Urban Area Classification Using Deep Convolutional Network

Sami Nieminen
School of Electrical Engineering
Aalto University
Espoo, Finland
sami.nieminen@aalto.fi

***Abstract*—Sentinel-1 synthetic aperture radar (SAR) data can be used to interpret urban areas using deep learning. Particularly convolutional neural networks such as VGG19 and DenseNET have been used to achieve good classification accuracies on urban SAR data. However classification accuracies for general residential areas, storage areas, dense and low-rise residential areas could be improved. This article investigates improving classification accuracies for OpenSARUrban dataset using a convolutional autoencoder. We show that a convolutional autoencoder can be used to improve classification accuracy for urban area Sentinel-1 SAR data. In particular the results show that convolutional autoencoders can be used to reduce classification confusion for selected urban area types such as storage areas. Additionally the ability to separate classes using manifolds is improved. As a consequence we demonstrate that SAR data, particularly Sentinel-1 data can be competitively used to classify and study urban areas. We expect that our results open up further research directions for developing more specific and advanced algorithms for urban area classification with SAR data. Additionally the results contribute towards possibilities for studying urbanization.**

***Index Terms*—Sentinel-1, convolutional autoencoder, SAR, deep learning, urban areas, classification**

I. Introduction

Sentinel-1 is a constellation of two SAR satellites operating in C-band with orbits at 693 km on a near-polar dawn-dusk sun-synchronous orbit producing earth observation data. The default data production mode over land is IW swath mode with 250 km swath with 5m x 20 m ground resolution but also three other modes are available. [1] With the help of deep learning, data from earth observation satellites such as Sentinel-1 SAR can be used to perform land classification without the help of humans. Going even further, classified urban areas can be classified further into different categories, such as skyscrapers, airports, villas, etc.

Zhao et al. [2] created OpenSARUrban data set by patching data into SAR images with help of optical images from Google Earth Engine. This allows OpenSARUrban to test algorithms with urban environment. OpenSARUrban data set contains two types of 20 m x 20 m resolution images of the same area; VH-polarized images and VV-polarized images obtained from ground range detected data with IW swath products. VH-polarized images in general perform better in classifying man-made structures in comparison with VV-polarized images [2]. We decided to only use VH-polarized images in this article.

OpenSARUrban dataset consists of 21 major cities in China such as Beijing, Hong Kong and Shanghai. The target areas are divided to 10 separate categories such as skyscrapers, airports and railways. The dataset has four image categories including original data, gray-scale representation, pseudo-colored visual data and radiometrically calibrated data. [2]

Zhao et al. [2] benchmarked OpenSARUrban with multiple different algorithms, such as VGG19 [3], DenseNET [4], particle component analysis [5], etc. We benchmark convolutional autoencoder with OpenSARUrban dataset and compare our results with resulting confusion matrices to the results of benchmarks from Zhao et al. [2]. General residential and high buildings provided most difficulty for traditional classification algorithms. Autoencoders are neural networks that attempt to encode the input to a latent space decodable to a construct similar to the original input and have been successful at dimensionality reduction tasks. For encoding and deencoding, it is possible to use convolutional neural networks (CNN) as feedforward and deconvoluting the latent space representation. [6]

By using a convolutional autoencoder we try to rectify the classification inaccuracies identified for specific urban area classes. For example Zhu et al. reviewed deep learning methods for SAR data. The possibility of using a deep convolutional autoencoder (DCAE) as a method for automatically discovering features and classifying for SAR data was identified in the article. [7] In particular Geng et al. had used a DCAE to achieve notable performance on TerraSAR-X data. The model consisted of eight layers including a convolutional layer, scale transformation layer, four sparse autoencoder layers and two postprocessing layers. Especially classification accuracy for buildings was good with also a notably good classification accuracy for roads. Both of the notably good accuracies obtained by Geng et al. relate strongly to classifying urban areas suggesting that a deep convolutional autoencoder may give good results. [8]

II. Related Work

Various large-scale image classification SAR projects relate to our work as it is probable to find human settlements over a large enough area. Our work is based on the work by Zhao et al. [2] and in particular the data-set of the work. The dataset is based on Sentinel-1 data obtained from ESA

Copernicus Scihub. The data is based on level-1 ground range detected (GDR) data using IW swath products. The dataset includes both the original GDR images and and radiometrically calibrated data which was obtained with SNAP software. The annotation scheme used for the dataset is a coarse-to-fine scheme where the coarse schemes are residential areas, business areas, transportation hubs, industrial areas and other areas.

