Applied Artificial Intelligence

An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Pattern recognition of soldier uniforms with dilated convolutions and a modified encoder-decoder neural network architecture

Manuel Eugenio Morocho-Cayamcela & Wansu Lim

APPLIED ARTIFICIAL INTELLIGENCE
2021, VOL. 35, NO. 6, 476–487
https://doi.org/10.1080/08839514.2021.1902124

ARTICLE

Pattern recognition of soldier uniforms with dilated convolutions and a modified encoder-decoder neural network architecture

Manuel Eugenio Morocho-Cayamcela[a,b,c] and Wansu Lim[d]

[a]Yachay University of Experimental Technology and Research, School of Mathematical and Computational Sciences, San Miguel De Urcuquí, Ecuador; [b]Scientific Computing Group (SCG) (www.yachay-scg.com); [c]Smart Data Analysis Systems Group (SDAS) (www.sdas-group.com); [d]Department of Aeronautics, Mechanical and Electronic Convergence Engineering, Kumoh National Institute of Technology, Gumi-si, Republic of Korea

ABSTRACT
In this paper, we study a deep learning (DL)-based multimodal technology for military, surveillance, and defense applications based on a pixel-by-pixel classification of soldier's image dataset. We explore the acquisition of images from a remote tactical-robot to a ground station, where the detection and tracking of soldiers can help the operator to take actions or automate the tactical-robot in battlefield. The soldier detection is achieved by training a convolutional neural network to learn the patterns of the soldier's uniforms. Our CNN learns from the initial dataset and from the actions taken by the operator, as opposed to the old-fashioned and hard-coded image processing algorithms. Our system attains an accuracy of over 81% in distinguishing the specific soldier uniform and the background. These experimental results prove our hypothesis that dilated convolutions can increase the segmentation performance when compared with patch-based, and fully connected networks.

ARTICLE HISTORY
Compiled 16 March 2021

Introduction

Countless efforts from military and civil defense agencies in the last decades have focused on detecting a known target in a video image Haralick (1979); Haralick, Shanmugam, and Dinstein (1973). The result of this hard work has always been based on image/video processing techniques. However, with the revolution of artificial intelligence in the last couple of years, the classical processing techniques are becoming obsolete Geron (2017); Goodfellow, Bengio, and Courville (2016); Mitchell (1997), in part for the fixed characteristic of the coding, and the almost inexistent versatility and reusability of the code from one application to another. Algorithms for basic segmentation such as *TextonForests* Shotton, Johnson, and Cipolla (2008) and *Random Forest* Shotton et al. (2011) are limited by its low performance. *Patch classification*, where every pixel is classified

CONTACT Wansu Lim, ✉ wansu.lim@kumoh.ac.kr Department of Aeronautics, Mechanical and Electronic Convergence Engineering, Kumoh National Institute of Technology, Gumi-si, Republic of Korea.

individually with patches, is limited by the requirement of fixed size images Ciresan, Alessandro Giusti, and Schmidhuber (2012); Shelhamer, Long, and Darrell (2017). Models based on convolutional neural networks (CNNs) have increased the segmentation performance on popular segmentation datasets such as *MSCOCO* Lin et al. (2014) and *PASCAL VOC 2012* Everingham et al. (2012). Fully connected layer CNN architectures allow generating segmentation from images of any size. Pre-trained CNNs allow to reuse the learned features for new tasks, enabling researchers to develop models faster, with less training data Pan and Yang (2010); Morocho-Cayamcela, Eugenio, and Kwon (2017); Shifat and Jang-Wook (2020).

Even though some pattern recognition techniques have been recently exploited in the classification area, there is no record of using artificial intelligence (AI) techniques to detect the uniformity of soldiers accurately using an image semantic segmentation network. To solve this problem, we propose a segmentation network using two CNNs that reassemble a semantic pixel classifier. This technique has been proven to generalize to any scenario if the training data are well-defined Maggiori et al. (2017a); Badrinarayanan, Kendall, and Cipolla (2015); Morocho-Cayamcela and Lim (2020); Morocho-Cayamcela, Eugenio, and Lim (2020a). We test our segmentation network along with different segmentation techniques from the literature and prove that our design outperforms them for the classes of *soldier* and *background*. Our system attains an accuracy of over 81% in distinguishing the specific soldier uniform and the background from the image.

