mapę cech, która służy jako wejście do kolejnej warstwy.

Ostatnim fragmentem sieci CNN jest sztuczna sieć neuronowa (ang. *artificial neural network,* ANN). Takie sieci przyjmują dane wejściowe w postaci pojedynczych wartości liczbowych. Sieć konwolucyjna działa natomiast na obrazach, a więc wejściem i wyjściem każdej warstwy są piksele zapisane w sposób macierzowy. By móc połączyć obie te sieci korzystamy z operacji spłaszczenia (ang. *flattening*). Przekształca ona dane postaci macierzowej w pojedynczy długi wektor.

Podczas treningu składnikom warstw przypisywane są wagi odzwierciedlające ich znaczenie w uzyskiwaniu szukanego wyniku. Wagi te są argumentami dostarczanymi do funkcji aktywacji (ang. *activation function*) w celu sprawdzenia ich istotności dla rozwiązania. Jeśli waga przekroczy próg aktywacji, to składnik warstwy jest uznawany za ważny dla procesu treningu sieci i zostaje wykorzystany w jego dalszej części. W przeciwnym wypadku dany komponent zostaje dezaktywowany, a więc nie bierze już

udziału w treningu. Najczęściej wykorzystywanymi funkcjami aktywacji w konwolucyjnych sieciach neuronowych są *ReLU, sigmoid* oraz *softmax*.

3.2. Architektura sieci i proces jej treningu

Konwolucyjne sieci neuronowe zbudowane są z dwóch znaczących części – Rys. 1. Pierwsza z nich odpowiada za ekstrakcję cech (ang. *feature extraction*) z obrazów wejściowych, a więc zbudowana jest z warstw konwolucji (ang. *convolution layers*). Wyniki każdej warstwy (mapy cech) zostają przepuszczone przez funkcję aktywacji ReLU, a następnie poddane operacji *pooling*. Ostateczne mapy cech są poddawane spłaszczeniu (ang. *flattening*), tworząc warstwę spłaszczoną (ang. *flatten layer*). Tak stworzona warstwa jest wejściem do drugiej części sieci CNN, a więc do sztucznej sieci neuronowej (ang. *artificial neural network*, ANN) odpowiadającej za klasyfikację obiektów znajdujących się na obrazie wejściowym. Uzyskane wyniki sieci przechodzą przez funkcję aktywacji *sigmoid* bądź *softmax* w zależności od liczby możliwych rezultatów. Wspomniane funkcje mają za zadanie wyznaczenia prawdopodobieństwa, z jakim skategoryzowano dany obiekt.

Sieć CNN nie potrafi na początku wykonywać zadań, do których jest przeznaczona. Sieć uczy się na podstawie otrzymywanych danych, a więc przygotowanego zbioru treningowego (ang. *training set*). Aby sieć uzyskała jak najwyższą dokładność, dane treningowe powinny być obszerne i różnorodne. Sieć uczona jest przez proces propagacji wstecznej (ang. *backpropagation*), który dzielony jest na 4 operacje: przejście w przód (ang. *forward pass*), funkcja kosztów (ang. *loss function*), przejście wstecz (ang. *backward pass*), aktualizacja wag (ang. *weight update*). Podczas przejścia w przód dane treningowe przechodzą przez całą sieć zgodnie z jej kierunkiem, tworząc wyniki odniesienia. Następnie obliczana jest funkcja kosztów, która określa wielkość błędu między uzyskanymi a oczekiwanymi rezultatami. W kolejnym kroku przeprowadzana jest operacja przejścia wstecz, a więc wykonywane są obliczenia w kierunku przeciwnym sieci. W ten sposób obliczane są gradienty między sąsiadującymi warstwami. Ostatnim etapem procesu propagacji wstecznej jest aktualizacja wag.

Proces propagacji wstecznej wykonywany jest iteracyjnie. Dane treningowe dzielone są na mniejsze zbiory zdjęć (ang. *batches*). Gdy wszystkie zdjęcia zostaną wykorzystane, proces jest powtarzany, a jedno takie zapętlenie nazywane jest epoką. Trening sieci CNN jest często czasochłonny, przetwarza bardzo dużą ilość danych w kilkuset epokach. Zbyt długi czas jego trwania stwarza ryzyko nauczenia się zależności, które nie są prawdziwe. Skutkuje to nadmiernym dopasowaniem uzyskanych wag do danych treningowych – tzw. przeuczeniem sieci. Żeby zapobiec wystąpieniu tego problemu, należy w trakcie treningu oceniać dokładność sieci na niezależnym zbiorze danych walidacyjnych (ang. *validation data*). Optymalna sieć CNN posiada bardzo wysoką precyzję na obu zbiorach danych, zachowując jak najmniejszą możliwą różnicę dokładności między nimi. Świadczy to o tym, że sieć potrafi poprawnie działać na danych innych niż te, na których została nauczona.