In addition to generating the dataset Zhao et al. performed benchmarking of the dataset with deep learning algorithms including DenseNet, ResNet50, SqueezeNet, VGG19 and AlexNet. Additionally the benchmarking included more traditional methods such as local binary patterns (LBPs), LogGabor features, Gabor features and Webel local descriptors (WLD). [2] Overall in their work deep learning models performed better for both VH and VH polarizations. It is notable that VH polarization performed better as a whole for deep learning approaches but for traditional methods both polarizations had a relatable classification accuracy. For 2 out of the 11 algorithms, AlexNet and PCA, VV polarization performed better while for rest the accuracies were equal or in favor of VH polarization. Investigating why the differences between VV and VH appear specifically for deep learning which do not appear with traditional methods could be interesting as this could lead to improvements in deep learning algorithms. Of the algorithms VGG19 [9] performed the best followed by DenseNet [4]. Notably convolutional networks achieved the best performance as we also implement a convolutional network albeit a different one for comparison with the previous results.

SARptical [10], [11] is a dataset using both SAR images and optical images for dense urban areas whereas we focus only on SAR data. With this dataset the challenges of interpreting urban areas due to SAR side-looking geometry caused by layover are addressed.The SAR data is based on TerraSAR-X and optical images were obtained from aerial UltraCAM.The 3-D point clouds were obtained by using SAR tomographic inversion for SAR data and multi-view stere matching for optical data. The 3-D point clouds were then matched to produce a matched 3-D point cloud. The authorgraphicxs noted that the dataset could be used for deep learning applications for dense urban areas.

Calota, Faur and Datcu [12] addressed issues related to applicability of deep learning for Earth Observation. The issues they identified include small number of labeled datasets and differences in data as they are usually based on different instruments. Specifically they studied the impact of image resolution reduction on image classification accuracy for a CNN for dense urban data. The two datasets they used include OpenSARUrban [2] with incremental resolution reductions for training and VHRUrbanSAR containing very high-resolution (5 m) images for fine-tuning. The CNN consisted of three layers with a 2D convolutional layer followed by a max pooling layer. The first two 2D convolutional layers have 32 filters and the filter window sizes are 5x5. The third layer has 64 filters and 3x3 window sizes. The max pooling window sizes are 2x2 for each. The optimizer choice was Stochastic Gradietn Descent (SGD) with a learning rate of 0.0001. The evaluation metrics employed were accuracy, precision and F1. [12] It is worth nothing that the accuracies in the range of 0.58 to 0.70 could most likely be improved by using a CNN with more layers as demonstrated by Simoyan and Zisserman at the expense of more training time [9]. Our work aims to focus on the original resolution of OpenSARUrban and employs a different algorithm.

Sun et al. [13] investigated image segmentation for very-high resolution (VHR) TerraSAR-X data focused on dense urban areas over Berlin. The focus is however on learning multi-level features in order to segment features in the images. To achieve this they introduced building footprints from geographic information system (GIS). The network employed is a conditional GIS-aware network which is based on VGG16 network demonstrated by Simoyan et al. [9]. The nework uses ReLU activation, five convolutional blocks, with each having two or three convolutional layers with 3 x 3 filters. Max-pooling layers employ 2 x 2 filtering interleaved among the convolutional blocks. The GIS module is employed to help with distilling geometry information and normalize them with the final predictions. [13]

Geng et al. [8] overcame problems of absence of feature representation and speckle noise in SAR images with a DCAE. DCAE obtains features by building convolutional units based on transformation, which is designed from GLCM and Gabor transformations. Geng et al. concluded that the new transformation provided better accuracy than GLCM or Gabor individually. However, Gabor provided better accuracy than GLCM invidually and it's accuracy was close to new transformation. After feature extraction Geng et al. applied scale transformation to combat speckle noise. An additional benefit of scale transformation was reduced complexity. After reducing complexity two sparse autoencoders are applied in order to distinguish more discriminating features. First autoencoder has more hidden units than input layer in order to produce linear features. Second autoencoder is more focused on reducing dimensions. Geng et al. utilize supervised manner to fine tune their network, by giving labels to training samples. A softmax classifier is utilized after auto encoders. The network is initialized through greedy layerwise training in order to optimize weights and biases. The classification is done according to maximum probability. Geng et al. use postprocessing in order to project classification to image as a classification map. The image is up-sampled to match the size of original image. Classification accuracy is further improved by applying morphological soothing. Region detection, dilation, and erosion are also implemented in order to improve connectivity of classification, which also increases classification precision. [8] In our work, the end result is not a classification map, therefore post processing done by Geng et al. is not used. We also utilize OpenSARURban data instead of TerraSAR-X data used by Geng et al. used. Therefore further adjustments in DCAE are expected, in order to reach highest classification accuracy.