Model architecture

We use an *encoder-decoder* structure to exploit the multi-scale features in the dataset and perform feature-dense extraction Maggiori et al. (2017b); Badrinarayanan, Kendall, and Cipolla (2015). This is where the encoder–decoder architecture excels at, as it compresses the input to represent all of the information. Our encoder-decoder segmentation network architecture is shown in Figure 1. The encoder stage uses a pre-trained CNN to downscale the images of the soldiers into a feature vector containing a dense pixel-location information. The decoder is employed to expand the compressed feature vector back to a categorical matrix with the original input size Morocho-Cayamcela, Eugenio, and Lim (2020a).

The backbone of the encoder is based on the ResNet-101 architecture He et al. (2016), which is a pre-trained CNN with 101-layers trained on the ImageNet dataset Deng et al. (2009), built with five convolutional (*Conv*) modules, where each one of the modules possesses the same number of convolutional layers as the original ResNet-101. The first four convolutional blocks of ResNet-101 are reused, and the last block is adapted with parallel copies to apply dilated spatial pyramid pooling at different scales. Our model then concatenates the extracted

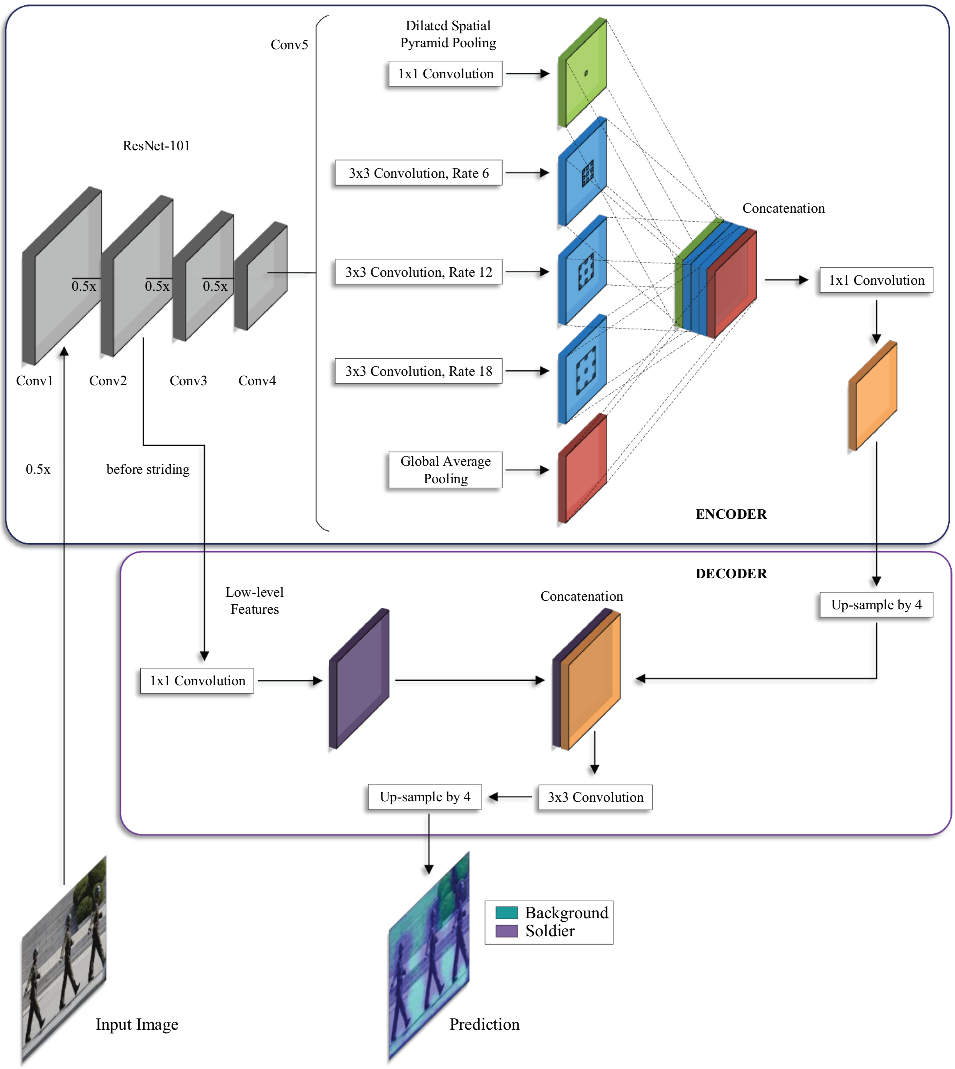

Figure 1. Our model architecture employs an encoder-decoder structure. The encoder applies dilated convolution at different scales to encode multi-scale contextual information. The decoder refines the segmentation along boundaries. Morocho-Cayamcela, Eugenio, and Lim (2020a). 2020 IEEE.

