Compressed VGG16 Auto-Encoder for Road Segmentation from Aerial Images with Few Data Training

Abdeldjalil Kebir[1], Mahmoud Taibi[1] and Francisco Serradilla[2]

[1]*LASA Laboratory, Departement of electronics, Badji Mokhtar Annaba University, 23005 Annaba, Algeria*

[2]*Department of Artificial Intelligence, Escuela Técnica Superior de Ingeniería de Sistemas Informáticos,Universidad Politécnica de Madrid, 28031 Madrid, Spain.*

Abstract

Deep Learning methods have found many applications such as segmentation, recognition and classification. However, almost all of these methods require large data-set for the training step and a long training time. Indeed, in surveillance video domain, as for many real-world applications, samples are only accessible in limited amounts owing to acquisition and experiments complexity. In this work, we introduce compressed VGG Auto-Encoder system for road image segmentation in high-resolution aerial imagery. The objective of our experiments is to improve the methodology of distinguishing the road network when only few Data is available. We propose an approach based on compressed Auto-encoder; focus on avoiding the over-fitting effect by generating new data augmentation, based on basic filter transformation to increase and enhance the quality of data training, in the aim of learn an appropriate and simplified representation of data from the original data set in order to obtain a deeper insight from large data-set, and to achieve a quick segmentation training time. Our model achieve a good result and is considered as the best network for fast and accurate segmentation of road images, compared to other models. Furthermore, we provide an explanation of these techniques and some recommendation for their use in the field of deep learning.

Keywords

Auto-Encoder, VGG16, Areal images, Data Augmentation, Road segmentation, Feature extraction

1. Introduction

The segmentation of foreground regions is the key task in many computer vision systems. While segmentation is considered as an essential pre-processing step, it presents a hindrance for many surveillance applications such as traffic monitoring, people counting, and action recognition [1, 2, 3].

Automated road segmentation from aerial imagery is a fundamental unit for many applications [4, 5], including geographic information systems, especially for vehicle navigation, traffic management and emergency response [6, 7]. It is also an important component of military topography and cartography. Furthermore, the extraction of roads taken from aerial or satellite

ICCSA 2021: The 2nd International Conference on Computer Science's Complex Systems and their Applications, May 25–26, 2021, Oum El Bouaghi, Algeria.

kebirabdeldjalil@gmail.com (A. Kebir); mahmoud.taibi@univ-annaba.org (M. Taibi); francisco.serradilla@upm.es (F. Serradilla)

imagery offers an effective solution for the rapid development of new cities [8], which requires frequent road updates.

In recent years, Deep Learning approaches have achieved outstanding performance in many computer vision tasks such as image classification, object and anomaly detection and natural language processing [9, 10, 11, 12]. Indeed, Deep learning is a subset of artificial intelligence. It is distinctive from conventional Machine Learning in how representations are learned from the raw data. Deep learning discovers the hidden structure of complex data using a hierarchical network through multiple layers, where each layer learns representations of data with multiple levels of abstraction. Furthermore, a variety of deep learning methods have been studied and discussed over the past few years [11, 13]. Generally, the Deep learning methods can be divided into four categories: Convolution Neural Networks (CNNs), Restricted Boltzmann Machines, Auto-encoders and Sparse Coding [14].

Nowadays, Deep Learning applications are becoming ubiquitous in the computer vision field, especially for the object segmentation task [15, 16, 17]. As it has show better performance compared to machine learning methods. Most researchers have worked on developing a robust and sophisticated segmenting objects models based on Deep Learning methods such as in [15, 18, 19]. Authors mainly discussed the accuracy of Deep Learning segmentation methods mainly when large training data are available and a powerful computing unit is used. They rarely explored how to train a model when only small data-sets are available.

Learning developing Deep Learning models trained on small data-set is one of the recent research topics in a variety of fields. However, few works have been conducted to address this problem. In medicine for instance, to make a semantic segmentation of the 3D images in abdominal tomography, the authors in [20] used a cylindrical transformation in a cylindrical coordinate system to increase the limited number of images. The results obtained by these transformations have a higher segmentation performance than the FCNs [21] when a limited number of annotated images is used. In [22] the authors proposed a new approach (SSF-CNNN) based on the reduction of the learning parameters number by modification of the structure and strength of the filters obtained by CNN to mitigate the lack of the number of training data. The approach has proven to be effective for multiple object classification and a real-world new-born face recognition problem. However, in [23] the authors increased the number of samples in the database by a radial transformation in the polar coordinate system at the pixel level for each image. The proposed technique has improved the generalization performance of the model for several data-sets.

This our paper is part of deep learning models development in the domain of road image segmentation. In deep learning, high-level features extracted from the network layers are learned from data using a basic learning procedure. In addition, to reconstruct the result segmentation mask easily and to achieve good results from these precise features, the presence of large databases is necessary. In fact, in the real world scenario, a large data-set is not always available. Based on the fact that, we will be going over the three models of auto-encoder road segmentation, one based on VGG16 and the others based on compressed VGG. Then, compare the performance of each approach on a adaptive road benchmark dataset called the Teselas dataset [24].

The objective of this paper is to perform feature extraction and high-dimensional data reduction with different filtering strategies in order to create an adaptive learning data set for accurate road segmentation with less computational time.

The rest of the paper is structured as follows: Section II presents the materials and the proposed methods. We describe the models and evaluate the performance of the proposed approaches in Section III. This is followed by a general discussion and Conclusion in Section IV.

2. Methods and Materials

In this section, we describe the method and material used to build a robust road image segmentation system. We provide an overview and a detailed description of each component introduced in the methodology of our work. As well as, the detail of the overall approach and data training process strategies.