III. Methodology

As we deploy an already developed model on an existing dataset we employ a methodology based on a light-weight computational modeling approach. We first discuss the dataset and its features followed up by details on model implementation. Lastly we discuss the training, validation and classification evaluation methods.

A. Data and features

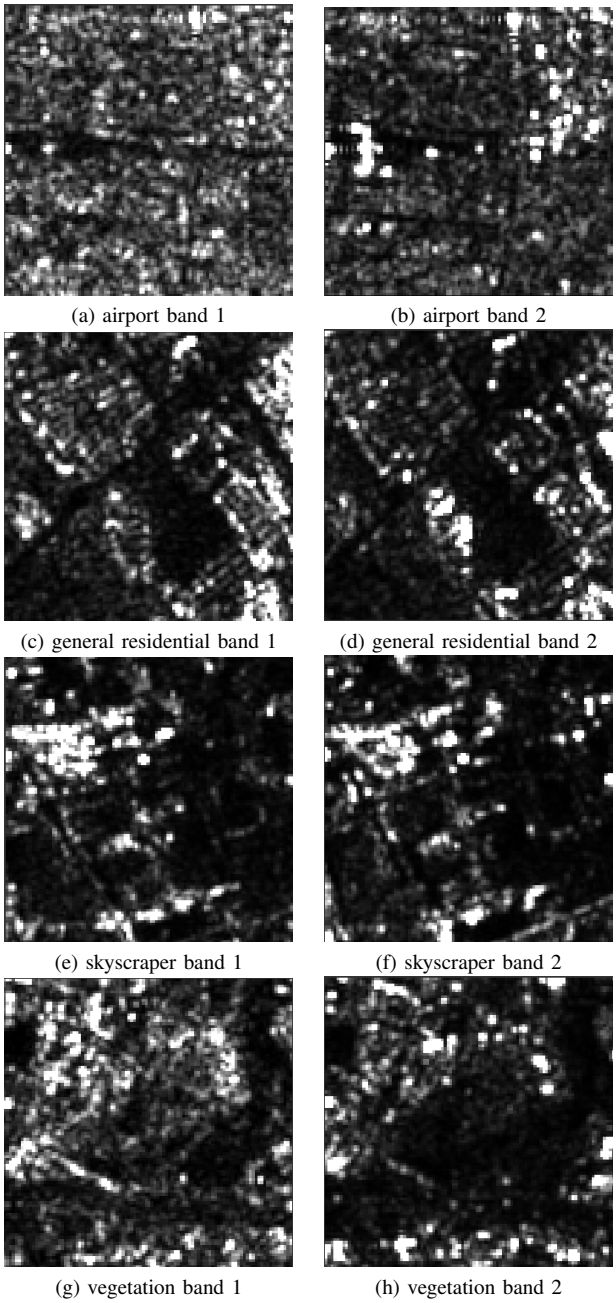
As can be seen from figure 1, the dataset structure stratified sampling should be employed to ensure that all samples are represented in both training and validation sets. In particular airport, highway and railway would be the most likely to be imbalanced. Additionally, as we are dealing with image data we will normalize the data using the obvious quality of images having a value range between 0 to 255.

TABLE I: OpenSAR calibrated subset structure

Class	Count	Percentage
Airport	98	0.61%
Dense Low Residential	1587	9.87%
General residential	3098	19.26%
Highbuildings	4320	26.86%
Highway	202	1.26%
Railway	41	0.25%
Single building	903	5.61%
Skyscraper	645	4.01%
Storage Area	3712	23.08%
Vegatation	1479	9.19%

While loading dataset for the model we noticed some minor discrepancies in the data dimensions that will be investigated in parallel while training the model. However, we are mostly talking about a loss of a few hundred image examples from a dataset of over 30,000 entries, so the final effect is very marginal. Additionally, it is fair to suspect that these images were removed due to issues with the data quality for example.

The dimensions of the loaded data indicate that the size of the dataset is a bit smaller than what was promised in the paper. For calibrated data VV and VH polarized date are available so we can train the model on both of the dataset if we wish to. At the moment we are not sure which polarization is in which band but this will become visible from the results as VH band performs better for classification.