features to send the data to the decoder stage. The dilated convolutions guarantee the robustness of our architecture to environment size changes caused by the multi-scale contextual information encoding. The dilated convolution function is represented as

$$y[i] = \sum_{k=1}^{K} x[i + r \cdot k]w[k] \tag{1}$$

for each position i, on the output y, and filter w. The operation of convolution is dilated over the input map of features x, where the dilation rate r indicates the step at which the input is sampled. The value of r controls the field of view of the convolution. This method can be seen as an analogy to use the convolution function on the input x with up-sampled filters with $r - 1$ zeros added between two values of the sequential filter. Note that the standard convolution is a special case of dilated convolution, with a value of $r = 1$. The filter's field-of-view is regulated by adjusting the value of r. As the sampling rate r increases, the number of weights applied to the effective feature area decreases. Figure 2 illustrates the dilated spatial pyramid pooling (DSPP) process of our system with four parallel functions (1 × 1 convolution, and 3 × 3 dilated convolution with r values of 6, 12, and 18).

The features that were generated in the last step are then concatenated and sent to an additional convolution and batch normalization before the last 1 × 1 convolution. The decoder estimates the feature responses by adding low-level features from the encoder. A four-factor fast bilinear interpolation is implemented before generating the final categorical matrix.

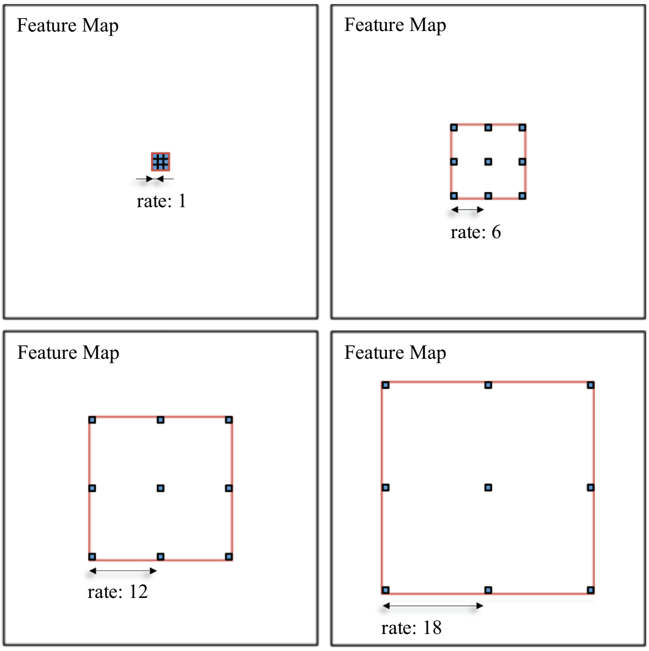

Figure 2. A systematical dilation creates an exponential receptive field growth without losing resolution. The figure presents the dilated convolutions in the proposed architecture with a 3 × 3 kernel and rates r of 6, 12, and 18. Morocho-Cayamcela, Eugenio, and Lim (2020a) 2020 IEEE.

Materials and methods

To build the network, we first created a customized ground truth database (with classes "soldier," and "background"). Our system is trained with these ground truth examples x along with their label y, such that the CNN model can learn to classify new examples. The initial ground-truth database was then used to generate an image data store and a pixel label data store. From the dataset statistics, 23% of the images contained the class "soldier," and 77% of the remaining pixels contained the class "background." Ideally, all classes would have the same number of observations. We solved this class weighting issue by normalizing the input data. A random split of 60% of the images for the training stage and 40% for the testing/validating is employed for the analysis. Figure 3 shows a subset of soldier images used as ground truth in our segmentation network.

