CILAMCE
PANACM
RIO DE JANEIRO 2021

Seismic Facies Segmentation Using Atrous Convolutional-LSTM Network

Maykol J. Campos Trinidad[1,2], Smith W. Arauco Canchumuni[1,2], Raul Queiroz Feitosa[2], Marco Aurelio C. Pacheco[2]

[1]*Dept. of Mechanical Engineering, National University of Engineering*
Av. Túpac Amaru 210, 15333, Lima, Peru
mcampos@uni.pe, saraucoc@uni.pe
[2]*Dept. of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro*
R. Marquês de São Vicente, 225 - Gávea, 22541-041, Rio de Janeiro, Brazil
raul@ele.puc-rio.br, marco@ele.puc-rio.br

Abstract. In this paper, we provided new end-to-end approaches to the task of seismic image segmentation, as human analysis requires a lot of effort and time due to the large pixel dimensions. Given that seismic dataset contains temporal information along its axis (inline and cross-line), we also proposed the use of recurrent neural networks (RNN) together with convolutional layers. After several experiments, we found that the application of crop and rescale (zoom) as a data augmentation technique, as well as the use of focal loss, shows significant improvements in performance and training speed. Our best LSTM-based model achieved a very close to the best one using fewer parameters.

Keywords: Seismic Classification, Semantic Segmentation, Deep Learning, Convolutional Recurrent Networks, Atrous Convolutional

1 Introduction

In Oil and Gas industry reservoir characterization is one of the most crucial tasks in hydrocarbon exploration. One of the stages of this process is the seismic interpretation, which consists of classifying three-dimensional sedimentary units, called lithofacies or facies, from the analysis of geological patterns and characteristics. Due to the large size of these three-dimensional blocks, as well as their complexity, manual facies classification requires a great deal of effort and time on the part of specialists. That is why in recent years, different techniques have been sought to automate this task, among which we can highlight the application of Deep Learning algorithms using convolutional neural network (CNN), due to its spatial awareness and automatic feature extraction [1]. The first papers using these techniques and seismic images began in 2017 given that they require a considerable amount of labeled data, which presumably were not shared because of the difficulty and cost involved in obtaining them.

Initially, image classification approaches were applied. For example, Dramsch and Lüthje [2] compared pre-trained known architectures as VGG16 [3] and ResNet [4], in addition to the one presented by Waldeland and Solberg [5]. Later on, different studies have raised it as a semantic segmentation problem, which in a few words means the pixel-level classification. Zhao [1] proposed to use the encoder-decoder architecture, and compared it with a patch-based model, where the inference represented only the pixel value of the center of the patch, and concluded that his proposal provides better classification quality. On the other hand, Civitarese et al. [6] designed custom models, which were called Danet. These were also encoder-decoder type, although residual units were also used [4] from which it was observed that their Danet-FCN2 model had the best balance in performance and training speed. For that work, it was used datasets annotated by specialists [7, 8].

To promote research in this task, Alaudah et al. [9] made available an annotated dataset, obtained from the well-known F3 block. In addition, metrics were defined and a benchmark was established using encoder-decoder networks, in order to compare possible future approaches. Up to now, different works have been developed to overcome the results [10–12], of which we can highlight our previous work [13] as it achieved great performance using optimization techniques as well as the application of Common Field Pattern (CRF) as post-processing.

Since seismic images have a temporal behaviour because they are generated in sequence as a function of depth, we can use Recurrent Neural Networks (RNN) such as Long Shor-Term memory (LSTM [14]) as it has shown great performance on sequential data. For instance, Shi et al. [15] proposed convolutional LSTM (ConvLSTM) which uses CNNs insted of a fully connected layer in each LSTM cell. Then there were variants such as those proposed by Song et al. [16] who used Bidirectional layers and multi-scale pyramidal architectures for Object Detection, but with a many-to-one (N-to-1) approach. Finally, Chamorro Martinez et al. [17] implemented a many-to-many (N-to-N) configuration for multitemporal remote sensing data, which showed considerable improvements.