2.1. Auto-Encoder

The Auto-encoder is Deep Neural Network, considered as an unsupervised learning technique. Its main goal is to reproduce its input Data at the output [25, 26] wherein the input and the output layers both have the equal number of neurons. It consists in compressing the data successively until the encoder part obtains a latent form. Then decompress them in the decoder part. It projects the data from a higher to a lower dimension through the use of non-linear transformation and preserve the significant features of the data by deleting the non-essential elements. This lower dimension is used to reconstruct the original data at the output of the network in order to guarantee the accuracy of the input through the usage of non-linear transformation. In fact, there exist four well known Auto-encoders namely; Convolutional auto-encoder, Denoising auto-encoder, Variational auto-encoder and Sparse auto-encoder. Several applications of Auto-encoders can be found in the literature such as Data denoising and Dimensional reduction [27]. The complete network is illustrated in Figure1. The Auto-encoder has three essential blocks:

- **Encoder:** As its name indicates, the encoder attempts to encode all of the useful information about the input into the latent space.
- **Latent space:** The space represented by compressed form of input.
- **Decoder:** In the decoding process the decoder reconstructs the input based only on the information in the latent space.

2.2. Data Augmentation

It is well known that the success of Deep Learning applications is strongly dependent on the amount of data available for its training. To overcome the limitations inherent in small numbers of training samples, we tested data-set augmentation [28]. Image data Augmentation is a regularization process that relies on applying transformations to images for uses both the original image and the transformed images to train model [29, 30]. Therefore, the idea is to use the existing data to create more data, in order to avoid over-fitting, to improve model performance. Some example adjustments include translating, cropping, scaling, rotating, changing brightness and contrast. However, these methods are generally used in all fields of image and video processing such as recognition, detection, segmentation etc.

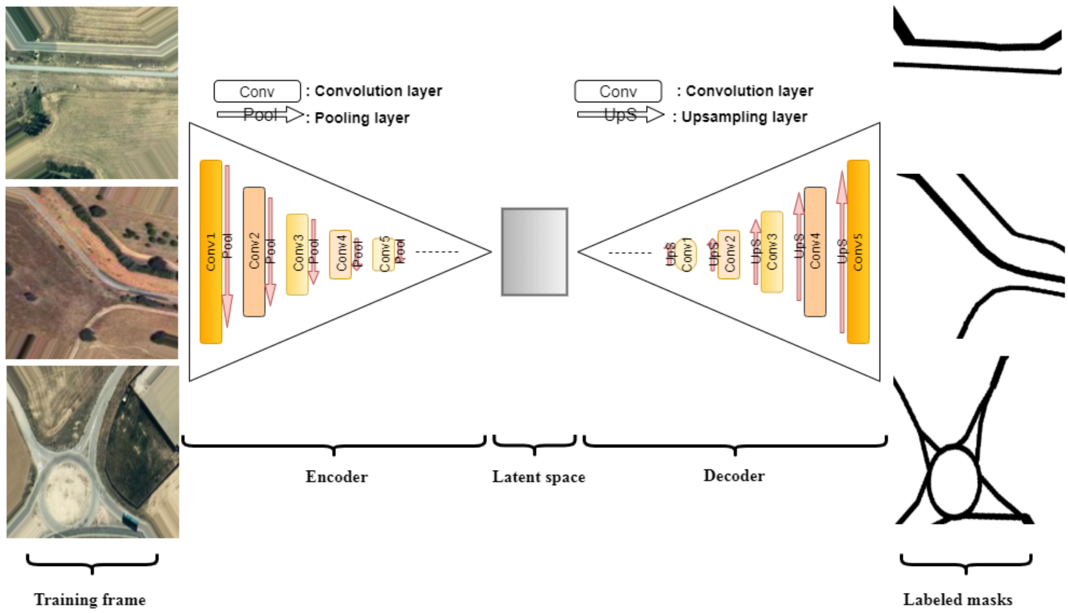

Figure 1: Architecture of General Auto-encoder approach.

2.3. Evaluation protocol

In our experiments we propose approaches based on VGG16 Auto-encoder and Multi-Depth VGG Auto-encoder. In deed, we use the Auto-encoder as supervised learning to perform segmenting the roads for the all approaches.

We construct the VGG16 Auto-Encoder Network with VGG16 model as the encoder part for the first approach by using Pooling and Convolution layers,This process allows to decrease the size of the input data Then, we replaced the fully connected layers as a latent space. For the decoder part, the transposed architecture of the VGG16 has been used for reconstructing the result mask of input frames. This reconstruction process increases the size of the latent space representation to bring it back to its input dimensions by using Upsampling and Convolution layers which are known as transposed VGG16 architecture.

In the Compressed VGG Auto-Encoder approaches, we stacked Convolution and Pooling layers in the Encoder parts for down-sampling the input images, and in the Decoder part we have placed Convolution and Upsampling layers for up-sampling the images in latent space. In fact, the hidden layers are in multi-depth. In such a way that the number of layers will be reduced, i.e. the VGG16 Auto-Encoder model will be compressed to VGG12 and VGG10.

Finally we train the whole models from scratch with Teselas dataset [24].

General approaches VGG16 Auto-Encoder and Multi-Depth VGG Auto-Encoder architecture can be visualizing in Figure4, Figure5 and Figure6.

2.4. Data-set and Metric

We trained and tested our model on Teselas Dataset [24] which contains real aerial images of transport routes, collected in Spanish regions and captured in challenging scenarios. The data-set contains 2 categories (category 1—no road, category 2—road exists), around 9000 labelled tiles (png format). Also, we evaluate the models through several metrics recommended in the literature [31] namely: Specificity, Precision, F-Measure.