(a) airport band 1 (b) airport band 2

(c) general residential band 1 (d) general residential band 2

(e) skyscraper band 1 (f) skyscraper band 2

(g) vegetation band 1 (h) vegetation band 2

Fig. 1: Example images of various image classes for both bands

B. Model implementation

We start model implementation by building Gabor filters. Gabon filters are formulated as [8]

$$G(x_0, y_0, \theta, \omega_0) = \frac{1}{2\pi\sigma^2} e^{-\frac{x_0^2+y_0^2}{2\sigma^2}} [e^{j\omega_0 x_0} - e^{\frac{w_0^2\sigma^2}{2}}] \quad (1)$$

After first layer, scale transformation is applied to treat speckle noise by average pooling so each patch gets averaged output. After reducing complexity multiple layers of convolutional layers are applied diverging from the approach used by Geng et al [8] due to time and computational resource constraints. Each convolution is performed with a kernel size of 3 and the number of filters is scaled up from 128 to 512. The total number of convolutional layers excluding the last dense layers and gabor computation is 16. In order to prevent overfitting dropout rate of 0.2 is applied every three CNN layers. At the end of the CNN two dense layers are applied with 1024 and 2048 layers respectively. Afterwards the result is flattened and a dense layer with output of 10 for predicting the final 10 classes is applied. When running tests on multiple configurations using a smaller network sigmoid activation function and rmsprop optimizer performed the best. As a consequence they have been used on the larger network.

Layer	Component	Parameters
Input	Input (Gabor+Band 0)	(1310, 100,100,5)
Scale transform	Average pooling	3x3
Layer 1	Conv2D	filt 128, kernel 3
	Activation	Sigmoid
Layer 2	Conv2D	filt 128, kernel 3
	Activation	Sigmoid
Layer 3	Conv2D	filt 128, kernel 3
	Activation	Sigmoid
	Dropuot	0.2
Layer 4	Conv2D	filt 256, kernel 3
	Activation	Sigmoid
Layer 5	Conv2D	filt 256, kernel 3
	Activation	Sigmoid
Layer 6	Conv2D	filt 512, kernel 3
	Activation	Sigmoid
	Dropout	0.2
Layer 7	Conv2D	filt 512, kernel 3
	Activation	Sigmoid
	MaxPool	(2,2)
Layer 8	Conv2D	filt 512, kernel 3
	Activation	Sigmoid
Layer 9	Conv2D	filt 512, kernel 3
	Activation	Sigmoid
	Dropout	0.2
Layer 10	Conv2D	filt 512, kernel 3
	Activation	Sigmoid
Layer 11	Conv2D	filt 512, kernel 3
	Activation	Sigmoid
Layer 12	Conv2D	filt 512, kernel 3
	Activation	Sigmoid
	Dropout	0.2
Layer 13	Conv2D	filt 512, kernel 3
	Activation	Sigmoid
Layer 14	Conv2D	filt 512, kernel 3
	Activation	Sigmoid
Layer 15	Conv2D	filt 512, kernel 3
	Activation	Sigmoid
	Dropout	0.2
Layer 16	Conv2D	filt 512, kernel 3
	Activation	Sigmoid
Dense	Dense	Sigmoid 1024
	Dense	Sigmoid 2048
	Flatten	
	Dense	Sigmoid 10
Output		

Fig. 2: Last implemented model structure

C. Training, Validation and Evaluation

We are employing a non-standard training-validation split for the data due to the heavy imbalance. An additional reason for having employed a non-standard split is due to the heavy resource usage by the model which is limited by computer specs. The training data was limited to a maximum 100 entries per class while the test was limited to 75 entries. As evaluation metric categorical cross-entropy was used as we are attempting to classify 10 categories. As an additional requirement for using Keras categorical cross-entropy was encoding the class labels to one-hot format.

. The two evaluation metrics will be classification accuracy [8]

$$Class(x_i) = argmax_t p(\mathbf{y_i} = t | x_i; \mathbf{W_6}, \mathbf{b_6}) \tag{2}$$

and a confusion matrix as they are employed by the work we want to compare our results against [2]. Besides the original work using classification accuracy there is a strong reason to use a confusion matrix as the dataset is heavily imbalanced and one could obtain a good accuracy just by classifying the

	Airport	Denselow	General Res.	Highbuilding	Highway	Railway	Single building	Skyscraper	Storage area	Vegatation
Airport	0	0	0	0	0	0	0	0	0	0
Denselow	0	0	0	0	0	0	0	0	0	0
General res.	0	0	0	0	0	0	0	0	0	0
Highbuilding	0	0	0	0	0	0	0	0	0	0
Railway	0	0	0	0	0	0	0	0	0	0
Single build.	0	0	0	0	0	0	0	0	0	0
Skyscraper	19	75	75	75	40	8	75	75	75	75
Storage area	0	0	0	0	0	0	0	0	0	0
Vegetation	0	0	0	0	0	0	0	0	0	0