This work seeks to establish more advanced benchmarks of the dataset presented by Alaudah et al. [9] using new techniques such as the implementation of the N-to-N approach proposed by Chamorro Martinez et al. [17] in UNet-based models. We divide this work into six sections: Section 2 presents the architectures used to train our models; Section 3 describe the public dataset Netherlands F3 block as the pre-processing procedure. Subsequently, Section 4 explains the training parameters and some more details for post-processing; Section 5 shows and discusses the experiment results; and finally, Section 6 presents the conclusions.

2 Deep Network Architectures

As shown in our previous work [13], the models based on UNet architecture [18] gave best results for seismic facies segmentation. Then, we have implemented variants to UNet such as adding atrous convolution to the bottleneck to extract features at different scales, such as in areas where the classes are very thin. Since our data are made up of sequential images, we found it very appealing to apply Convolutional LSTM, which has shown great information retention capacity for temporal images. Further details will be explained in the following subsections.

2.1 UNet

This topology is made up of two paths. The first one is called encoder and is in charge of extracting the image features while downsampling the image using pooling layers. In our implementation we use blocks with two convolutional layers, each followed by a Batch Normalization layer and ReLU activation. The second path is known as decoder and its function is to recover the original dimension (upsampling), with the number of channels equal to the number of classes of the target image. In our design we use transpose convolutions, which performs the same function of upsampling but with trainable parameters, which give more flexibility to this task. Something very important to highlight in this architecture is the use of skip-connections whose function is not to lose feature map information when concatenating them in parallel.

2.2 Atrous UNet

The implementation of atrous convolution allows us to enlarge the field of view of filters to incorporate larger context as Chen et al. [19] explained. There are different ways to apply it to the UNet architecture, but the one used by us was presented as a solution[1] for a image segmentation competition on Kaggle platform[2], where it reached third place. The UNet with Dilated Convolutions or Atrous UNet (as we called) was also used by Piao and Liu [20] for satellite image segmentation. It consists of the use of several dilated or atrous convolution layers with different rates in bottleneck block. For the incorporation of these dilated convolutions, two modes were proposed: in series or cascade, and in parallel. In both cases, the bottleneck output is represented by the sum of the resulting feature maps.

2.3 Bidirectional UNet ConvLSTM (BiUNetConvLSTM)

As mentioned, the idea of applying convolutional LSTM was generated from the fact of using temporal images. In our first attempts we tried simple versions such as fully convolutional LSTM networks, varying the number of time-steps, but the results were much inferior to those obtained with the previous models. Inspired by Chamorro Martinez et al. [17], we decided to use a UNet-based model for sequential images, where the bottleneck was replaced by a Bidirectional ConvLSTM layer. In addition, we took the many-to-many (N-to-N) approach, where the number of timesteps of images at the input is maintained at the output. Since the training time increased

[1] https://github.com/lyakaap/Kaggle-Carvana-3rd-place-solution/

[2] https://www.kaggle.com/c/carvana-image-masking-challenge

considerably, we decided to reduce the number of convolutional layers to only one per block. For this model, Average Pooling was empirically selected as downsampling operator.

2.4 Atrous Bidirectional UNet ConvLSTM (BiAtrousUNetConvLSTM)

This architecture was born from the union of Atrous UNet and BiUNetConvLSTM. Instead of using a simple Bidirectional ConvLSTM in the bottleneck, several Atrous ConvLSTM were implemented following the same idea of Atrous UNet, thus also resulting in the two mentioned modes (parallel and cascade). Figure 1 shows the parallel mode of this architecture, although we can also use it to clarify the neural networks mentioned above. For example, Atrous UNet differs from this one in that the input is only one image, and the bottleneck is composed only of Convolutional layers with dilated rates. The Table 1 specifies the number of trainable parameters for each architecture described.