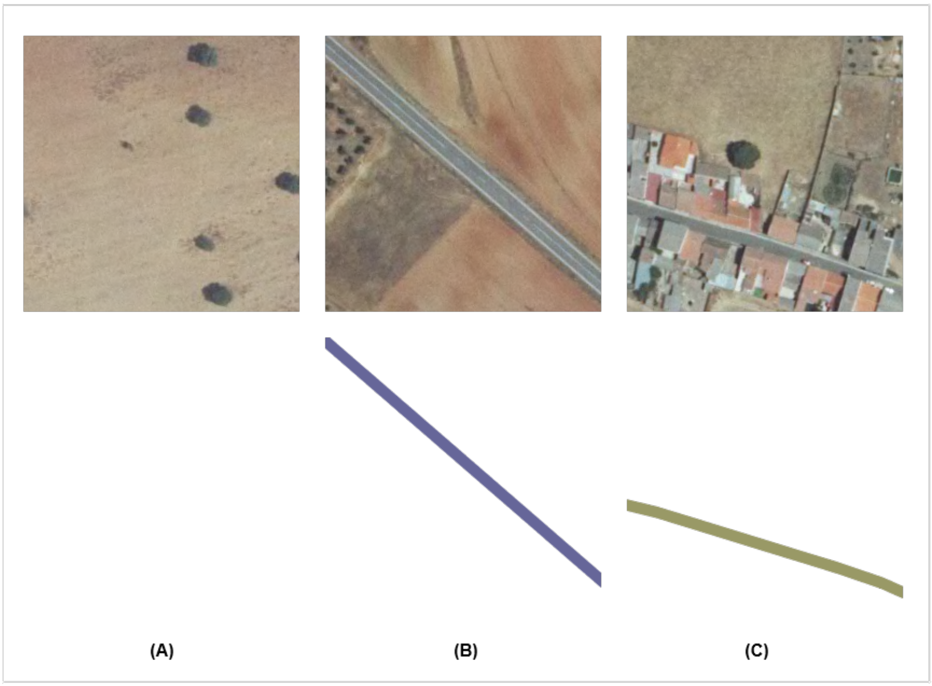

Figure 2: Examples of labelled aerials images of Teselas Dataset (A) Category 1—no road, (B) and (C)category 2—road exists .

2.5. Data Training

In this sub-section, we describe the data training process of our data strategies selection, we selected twenty 520 frames for each category. However, the model was fed through various data training strategy namely:

- **Strategy DA:** consists of augmenting data by generating Three classical transformation frames (rotation to cover the essential angles) for each selected training frame.
- **Strategy Filter:** consists in generating four operation filter transformation by (contour detection, Detail enhance, EDGE enhance and FIND edges) for each selected training frame to facilitate data visualization and eliminate non-informative variables.

Table 1
The description and the number of samples used for the training of each strategy.

Strategy	Description	nbr of samples
FEW DATA	520 frames of each category)x2	1040
CLASSIC DATA AUG	Few Data+(3 transformation DA)	4160
FILTER DATA AUG	Few Data+(3 transformation Filter)	4160

The table 1 shows the description and the number of samples used for the training of each strategy.

The Detailed transformation of different strategies are presented in Figure3.

The AE was implemented (trained and tested) using Keras library [32] programming in Python and the training process on GPU Nvidia GTX1080TI. The models was trained during 150 epochs. We train the networks with RMSprop optimizer. Loss is computed between the Ground truth label and the predicted result using a binary cross entropy loss function. The entire network was trained with 80% frames from the data training, and 20% frames for validate the training of the networks. We also do the evaluation of the models with 50% of the data-set.

3. Experiments and Results

In this section, we describe how the proposed approaches are implemented as well as the obtained results. Then, we compare the three approaches.

3.1. Experiments

For the first approach VGG16 Auto-Encoder part. The explanatory diagram is shown in Figure4.

In the two second approach we built models at different depths, we start with 12 hidden layers then gradually decrease to 10 hidden layers.

The detailed information and structure concerning the layers of the approaches models can be found in figure5 and figure6 .

3.2. Evaluation of the approaches

In this sub-section, we analyze the influence and the result of the training strategies for each approach.

- **Training Time :**

The Training Time of the training strategies results for each approach are shown in Table 2 on second.

The results clearly show that VGG10 AE model trained on New filter data augmentation strategy has less computational time than the other strategies.

To more analyse our models performance we plotting the accuracy and the loss functions between training and validation data for the all approaches trained on different strategies in Figure7, Figure8 and Figure9.