3.3. Superrozdzielcze konwolucyjne sieci neuronowe

Superrozdzielcze konwolucyjne sieci neuronowe (ang. *super-resolution convolutional neural networks*, SRCNN) rozwiązują zadanie zwiększenia rozdzielczości przestrzennej metodą porównywania małych fragmentów zdjęć (*patchy*). Superrozdzielcze *patche* łączy się w pary z ich przeskalowanymi do niższej rozdzielczości odpowiednikami. Utworzone *patche* tworzą zbiór treningowy (ang. *training set*), dzielący się na podzbiory *patchy* wejściowych (ang. *input patches*) i docelowych (ang. *target patches*). Sieci CNN trenuje się, by uzyskać jak najlepiej dopasowane wagi, które determinują dokładność sieci. Natomiast w SRCNN najważniejsze są filtry, gdyż to one odpowiedzialne są za zmianę rozdzielczości między zdjęciem wejściowym a wyjściowym. SRCNN trenowane są, aby uzyskać jak najdokładniejsze filtry, a nie ogólną dokładność sieci, która nie odgrywa istotnej roli w tym przypadku. Z tego powodu nie stosuje się również zbioru danych walidacyjnych. SRCNN posiadają warstwy konwolucji, a po każdej z nich stosowana jest funkcja aktywacji ReLU. Operacja *pooling* nie znajduje zastosowania w superrozdzielczych sieciach, gdyż znacznie zmniejsza obraz i służy do uzyskiwania niezmienności przestrzennej (ang. *spatial invariance*), która nie ma znaczenia dla procesu przeskalowywania zdjęć. Podobnie nie są wykorzystywane kolejne operacje, które odpowiedzialne są za klasyfikację. Czyni to sieci SRCNN prostszymi w budowie od sieci CNN, zawierając jedynie warstwy konwolucji i funkcje aktywacji ReLU (Rosebrock, 2017).

Trening sieci CNN wykrywających i kategoryzujących obiekty na obrazach wymaga przeprowadzenia wielu set, a w niektórych sytuacjach nawet tysięcy iteracji (epok). W przypadku sieci SRCNN jest to 10 do 12 epok. Większa liczba iteracji negatywnie wpływa na działanie superrozdzielczych sieci (Dong *et al.*, 2015).

4. EKSPERYMENT

4.1. Dane wejściowe

Eksperyment wymagał utworzenia algorytmu głębokiego uczenia w języku programowania Python. Język posiada wiele bibliotek. Wykorzystano biblioteki: OS, random, regular expression (re), time, Numerical Python (numpy), h5py, OpenCV-Python (cv2), Matplotlib, TensorFlow, Keras i Scikit-image (scimage). OpenCV-Python używany jest do rozwiązywania problemów widzenia komputerowego (ang. *computer vision*) takich jak odczytywanie obrazów jako macierzy pikseli, bądź zapisywane takich danych, jako zdjęcia. Biblioteki TensorFlow i Keras działają równocześnie, dostarczając narzędzi do pracy z zagadnieniami głębokiego uczenia, takich jak gotowe warstwy konwolucyjnych sieci neuronowych.

Danymi wykorzystanymi w eksperymencie są zdjęcia lotnicze, pochodzące z nalotu fotogrametrycznego, przeprowadzonego w kwietniu 2015 roku nad miastem Łódź. Zdjęcia o GSD=5 cm pozyskano z użyciem cyfrowej kamery fotogrametrycznej. Wykorzystano 50 takich zdjęć o wymiarach 17310x11310px. Dane podzielono na zbiór treningowy i zbiór testowy. Zbiorowi treningowemu przypisano 45 obrazów, które w późniejszych etapach posłużyły do wytrenowania sieci SRCNN. Pozostałe 5 zdjęć znalazło swoje miejsce w zbiorze testowym. Jako niezależne dane, posłużyły do analizy wyników wytrenowanej

sieci polegającej na wizualnej ocenie przekształcenia oraz liczbowemu sprawdzeniu jego dokładności wskaźnikami PSNR i SSIM.