Fig. 3: Confusion matrix results

classes with larger quantities correctly [14]. Example measures becoming available with the confusion matrix are recall

$$R = \frac{TP}{TP + FN}, \tag{3}$$

specificity

$$S = \frac{TN}{TN + FP}, \tag{4}$$

precision

$$P = \frac{TP}{TP + FP}, \tag{5}$$

and F1-score

$$F_1 = \frac{2TP}{2TP + FP + FN}. \tag{6}$$

It is notable that the true positives for some of the more prevalent classes in table 1 had a worse predictive value for [2] than some of less prevalent classes. For example airport had TP of 1.00 while general residential class had a TP of 0.77.

IV. Results

After training a makeshift model due to time limits the model was not able to capture the classes correctly as indicated in figure 2 below. As a hypothesis it seems either the data was not scaled correctly and for example applying standardization or a similar measure more carefully could have helped improve the result. An additional issue could be that the Gabor filters were not applied in a suitable manner. The loss for the categorical cross-entropy got stuck in the range of 2.1 to 2.4 for most of the attempted networks besides the last one for which the structure is reported in model implementation. Additionally when training a simpler network with GLCM mean and variances added to the input with a window size of 3 it did not help improve the result either.

Also due to the issues with making a correct prediction with the model computing the specificity, precision and F1-score values unfortunately adds very little value here.

V. Conclusion

Overall the model described by Geng et al. [8] seems implementable and I imagine it could have obtained good results. However, due to project scope issues we were not able to implement the model properly and the final result was lackluster as well. An additional consideration from student perspective is that your own computer may not always be most suitable for developing some of the models. It is also worth nothing that it maybe could have been possible to improve the VGG16 predictive capability implemented by [2] simply by augmenting it with either precalculated GLCM and Gabor results as they are factors known to contribute towards predictions for SAR data. One more alternative and easier to implement approach could have been to use both VV and VH bands together for making the prediction and comparing it to the VV and VH prediction results of Zhao et al. [2]. For ensuring better results in the future it should be worth the time to study more carefully which data standardization method is suitable for which dataset structure.

References

[1] R. Torres, P. Snoeij, D. Geudtner, D. Bibby, M. Davidson, E. Attema, P. Potin, B. Rommen, N. Floury, M. Brown, I. N. Traver, P. Deghaye, B. Duesmann, B. Rosich, N. Miranda, C. Bruno, M. L'Abbate, R. Croci, A. Pietropaolo, M. Huchler, and F. Rostan, "Gmes sentinel-1 mission," *Remote Sensing of Environment*, vol. 120, pp. 9–24, 2012, the Sentinel Missions - New Opportunities for Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425712000600

[2] J. Zhao, Z. Zhang, W. Yao, M. Datcu, H. Xiong, and W. Yu, "Opensarurban: A sentinel-1 sar image dataset for urban interpretation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 187–203, 2020.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[4] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[5] H. HOTELLING, "Analysis of a complex statistical variables into principal components," *J. Educational Psychology*, vol. 24, 417-441, pp. 498–520, 1933. [Online]. Available: https://ci.nii.ac.jp/naid/30013746117/en/

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[7] X. X. Zhu, S. Montazeri, M. Ali, Y. Hua, Y. Wang, L. Mou, Y. Shi, F. Xu, and R. Bamler, "Deep learning meets sar," 2021.

[8] J. Geng, J. Fan, H. Wang, X. Ma, B. Li, and F. Chen, "High-resolution sar image classification via deep convolutional autoencoders," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 11, pp. 2351–2355, 2015.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[10] Y. Wang, X. X. Zhu, S. Montazeri, J. Kang, L. Mou, and M. Schmitt, "Potential of the "sarptical" system," in *FRINGE 2017*, 2017, pp. 1–6. [Online]. Available: https://elib.dlr.de/115950/

[11] Y. Wang and X. X. Zhu, "The sarptical dataset for joint analysis of sar and optical image in dense urban area," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018, pp. 6840–6843.

[12] I. Calota, D. Faur, and M. Datcu, "Low resolution for dnn in sar," in *2020 IEEE Radar Conference (RadarConf20)*, 2020, pp. 1–5.

[13] Y. Sun, Y. Hua, L. Mou, and X. X. Zhu, "Cg-net: Conditional gis-aware network for individual building segmentation in vhr sar images," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–15, 2021.

[14] S. Rogers and M. Girolami, *A First Course in Machine Learning, Second Edition*, 2nd ed. Chapman Hall/CRC, 2016.