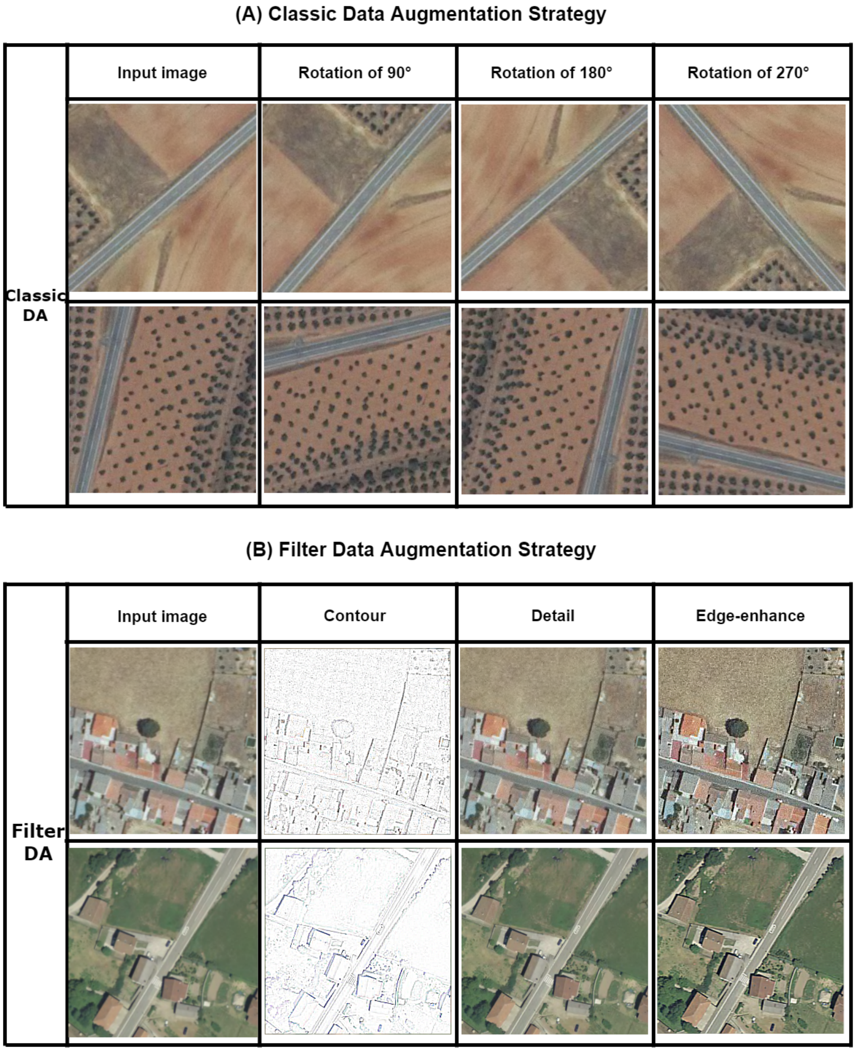

Figure 3: Detailed transformation of the different Data Augmentation strategies.

Table 2
Training Time of the training strategies for each approach

Architecture	Trainable Params	Different training time strategies (S) *Small Data*	*DA Classic*	*DA Filter*
VGG16 AE	**18087715**	328045	1315534	1312389
VGG12 AE	**9590145**	218174	860137	883612
VGG10 AE	**5460097**	163077	658538	**635932**

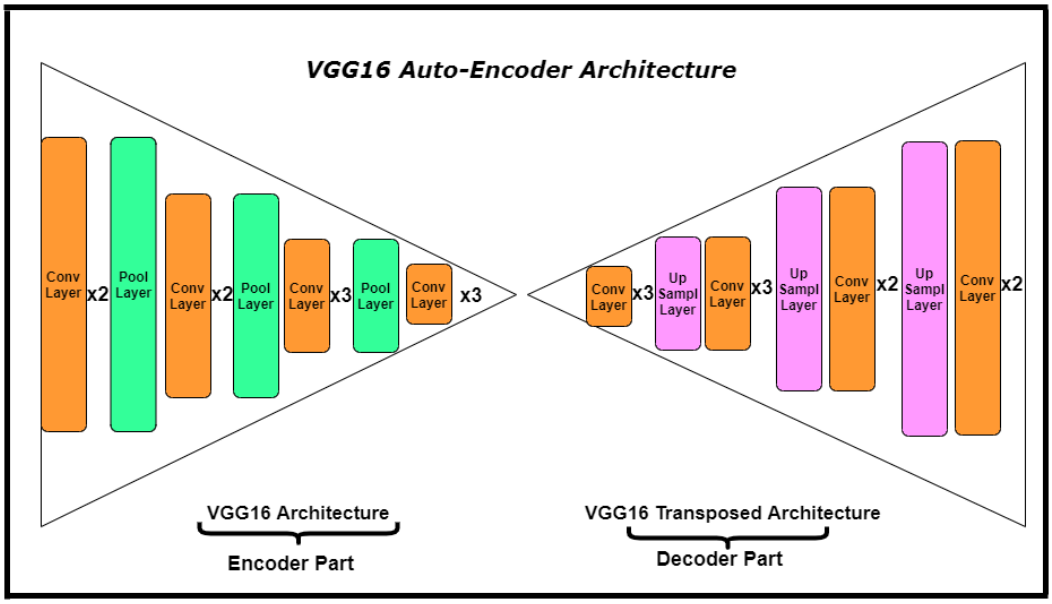

Figure 4: Structure layers of the approach model based VGG16 Auto-Encoder.