Duży format zdjęć (17310x11310 px) dostarcza ogromną liczbę danych, którą trudno było przetworzyć na użytym w eksperymencie komputerze. Żeby rozwiązać ten problem, stworzono algorytm wycinający losowo fragmenty 600x450px, traktowane jako nowe mniejsze zdjęcia. Możliwe jest wycięcie po około 725 unikalnych fragmentów (600x450px) z każdego oryginalnego zdjęcia. Sieć SRCNN poddawano treningom na 5 różnych zbiorach danych, różniących się liczbą fragmentów zdjęć: 315 (7*45), 450 (10*45), 1350 (30*45), 2700 (60*45) i 4500 (100*45). Maksymalnie użyto niemal 1/7 pierwotnych danych.

W następnym kroku należało pozyskać zdjęcia niższej rozdzielczości (zdegradowane), które algorytm będzie porównywał z obrazami oryginalnymi. W tym celu zbiór treningowy dzieli się na dwa podzbiory. Pierwszym z nich jest podzbiór zdjęć treningowych docelowych (ang. *training target set*), który składa się z pozyskanych wcześniej małych fragmentów zdjęć wielkoformatowych. Drugim podzbiorem jest zbiór zdjęć treningowych wejściowych (ang. training input set), które tworzy się poprzez degradację zdjęć należących do pierwszego podzbioru, Degradację wykonano poprzez interpolację dwusześcienną (ang. *bicubic interpolation*). Takim sposobem przygotowane dwa zbiory, tworzące między sobą pary zdjęć oryginalnych i zdegradowanych. W eksperymencie testowym wykorzystano algorytm SRCNN głównie do poprawy rozdzielczości zdjęć zdegradowanych dwukrotnie do GSD=10 cm, jednakże przeprowadzono również testy dla zmian rozdzielczości 3, 4 i 6 krotnych.

4.2. Budowa zaprojektowanej sieci i proces jej treningu

Zaprojektowana sieć SRCNN posiada 3 warstwy konwolucji. Każda warstwa przechodzi przez funkcję aktywacji ReLU (ang. *Rectified Linear Unit*). Zastosowaną funkcją kosztu jest błąd średniokwadratowy, którego użycie pozwala porównać działanie algorytmu na zbiorach treningowym i testowym poprzez obliczenie wskaźnika PSNR. Sieć nauczona na 4500 zdjęciach osiągnęła najlepsze rezultaty ze wszystkich przeprowadzonych treningów. Sieci trenowane na zbiorach 1350 i 2700 zdjęć osiągały niewiele gorsze wyniki, potwierdzając logarytmiczną zależność między ilością danych treningowych a uzyskiwanymi dokładnościami treningu, występującą w każdym rodzaju sieci CNN (Rys. 3).

Oznacza to, że aby osiągnąć niewiele lepsze rezultaty, należy użyć znacznie obszerniejszego zbioru treningowego. Użyty zbiór 4500 zdjęć wiązał się z wytworzeniem 5.535 miliona par patchy, gdzie każdy zapisany w formacie HDF5 zajmował 6 KB pamięci komputera, co wiązało się z wytworzeniem łącznie ponad 63 GB danych treningowych. Wczytanie takich danych bardzo obciążało procesor, a łącznie z treningiem sieci (sam trening wykonywany już na jednostce GPU) cały proces trwał prawie 8 godzin.

4.3. Analiza wyników

Sieci wytrenowane na zbiorach 1350 i 2700 zdjęć uzyskały wyniki treningu niewiele gorsze niż sieć wytrenowana na 4500 zdjęciach. Większość treningów była przeprowadzana na próbce 450 zdjęć treningowych. Pozostałe treningi sieci przeprowadzane były tylko dla dwukrotnej poprawy rozdzielczości. Po zbadaniu ich jednak na zbiorze testowym okazało się, że produkują znacząco więcej artefaktów (szumów). Wynika z tego, że liczbowa analiza wyników wskaźnikami PSNR i SSIM nie jest wystarczająca i powinna zostać uzupełniona wizualną kontrolą wyników. Poddano analizie dokładność działania algorytmu dla wszystkich przeskalowań wskaźnikami PSNR i SSIM. Wyniki prezentuje tabela 1.

Uzyskane wskaźniki pokazują bardzo dobrą skuteczność sieci SRCNN dla podwójnego przeskalowania rozdzielczości przestrzennej obrazów lotniczych. Dla pozostałych skal, wskaźniki nie są aż tak wysokie, jednakże należy pamiętać, że są wynikiem treningu sieci na zaledwie 10% danych wykorzystanych do treningu skali równej 2.