The segmentation network was built using VGG-16 He, Shaoqung, and Jian (2018), a pre-trained CNN in order to transfer the initially learned weights to our segmentation network. Data augmentation techniques such as random translation and reflection were added to make the network robust to variability in the input data. Figure 5 illustrates the proposed system.

The model is trained by measuring how much each pixel belongs to a particular ground truth pixel in each iteration. To measure the performance of our model, we employed the difference between the probability distribution of the ground truth and the output using pixel-wise *cross-entropy*. If the predicted probability is different from the ground truth, the loss will increase. Our selected loss function is based on parameters, and the objective of our model is to find these parameter values that minimize the cost function. The training set has the values of $(x^{(i)}, y^{(i)})$ for $i = 1, \ldots, m$. We find the weights $\theta = \{\theta^{(1)}, \theta^{(2)}, \theta^{(3)}, \ldots, \theta^{(n)}\}$ that minimize $J(\theta)$ (cost function) as follows:

$$J(\theta) = -1 \sum_{\mathrm{i}=1}^{\mathrm{m}} \sum_{\mathrm{k}=1}^{\mathrm{K}} \mathrm{y}_{\mathrm{k}}^{(\mathrm{i})} \log(\hat{\mathrm{p}}_{\mathrm{k}}^{(\mathrm{i})}) \tag{2}$$

where the value $y_k^{(i)}$ is 1 if the target for the i^{th} training example is k; otherwise, it is 0. The gradient vector of this loss function is represented with respect to $\theta^{(k)}$ as

$$_{\theta^{(k)}}J(\theta) = 1 \sum_{i=1}^{m} (\hat{p}_k^{(i)} - y_k^{(i)}) x^{(i)} \tag{3}$$

where $x^{(i)}$ contain the feature values of the i^{th} image, and $y_k^{(i)}$ is the desired output for the i^{th} image in class k. Our model uses a partial differential iterative process to minimize the parameters in $J(\theta)$ Cauchy (1976). To avoid the vanishing gradient problem, θ is initialized using *Xavier*'s technique Glorot and Bengio (2010). The decomposition of the cost function as a sum over the

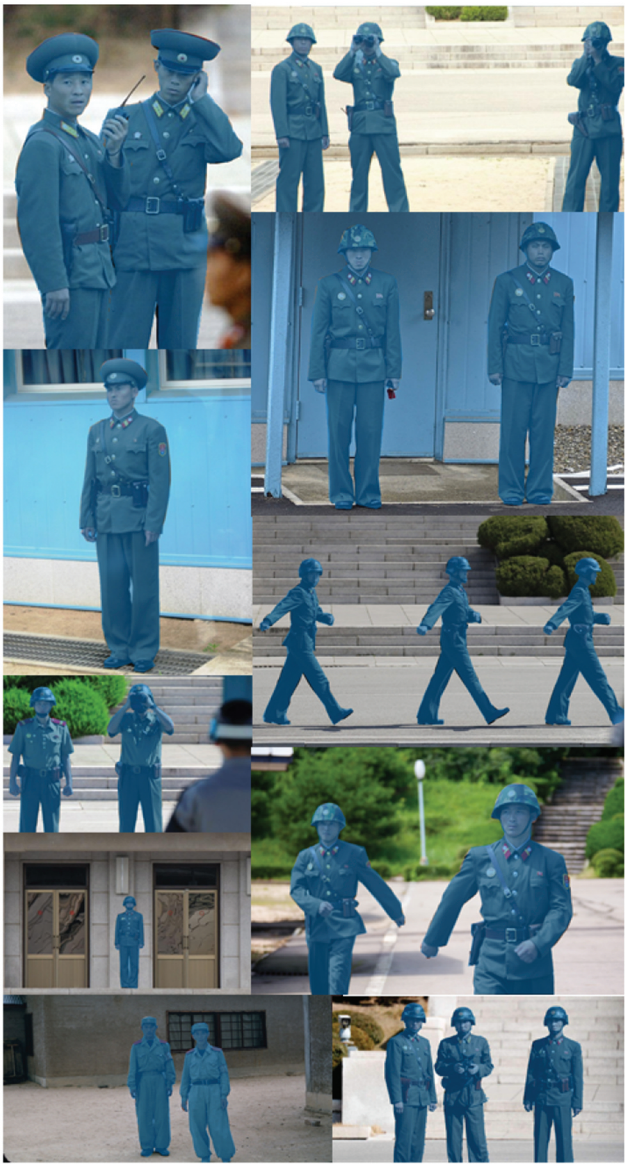

Figure 3. A small subset of labeled (ground truth) images from our database used to train the artificial intelligence-based image semantic segmentation network.