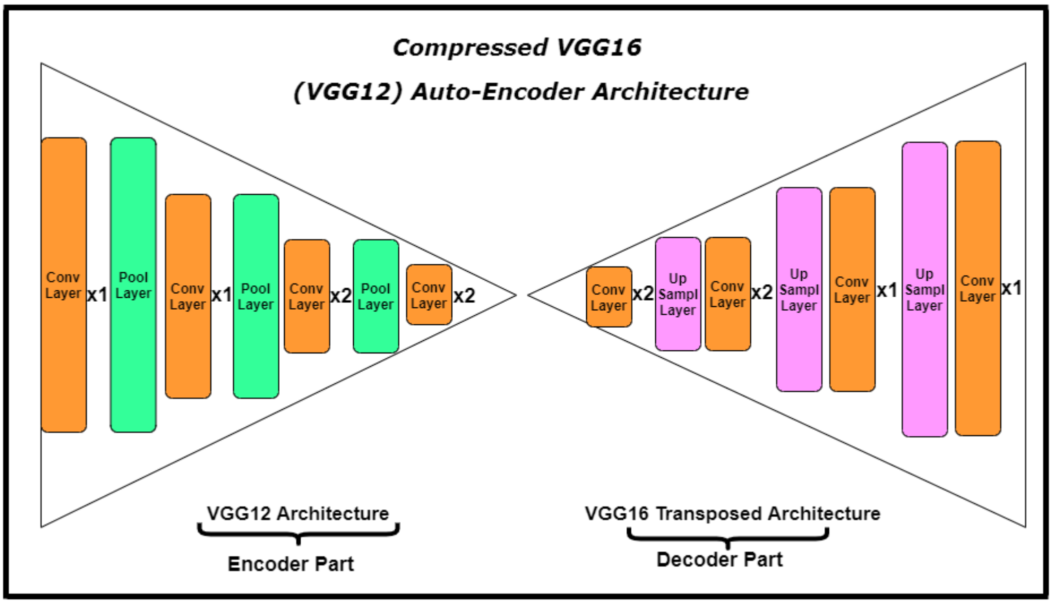

Figure 5: Structure layers of the approach model based VGG12 Auto-Encoder.

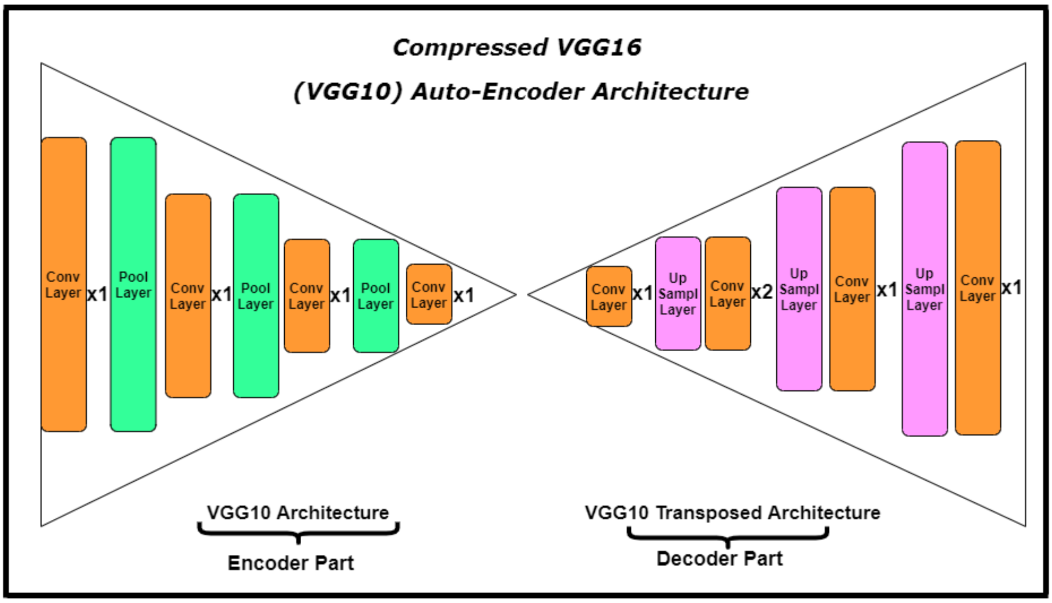

Figure 6: Structure layers of the approach model based VGG10 Auto-Encoder.

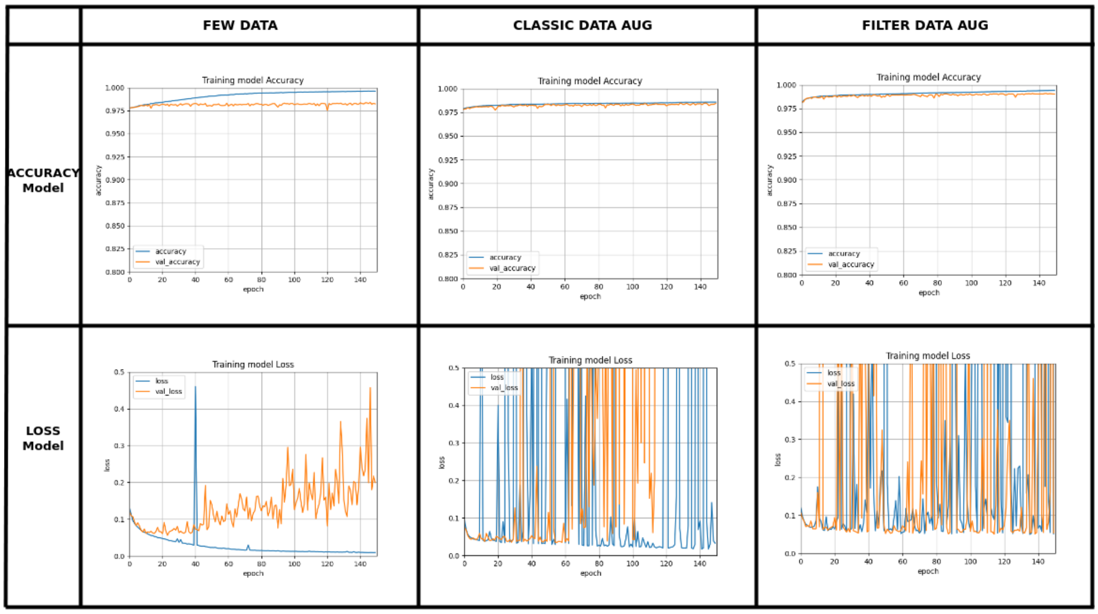

Figure 7: Training and Validation Accuracy/Loss of the used strategies on VGG16 AE.