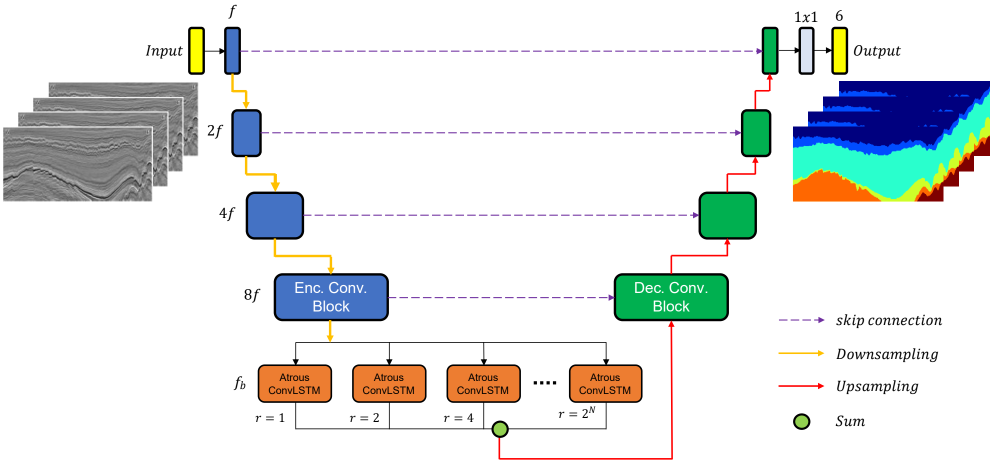


Figure 1. Representation of the architecture Atrous Bidirectional UNet ConvLSTM (BiAtrousUNetConvLSTM).

Table 1. Number of trainable parameters.

Model name	UNet	Atrous UNet	BiUNetConvLSTM	BiAtrousUNetConvLSTM
Millions of parameters	8.6	12.2	2.2	5.6

3 Seismic Dataset

The dataset used was the Netherlands F3 block, which is a fully-annotated 3D geological model open-sourced by Alaudah et al. [9]. They defined six classes, where each one represents a facies with the exception of one that is the union of two facies, Rijnland and Chalk, because they found it difficult to define the limits between them. The three-dimensional block consists of 600 inline and 900 crossline sections, where inline refers to the direction in which the data were acquired; and crossline, its perpendicular direction. In order to get a model that generalizes correctly, ranges were defined to split the data in one block for training and two testing blocks.

The Figure 2 shows the sections that were defined to separate the dataset, as well as the six classes with their respective legends. These three blocks can be obtained from a open repository [3] in an easy-to-access format. Table 2 details the percentage of each class in the training and test blocks, as well as in the totality, where a clear problem of unbalanced data is exposed. From this table we can also highlight that the facies Scruff is most present in test set 2.

[3] `https://github.com/yalaudah/facies_classification_benchmark`

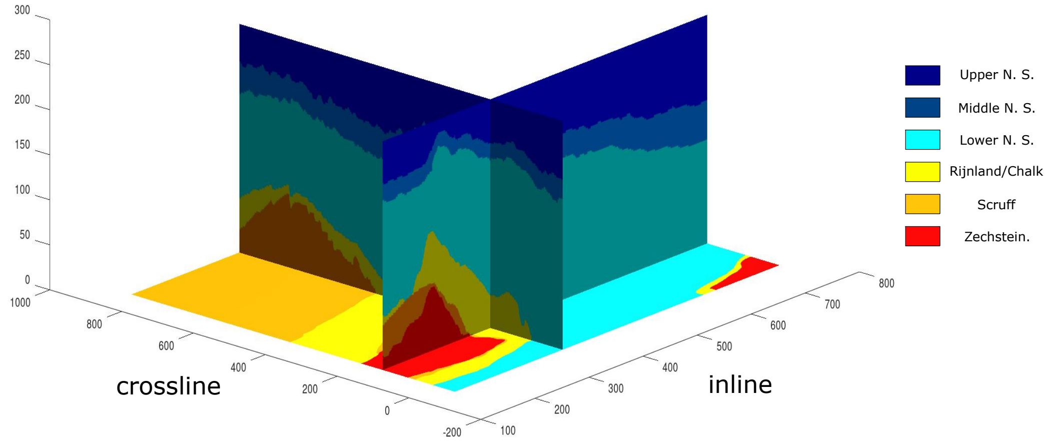


Figure 2. Netherland offshore F3 block in the North Sea, where the six labels are shown.