Najlepszy pogląd o skuteczności działania sieci daje wizualna konfrontacja odpowiadających sobie fragmentów zdjęć: oryginalnego, wejściowego (zdegradowanego) i wynikowego. Takie przykłady ilustrują rysunki 4 i 5. Każdy z nich prezentuje serię powiększeń fragmentów zdjęć z różnym stopniem przeskalowania. W podpisach pod rysunkami podano dodatkowo odpowiadające ilościowe wskaźniki PSNR i SSIM, co pozwala wiązać efekt wizualny z ilościowymi miarami jakości.

Zdjęcia przeskalowywane 2-krotnie osiągnęły stosunkowo lepsze wyniki niż reszta z uwagi na użycie 10-krotnie większego zbioru treningowego. Dla pozostałych przeskalowań wyniki wciąż są dobre, podnosząc zauważalnie wartość wskaźników zdjęć wejściowych. Potwierdzają to również wykonane obserwacje wynikowych obrazów.

Mimo że dane użyte w eksperymencie pochodziły z nalotu fotogrametrycznego przeprowadzonego nad miastem, pokrycie terenu występujące na zdjęciach było zróżnicowane, zawierało obok terenów zurbanizowanych również tereny zielone. Postanowiono uzupełnić analizę wyników, wycinając fragmenty zdjęć testowych tak, aby przeważało na nich jednorodne pokrycie terenu (tab. 2).

Rezultaty badań przedstawione w tabeli 2 prezentują ciekawe zależności. Wyniki poparto dodatkowo analizą wizualną. Dane treningowe zawierały przewagę terenów zurbanizowanych, z czego wynikają mniejsze wahania wskaźnika SSIM i dużo wyższy maksymalny wynik PSNR niż w przypadku terenów zielonych. Tereny zielone reprezentowane przez mniejszą ilość danych, są bardziej niepewne, co pokazują duże wahania wskaźnika SSIM i dużo niższy maksymalny wskaźnik PSNR. Średnie wartości wskaźników PSNR są do siebie zbliżone dla obu typów terenów, jednakże wynika to z większej liczby występujących artefaktów na obszarach zurbanizowanych, zaniżając skutecznie średni wynik wskaźnika. Mimo mniejszej ilości danych dla terenów pokrytych roślinnością osiągnęły one lepszą wierność odwzorowania niż tereny zurbanizowane.

Obecność niejednorodnego pokrycia terenu na wykorzystanych zdjęciach lotniczych mogła wpłynąć na wyniki działania całego algorytmu.

5. PODSUMOWANIE I WNIOSKI KOŃCOWE

Zarówno wskaźniki liczbowe (PSNR i SSIM), jak i obserwacja wyników pozwalają stwierdzić, że rezultaty osiągnięte przez stworzoną sieć SRCNN są bardzo dobre dla wszystkich zbadanych przeskalowań rozdzielczości przestrzennej. Tym samym potwierdzono skuteczność algorytmu głębokiego uczenia na danych fotogrametrycznych. Technologia ta może być wykorzystana do tworzenia produktów fotogrametrycznych o lepszej jakości oraz w celu detekcji i klasyfikacji obiektów na obrazach lotniczych. Można również połączyć tak stworzony algorytm z typową siecią CNN wykrywającą i kategoryzującą obiekty na zdjęciach w celu zwiększenia jej skuteczności. Wyniki eksperymentu potwierdzają, że sieci SRCNN to technologia, którą można się posłużyć, dla zautomatyzowania i zoptymalizowania wiele zadań zarówno fotogrametrycznych, jak i teledetekcyjnych, wyręczając żmudną pracę człowieka.

Wyniki przedstawione w pracy poglądowo pokazują skuteczność sieci SRCNN. Jednakże, efektywne wykorzystanie takich algorytmów wymaga użycia komputerów o dużej mocy obliczeniowej. Pozwalałoby to również na trenowanie sieci na większych zbiorach danych.

Jako że sieć SRCNN osiąga różne wyniki dla terenów zurbanizowanych i terenów zielonych można dodatkowo stworzyć algorytm wspomagający, wykorzystując standardowe sieci CNN. Taki algorytm zostałby nauczony rozróżniać i sortować fragmenty danych obrazów ze względu na rodzaje pokrycia terenu i traktowałby je oddzielnie, wycinając patche i wprowadzając je do dwóch różnych sieci SRCNN. Wyniki działania obu sieci zostałyby łączone ponownie w jedno zdjęcie o jeszcze skuteczniejszej poprawie rozdzielczości.

Details of authors:

inż. Kamil Bartosik
e-mail: kamil.bartosik.stud@pw.edu.pl

Zdzisław Kurczyński
e-mail: zdzislaw.kurczynski@pw.edu.pl
tel: 22 234 7694

Submitted 1.12.2021
Accepted 31.12.2021