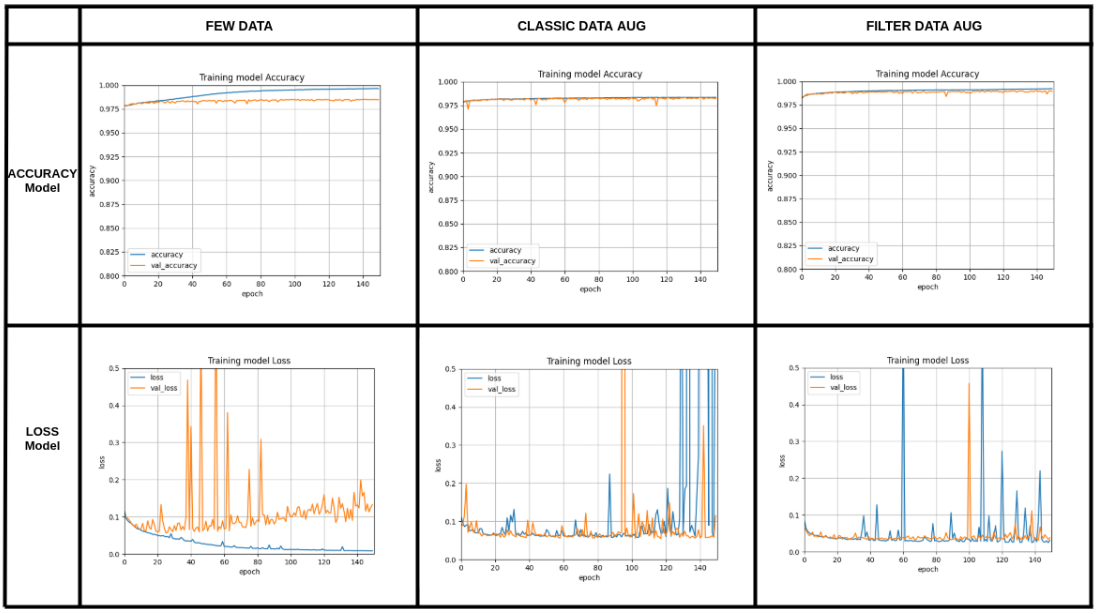

Figure 8: Training and Validation Accuracy/Loss of the used strategies on VGG12 AE.

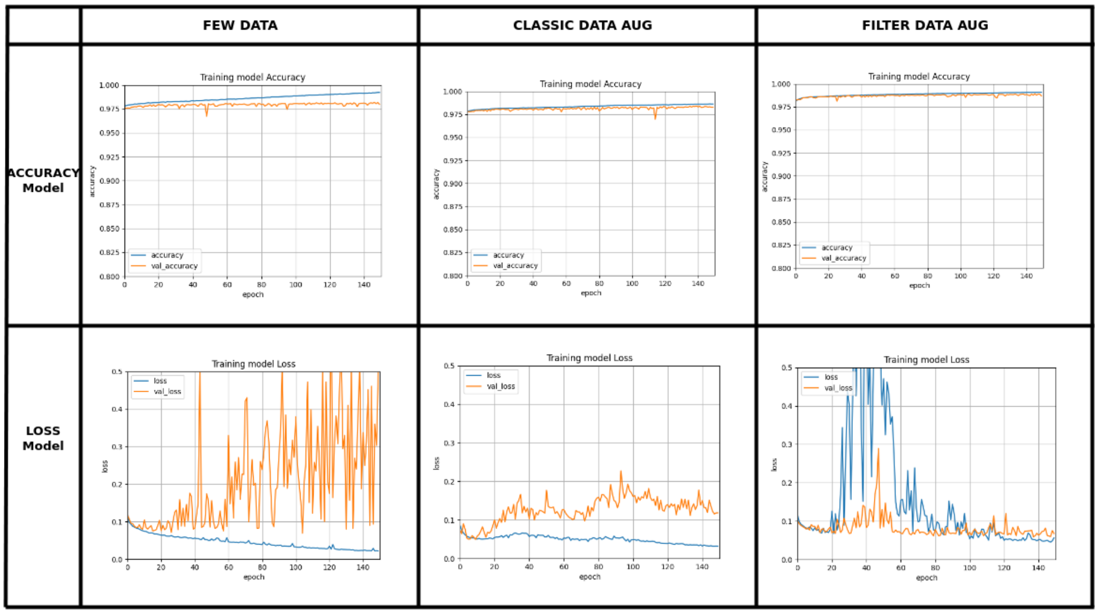

Figure 9: Training and Validation Accuracy/Loss of the used strategies on VGG10 AE.

We can see that for training loss plot of training on data without making any increases for all approaches, the gap between validation and training loss is clear, it shows that the models are over-fitting due to the lack of data training.

From the training loss plot of training on data increased by classical data augmentation of all approaches, we can suggests some intuition that the model is over-fitting, it can clearly be seen starts over-fitting after 50-80 epoch.

From the (VGG10 AE trained on classical data augmentation) training loss plot, there are rarely differences between training and validation loss curves, Therefore, we can see that the effect over-fitting has decreased due to the presence of data augmentation transformation

From the training loss plot of VGG10 AE trained on filter data augmentation strategy, the loss validation and loss training curves are both converging, so the model avoids over-fitting through the new samples added to the training data-set with higher precis features which help and facilitates the system for the learning task, and the number decreased of layer.

For the accuracy curves, observing that with each decrease in the hidden layers of the models the accuracy increases and the both curves converge even more.

We show that the model (VGG10) trained on FILTER data augmentation gives better results compared to others in terms of accuracy and error. When we added more data training through filter transformation in model, we can notice that the validation loss decreased whereas the accuracy validation increased, therefore, we avoided the over-fitting. In this case, the model sounds more robustness.

Our objective is to make a compromise between the four factors (decreasing the number of hidden layers, the error decreasing without presence of over-fitting and the precision increases on less time training possible).