3.1 Dataset Preparation

After the split, our training dataset is a block of dimension $401 \times 701 \times 255$. As a good practice in machine learning, it is recommended to reserve a part of the training dataset to validate the results, known as validation set. To perform this separation, the technique applied in [6] was used, which consists of dividing the block into n groups, and then taking the first sections of each group to cover 70% for training and the remaining part for validation. This way of separating the data gives us a greater certainty of obtaining sections with a similar percentage ratio of pixels of classes than doing it randomly. In this work, we divided in ten groups for both inline and crossline directions.

Our data generator was configured to deliver the sections in both directions alternately during training. The use of both sections is justified in that the test sets are continuations of the training set in both axes. The data generator was implemented to accept different number of timesteps for ConvLSTM-based models. This implementation configured for only one timestep can be used for the first two architectures.

Since all the models used are based on UNet, our data generator resizes the dimensions of the sections to sizes that are divisible by 16, in order to avoid sizing problems in pooling layers where dimensions are halved. Thus, there were inline and crossline sections of 688×256 and 400×256 pixels, respectively on training set.

Table 2. Percentage of pixels from different classes in each dataset [13]

	Zechstein	Scruff	Rijnland/Chalk	Lower N. S.	Middle N. S.	Upper N. S.
Training set	1.50%	3.27%	6.64%	48.59%	11.88%	28.09%
Testing set 1	1.85%	17.08%	6.95%	45.17%	9.72%	19.19%
Testing set 2	2.72%	0.78%	4.99%	57.20%	10.16%	24.13%
Total	1.87%	6.3%	6.35%	49.62%	10.94%	24.91%

4 Experiments

If we add the number of sections in both axes, we only have 1102 images, which is a relatively small number of samples considered for a deep learning problem. To increase this number (data augmentation), we applied random zoom operation (crop and re-scale) as in [13] as it proved to be very helpful to classify the thinner areas belonging to facies such as Middle North Sea (Middle N. S.), Rijnland/Chalk and Scruff.

To perform the experiments and avoid the class imbalance problem between the majority and minority classes (see Table 2), we set different loss function such as categorical cross-entropy and its weighted version. It was also used the categorical focal loss Lin et al. [21], since it gave the best results in our earlier work, in addition to a higher speed of convergence in training. The main characteristic of this loss function is that it concentrates on the most difficult pixels to classify, which are generally found at the boundary between facies.

In the RNN-based model, we tested varying the fixed size window of our sequenced images with values between 2 and 10, since higher values resulted in very slow training. In addition, as part of our experiments, the N-to-1 approach was also tried, however, it did not show any improvement, so we preferred to avoid showing the results.

All models were implemented using the Keras framework, for different configurations, both in the number of traditional convolution filters and within ConvLSTM cells[4]. We trained them in a GPU Nvidia Volta V100 using the Adam optimizer [22], with a learning rate that starts in $1e - 04$ and is reduced by a factor of 2 each time the loss stopped improving after five epochs. The training stopped if there is no improvement in ten epochs (early stopping).

Table 3. Results of models when tested on both test splits of the dataset.

Model	PA	Class Accuracy						MCA	FWIU
		Zechstein	Scruff	Rijnland/Chalk	Lower N. S.	Middle N. S.	Upper N. S.		
Alaudah et al. [9]	0.905	0.602	0.674	0.772	0.941	**0.938**	0.974	0.817	0.832
UNet [13]	0.939	0.723	0.817	0.797	0.981	0.916	0.972	0.867	0.891
Atrous UNet [13]	**0.943**	**0.764**	0.82	0.774	0.986	0.902	**0.981**	**0.871**	**0.895**
BiUNetConvLSTM	0.936	0.66	0.731	**0.798**	**0.987**	0.932	0.976	0.847	0.884
BiAtrousUNetConvLSTM	0.942	0.595	**0.825**	0.789	**0.987**	0.916	0.977	0.848	0.894

5 Results

Inference was performed along the inline and crossline sections, where probabilities were averaged in both directions. The metrics used were pixel accuracy (PA), class accuracy (CA) for each class (facies), mean class accuracy (MCA) and frequency-weighted intersection over union (FWIU), which is a more suitable multi-class unbalanced version of the well-known mIoU (mean intersection over union) since it uses the pixel frequency as weights in a weighted average.