example images can be represented as the negative conditional log-likelihood as

$$J(\theta) = 1 \sum_{i=1}^{m} L(x^{(i)}, y^{(i)}, \theta) \tag{4}$$

with L as the *loss per-example* $L(x, y, \theta) = -\log p(y|x; \theta)$. For these additive loss functions, we estimate

$$_{\theta}J(\theta) = 1\sum_{i=1}^{m} \quad _{\theta}L(x^{(i)}, y^{(i)}, \theta) \tag{5}$$

An extensive computational memory is required to compute ((5)). To balance the system memory usage, our model samples a minibatch of $\mathbb{B} = \{x^{(1)}, \ldots, x^{(m')}\}$ example images before each iteration. In addition, our model set m' to a multiple of m to optimize computation memory Robbins and Monro (1951). Using soldier images from $\mathbb{B}$, the algorithm optimizes the gradient descent as

$$g = \frac{1}{m'} \quad _{\theta}\sum_{i=1}^{m'} L(x^{(i)}, y^{(i)}, \theta) \tag{6}$$

$$\theta \leftarrow \theta - \varepsilon g \tag{7}$$

with the value of ε as the learning rate.

The oscillations in (6) and (7) can cause the algorithm to not converge or diverge. To avoid these oscillations, the proposed model estimates the exponentially weighted average of past gradients and employs them to update θ. The algorithm uses a learning rate ε, an initial velocity v, and an initial set of parameters θ. The gradient is estimated using (8) for every epoch, v is computed with (9), and θ is updated using (10), as follows

$$g \leftarrow 1 \quad _{\theta}\sum_{i} L(f(x^{(i)}; \theta), y^{(i)}) \tag{8}$$

$$v \leftarrow \alpha v - \varepsilon g \tag{9}$$

$$\theta \leftarrow \theta + v \tag{10}$$

with $0 \leq \alpha \leq 1$ as the previous step contribution. Finally, our model uses *maxout* regularization, and *dropout* by overwriting random features to zero to prevent the overfitting problem. Algorithm 4 illustrates a high-level learning process for our image segmentation network.

```
Algorithm 1 Parameter Learning and Optimization
Input: $m$, $K$, $x$, $y$, learning rate $\varepsilon$, momentum parameter $\alpha$.
Output: Optimal hyperparameter values $\theta$ for segmentation.
   Initialization:
1: Initialize $v$ to zero.
2: Initialize $\theta$Xavier's initialization.
   Data acquisition
3: Get soldier images from online server.
   LOOP Data pre-processing
4: for each soldier image do
```

```
    Resize images720 × 720 pixels.
    Image augmentationRandom rotation and translation.
end for
Compute class weighting using the inverse frequency.
    Define the cross-entropy cost function.
$J(\theta) = -1 \sum_{i=1}^{m} \sum_{k=1}^{K} y_k^{(i)} log(\hat{p}_k^{(i)})$
${}_{\theta^{(k)}} J(\theta) = 1 \sum_{i=1}^{m} (\hat{p}_k^{(i)} - y_k^{(i)}) x^{(i)}$
    Calculate the steepest descent with PDEs.
while stopping criterion not met do
    Sample a minibatch $\mathbb{B}$ of $m'$ samples from the training
    set $\{x^{(1)}, \ldots, x^{(m)}\}$ with corresponding targets.
    Compute the gradient estimate:
        $$g \leftarrow \frac{1}{m'} \quad {}_{\theta} \sum_{i} L(f(x^{(i)}; \theta), y^{(i)})$$
    Compute the velocity update: $v \leftarrow \alpha v - \varepsilon g$.
    Apply update: $\theta \leftarrow \theta + v$
end while
return $\theta$
```

Experimental results and simulation

After 200 epochs, the segmentation accuracy of the proposed model is compared against state-of-the-art segmentation techniques from related works. Table 1 shows the accuracy of the two classes under study for the image segmentation models under study. We prove that using transfer learning and combining two CNNs in an encoder-decoder architecture, and employing stochastic gradient descent with momentum as the parameter optimizer, the accuracy of the segmentation attains 81.49% and 82.64% for the *soldier* and *background* classes. Figure 4 shows a subset of segmented images used our proposal. The images in the left show the segmented pixels overlapped with the image from the test set, and the images from the left show the semantic segmentation network pixel labeling overlapped with the ground truth. The green and magenta regions represent the regions where the segmentation results diverge from the expected ground truth. The visual metrics confirm the numerical results.