3.3. Comparison of our results

Table 3
A comparison between our result Test approach

Method	FM	Pr	Sp
VGG16 AE	0.785	0.863	0.966
VGG12 AE	0.797	0.899	0.971
VGG10 AE	**0.841**	**0.913**	**0.976**

To make a better visual comparison between our approaches. We present in Figure10, qualitative comparison results, we selected as a demonstrative example two frame from the Teselas data-set.

The first row show input frames, the second row shows the ground truth, the third row present VGG16 AE model, the fourth row show VGG12 AE model and the last row present the results VGG10 AE model. All the models trained on FILTER strategy.

The results from Table 3 and Figure10 shows that VGG10 AE trained on FILTER data augmentation approach has better performance than the other models and shows a very selectivity segmentation.

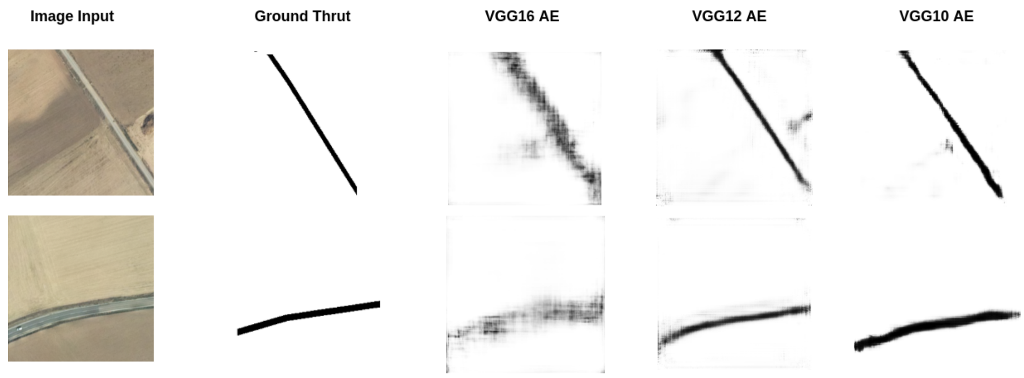

Figure 10: Visual comparison of foreground masks generated by our selected models.

4. Discussion and Conclusion

This paper addresses the problem of object segmentation in high-resolution aerial imagery and discusses the application of deep learning techniques to solve a problem related to segmentation and existence of geo-spatial elements (road network) in the available cartographic support. This challenge is addressed by constructing an auto-encoder neural network trained to segment roads in aerial imagery using manually labelled data.

In this study, we use VGG-Auto-Encoder approach with applying data and feature reduction on the data training using an adaptive data augmentation techniques (FILTER transformation) to perform road segmentation from aerial images model with less training time and avoiding the over-fitting effect.

we presented experiments comparing Multi depth VGG Auto-Encoder Deep Learning approaches trained through three strategies of few data. Our techniques for increasing and enhancing sample training are morphological data augmentation and geometric transformation based on filter transformations. The afore-mentioned techniques have been used in order to minimize and reduce the over-fitting effect as well as to generate the features that are necessary for road segmentation and to increase the model performance in the quickest possible time training.

As can be seen from our previous results, the VGG10 AE model trained on FILTER data augmentation strategy significantly improves the performance in terms of all the objective metrics compared to other strategies for the VGG16 AE and VGG12 AE approaches. From the results obtained, we can see that the results of VGG10 AE model achieve improved performance that obtained by other models.

The results indicate that the compressed learning model VGG10 AE trained on FILTER data augmentation can successfully learn road segmentation generalisation in the short possible time with little learning data. In addition, the preservation of the essential information given through the filter-based transformations (contour and edge enhance), has a purpose to increase the size of the data without losing primary information. As well as, to create adapted and necessary information . As a result, new data augmentation strategy were demonstrated based on the preservation of essential and necessary information to adapt the compress Auto-Encoder technique and a new technique to avoid the over-fitting effect.

The main goal is to enhance object segmentation (road network) in high-resolution aerial imagery technique based on supervised deep Auto-Encoder with few data training. However, we can concluded that filter-based transformations for training model, help the model improve the generalization capabilities and build an accurate model with few data training. Moreover, compared to the traditional techniques of data augmentation (flipping, rotation and translation) that relies on the change of location of the coordinates in the same mathematical plane which produce little improvements. Our work shows the importance of augmenting the data training with purpose, as with for creating a new representation of the variables with the results of the operation filters to extract the variables necessary for the segmentation task such as (edge enhance and contour detection) to eliminate the irrelevant variables that could distort the predictions and to achieve a quick segmentation training time.

References

[1] W. Ge, Z. Guo, Y. Dong, Y. Chen, Dynamic background estimation and complementary learning for pixel-wise foreground/background segmentation, Pattern Recognition (2015). URL: http://dx.doi.org/10.1016/j.patcog.2016.01.031. doi:10.1016/j.patcog.2016.01.031.

[2] T. Bouwmans, Traditional and recent approaches in background modeling for foreground detection: An overview, Computer Science Review 11-12 (2014) 31–66. URL: http://dx.doi.org/10.1016/j.cosrev.2014.04.001. doi:10.1016/j.cosrev.2014.04.001.

[3] H. Liu, X. Han, X. Li, Y. Yao, P. Huang, Z. Tang, Deep representation learning for road detection using siamese network, Multimedia Tools and Applications 78 (2019) 24269–24283.

[4] W. Wang, N. Yang, Y. Zhang, F. Wang, T. Cao, P. Eklund, A review of road extraction from remote sensing images, Journal of traffic and transportation engineering (english edition) 3 (2016) 271–282.

[5] K. K. Eerapu, B. Ashwath, S. Lal, F. Dell'Acqua, A. N. Dhan, Dense refinement residual network for road extraction from aerial imagery data, IEEE Access 7 (2019) 151764–151782.

[6] Y. Wei, Z. Wang, M. Xu, Road structure refined cnn for road extraction in aerial image, IEEE Geoscience and Remote Sensing Letters 14 (2017) 709–713.

[7] L. Zhou, C. Zhang, M. Wu, D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 182–186.

[8] J. Wang, Q. Qin, Z. Gao, J. Zhao, X. Ye, A new approach to urban road extraction using high-resolution aerial image, ISPRS International Journal of Geo-Information 5 (2016) 114.

[9] J. Schmidhuber, Deep Learning in neural networks: An overview, Neural Networks 61 (2015) 85–117. doi:10.1016/j.neunet.2014.09.003. arXiv:1404.7828.

[10] C. Zhao, X. Li, H. Zhu, Hyperspectral anomaly detection based on stacked denoising autoencoders, Journal of Applied Remote Sensing 11 (2017) 042605.

[11] A. Garcia-garcia, S. Orts-escolano, S. Oprea, V. Villena-martinez, P. Martinez-gonzalez, J. Garcia-rodriguez, A survey on deep learning techniques for image and video semantic segmentation, Applied Soft Computing Journal 70 (2018) 41–65. URL: https://doi.org/10.1016/j.asoc.2018.05.018. doi:10.1016/j.asoc.2018.05.018.
[12] M. G. Narasimhan, S. Kamath, Dynamic video anomaly detection and localization using sparse denoising autoencoders, Multimedia Tools and Applications 77 (2018) 13173–13195.
[13] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M. S. Lew, Deep learning for visual understanding: A review, Neurocomputing 187 (2016) 27–48. doi:10.1016/j.neucom.2015.09.116.
[14] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F. E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26. URL: http://dx.doi.org/10.1016/j.neucom.2016.12.038. doi:10.1016/j.neucom.2016.12.038.
[15] Y. Wang, Z. Luo, P. M. Jodoin, Interactive deep learning method for segmenting moving objects, Pattern Recognition Letters 96 (2017) 66–75. URL: http://dx.doi.org/10.1016/j.patrec.2016.09.014. doi:10.1016/j.patrec.2016.09.014.
[16] M. Braham, M. Van Droogenbroeck, Deep background subtraction with scene-specific convolutional neural networks, International Conference on Systems, Signals, and Image Processing 2016-June (2016). doi:10.1109/IWSSIP.2016.7502717.
[17] J. Gracewell, M. John, Dynamic background modeling using deep learning autoencoder network, Multimedia Tools and Applications (2019) 1–21.
[18] M. Babaee, D. Tung, G. Rigoll, A deep convolutional neural network for video sequence background subtraction, Pattern Recognition 76 (2018) 635–649. URL: https://doi.org/10.1016/j.patcog.2017.09.040. doi:10.1016/j.patcog.2017.09.040.
[19] D. M. Vo, S.-W. Lee, Semantic image segmentation using fully convolutional neural networks with multi-scale images and multi-scale dilated convolutions, Multimedia Tools and Applications 77 (2018) 18689–18707.
[20] H. Salehinejad, S. Naqvi, E. Colak, J. Barfett, S. Valaee, Cylindrical transform: 3d semantic segmentation of kidneys with limited annotated images, in: 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), IEEE, 2018, pp. 539–543.
[21] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.
[22] R. Keshari, M. Vatsa, R. Singh, A. Noore, Learning structure and strength of cnn filters for small sample size training, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9349–9358.
[23] H. Salehinejad, S. Valaee, T. Dowdell, J. Barfett, Image augmentation using radial transform for training deep neural networks, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 3016–3020.
[24] C.-I. Cira, R. Alcarria, M.-Á. Manso-Callejo, F. Serradilla, A deep convolutional neural network to detect the existence of geospatial elements in high-resolution aerial imagery, in: Multidisciplinary Digital Publishing Institute Proceedings, volume 19, 2019, p. 17.
[25] K. Lim, C.-s. Kim, Background Subtraction Using Encoder-Decoder Structured Convolutional Neural Network, 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (2017) 1—-6.
[26] S. Leonov, A. Vasilyev, A. Makovetskii, V. Kuznetsov, J. Diaz-Escobar, An algorithm

for selecting face features using deep learning techniques based on autoencoders, in: Applications of Digital Image Processing XLI, volume 10752, International Society for Optics and Photonics, 2018, p. 107522M.

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

[28] K. Seddiki, P. Saudemont, F. Precioso, N. Ogrinc, M. Wisztorski, M. Salzet, I. Fournier, A. Droit, Towards cnn representations for small mass spectrometry data classification: From transfer learning to cumulative learning, bioRxiv (2020).

[29] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, J. Garcia-Rodriguez, A Review on Deep Learning Techniques Applied to Semantic Segmentation, arXiv preprint arXiv:1704.06857 (2017) 1–23. URL: http://arxiv.org/abs/1704.06857. doi:10.1007/978-1-4471-4640-7. arXiv:1704.06857.

[30] Y.-D. Zhang, Z. Dong, X. Chen, W. Jia, S. Du, K. Muhammad, S.-H. Wang, Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation, Multimedia Tools and Applications 78 (2019) 3613–3632.

[31] Y. Wang, P. M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, P. Ishwar, CDnet 2014: An expanded change detection benchmark dataset, IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (2014) 393–400. doi:10.1109/CVPRW.2014.126.

[32] F. Chollet, et al., Keras: The python deep learning library, Astrophysics Source Code Library (2018).