†Convolutional encoder-decoder architecture, optimized with stochastic gradient descent with momentum.

Table 1. Segmentation accuracy obtained with different models*.

Classes	Texton forests	Patch based	FCNs	U-Net	Proposed model† ‡
Soldier	55.82%	74.03%	75.24%	77.28%	**81.49%**
Background	51.07%	74.12%	76.44%	78.03%	**82.64%**

*Trained using 4 NVIDIA GTX 1080Ti with local parallel pool.

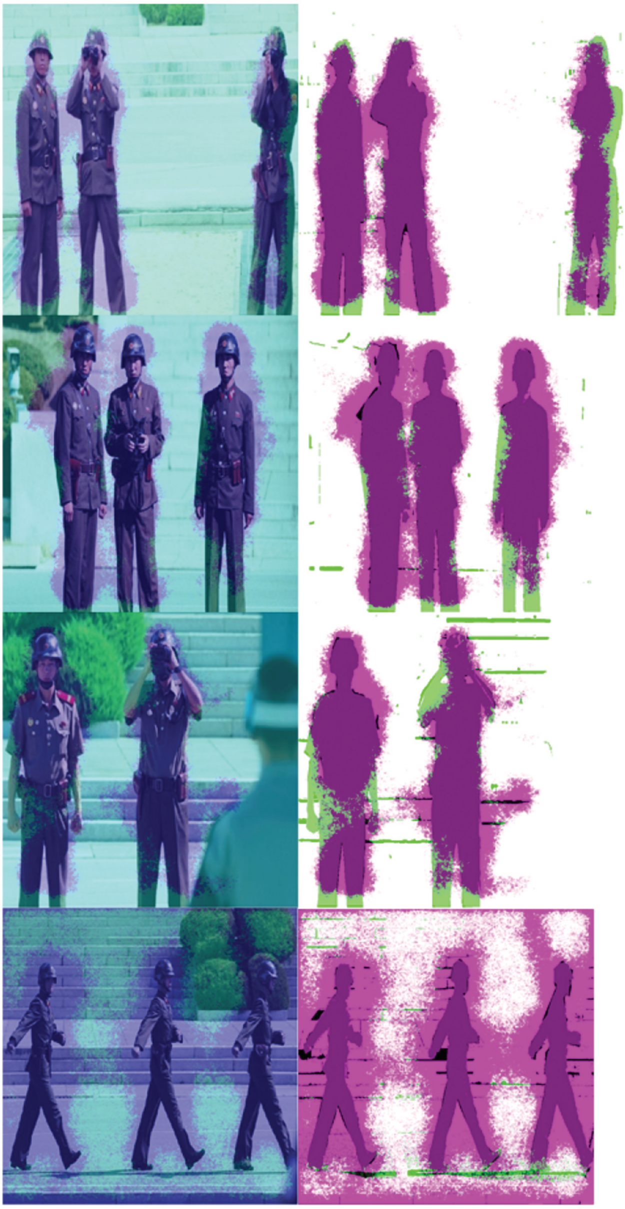

Figure 4. Results from the test set of images. The images from the left show the labeled pixels overlapped with the original image. The images from the left show the labeled pixels overlapped with the ground truth. The green and magenta regions highlight areas where the segmentation results differ from the expected ground truth.

Conclusions

The results obtained from our proposed segmentation network are very promising, with an accuracy over the 80%. The segmentation of the image is by far the most difficult part of a tracking system, with the blobs generated

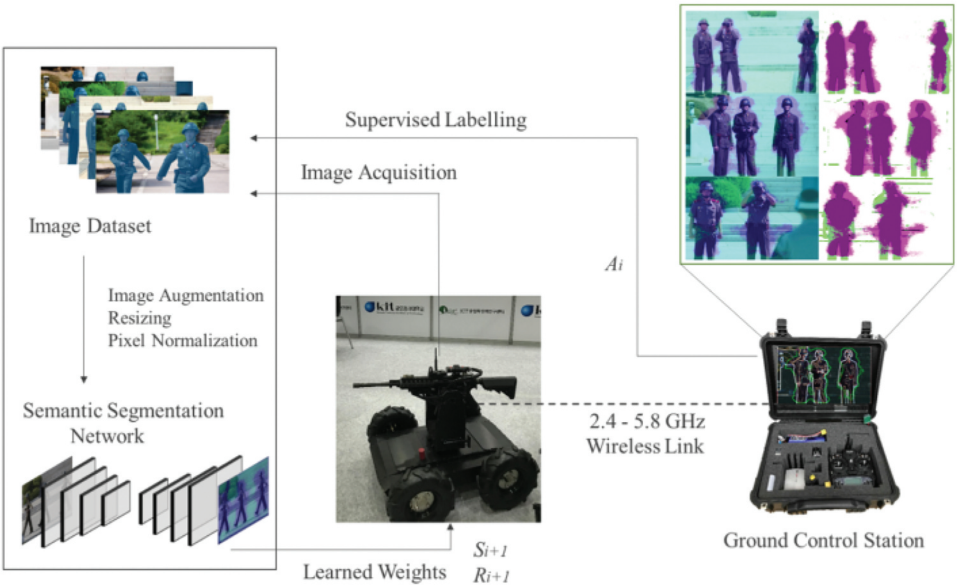

Figure 5. Proposed system architecture. The image dataset and training of the deep learning algorithm are shown on the left side. On the right side, the ground control station takes an action A_i on the environment. The tactical-robot receives a new state S_{i+1} and a reward R_{i+1} (can be positive or negative) based on some policy, and the goal is to find a policy that maximizes the cumulative reward over a finite number of iterations. The green and magenta regions in the resulting images highlight the areas where the segmentation results differ from the expected ground truth.

from the proposed segmentation network we can easily find the center and feedback the information to the camera moving system to change the position and center the target. The proposed segmentation network can also be re-trained over a different dataset to detect other targets, like enemy artillery, or their own soldiers for rescue purposes. The benefit found in the use of our proposed technique is that the CNN can find patterns that most image processing algorithms cannot and are impossible to recognize by the human eye. This technology helps de-camouflaging the targets through exploratory image analysis. The methodology presented in this paper is not intended to replace any triggering mechanism of the tactical robot, but to help the operators to take better decisions in the battlefield.

Disclosure Statement

The authors declare that they have no competing interests.

Funding

This work was supported by the Ministry of SMEs and Start-ups, S. Korea (S2829065, S3010704), and by the National Research Foundation of Korea (2020R1A4A101777511);

Ministry of SMEs and Startups [S2829065];Ministry of SMEs and Startups [S3010704];National Research Foundation of Korea [2020R1A4A101777511];

ORCID

Manuel Eugenio Morocho-Cayamcela http://orcid.org/0000-0002-4705-7923
Wansu Lim http://orcid.org/0000-0003-2533-3496

References

Badrinarayanan, V., A. Kendall, and R. Cipolla. 2015. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (12):2481–95. http://mi.eng.cam.ac.uk/projects/segnet/.

Cauchy, M. A. 1976. Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes Rendus Hedb. Séances Academic Science* 25 (10):536–38.

Ciresan, D., L. M. G. Alessandro Giusti, and J. Schmidhuber. 2012. "Deep neural networks segment neuronal membranes in electron microscopy images." In *NIPS Proceedings: Advances in Neural Information Processing Systems*, 2843–51. https://papers.nips.cc/paper/4741-deep-neural-networks-segmentneuronal-membranes-in-electron-microscopy-images

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. "ImageNet: a large-scale hierarchical image database." *IEEE Computer Vision and Pattern Recognition* http://image-net.org/about-overview

Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, et al. 2012. The PASCAL visual object classes (VOC) Challenge. *International Journal of Computer Vision*, 88: 303–338.

Geron, A. 2017. *Hands-on machine learning with scikit-learn and tensorflow: concepts, tools, and techniques to build intelligent systems.* 1st. Sebastopol, CA:O'Reilly Media, Inc,http://shop.oreilly.com/product/0636920052289.do

Glorot, X., and Y. Bengio. 2010. "Understanding the difficulty of training deep feed-forward neural networks." *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS),* Chia Laguna Resort, Sardinia, Italy 2010 9: 249–56.

Goodfellow, I., Y. Bengio, and A. Courville. 2016. *Deep Learning.* 1st. London, England:The MIT Press,www.deeplearningbook.org

Haralick, R. M. 1979. Statistical and structural approaches to texture. *Proceedings of the IEEE* 67 (5):786–804. http://ieeexplore.ieee.org/document/1455597/

Haralick, R. M., K. Shanmugam, and I. Dinstein. 1973. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-3 (6):610–21. http://ieeexplore.ieee.org/document/4309314/

He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep residual learning for image recognition." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2016-Decem, 770–78. Las Vegas, NV, USA: IEEE Computer Society.

Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. Lawrence Zitnick. 2014. "Microsoft COCO: common objects in context." In *European Conference on Computer Vision (ECCV 2014)*, 740–55. Springer, Cham. http://link.springer.com/10.1007/978-3-319-10602-148

M. E. Morocho-Cayamcela, M. Maier and W. Lim, "Breaking Wireless Propagation Environmental Uncertainty With Deep Learning,“ in *IEEE Transactions on Wireless Communications*, 19 (8): 5075–5087. doi:doi: 10.1109/TWC.2020.2986202

Maggiori, E., Y. Tarabalka, G. Charpiat, and P. Alliez. 2017a. "Can semantic labeling methods generalize to any city? The inria aerial image labeling benchmark." *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* 3226–29. https://hal.inria.fr/hal-01468452

Maggiori, E., Y. Tarabalka, G. Charpiat, and P. Alliez. 2017b. HighResolution aerial image labeling with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing* 55 (12):7092–103. doi:10.1109/TGRS.2017.2740362.

Mitchell, T. M. 1997. *Machine Learning*. 1st ed. New York, United States: McGraw-Hill Science/Engineering/Math. https://www.cs.ubbcluj.ro/gabis/ml/ml-books/McGrawHill-MachineLearning–TomMitchell.pdf

Morocho-Cayamcela, M., H. L. Eugenio, and W. Lim. 2020a. Machine learning to improve multi-hop searching and extended wireless reachability in V2X. *IEEE Communications Letters* 24 (7):1477–81. https://ieeexplore.ieee.org/document/9046001?source=authoralert

Morocho-Cayamcela, M., W. L. Eugenio, and D. Kwon. 2017. "A transfer learning approach for image classification on a mobile device." In *2017 International Conference on Next Generation Computing (ICNGC)*, Kaohsiung, Taiwan, 12, 180–82. Korean Institute of Next Generation Computing. http://www.kingpc.or.kr/wp/http://www.icngc.org/img/file/programSchedule2017b.pdf

Morocho-Cayamcela, M. E., and W. Lim. 2020. Lateral confinement of high impedance surface-waves through reinforcement learning. *IET Electronics Letters* 56 (23):1262–64. doi:10.1049/el.2020.1977.

Pan, S. J., and Q. Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22 (10):1345–59. http://ieeexplore.ieee.org/document/5288526/

Robbins, H., and S. Monro. 1951. A stochastic approximation method. *The Annals of Mathematical Statistics* 22 (3):400–07. doi:10.1214/aoms/1177729586.

Shelhamer, E., J. Long, and T. Darrell. 2017. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (4):640–51. http://ieeexplore.ieee.org/document/7478072/.

Shifat, T. A., and H. Jang-Wook. 2020. Remaining useful life estimation of BLDC motor considering voltage degradation and attention-based neural network. *IEEE Access* 8:168414–28. doi:10.1109/ACCESS.2020.3023335.

Shotton, J., A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. 2011. Real-time human pose recognition in parts from single depth images. *CVPR 2011*. vol. 6, 1297–304. Colorado Springs, CO, USA: IEEE. http://ieeexplore.ieee.org/document/5995316/

Shotton, J., M. Johnson, and R. Cipolla. 2008. "Semantic texton forests for image categorization and segmentation." In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 6, 1–8. IEEE. http://ieeexplore.ieee.org/document/4587503/

Wu, S., Zhong, S. & Liu, Y. "Deep residual learning for image steganalysis". *Multimed. Tools Appl.* 77, 10437–10453 (2018). https://doi.org/10.1007/s11042-017-4440–4