As it could be observed in Table 3, the Atrous UNet still has the best results. However, since the performance of BiAtrousUNetConvLSTM is very close to that of Atrous UNet, we can say that it is a technical tie considering that a smaller number of trainable parameters were used. Unfortunately, this cannot be said between UNet and BiUNetConvLSTM, although they are still great results compared to the paper that shared the dataset.

Not surprisingly, the most difficult facies to predict are those with the fewest number of pixels (Zechstein, Scruff and Rijnland/Chalk). Added to this is the complexity involved in defining the boundaries between these facies. For example, in the inline 200 of the testing set 1, which is showed in the Figure 3, the facies Rijnland/Chalk is so thin that it can easily be confused with the other adjacent facies.

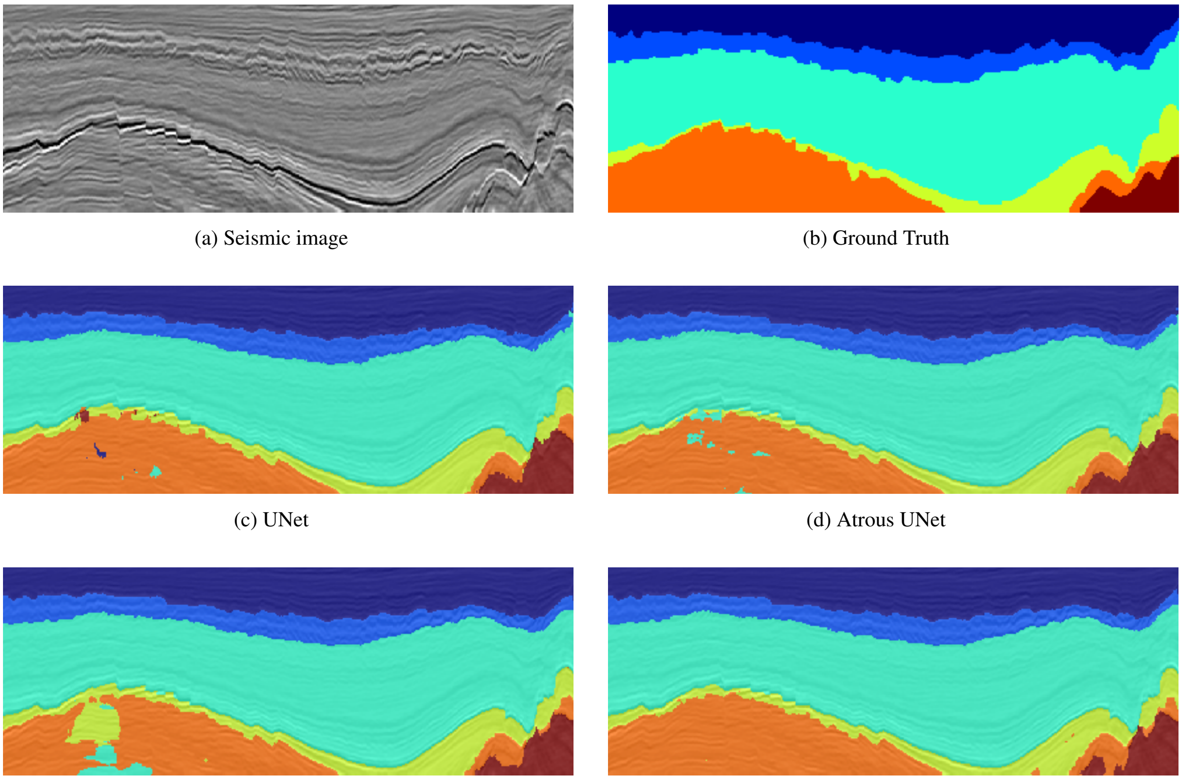
Since in both cases, the Atrous version gave better results, we can say that the implementation of this special bottleneck achieved its goal of extracting features at multiple scales. On the other hand, although the LSTM versions did not manage to outperform the normal ones in their entirety, they did not require as many parameters as their counterparts since the number of convolution layers was reduced by half.

Another point to mention is that in the two best inferences in Figure 3 we can perceive small inconsistencies of misclassified pixels surrounded by a large number of neighbors belonging to the correct class. To solve that problem there are several algorithms such as Conditional Random Fields (CRF) that allow to reclassify pixels according to their neighbors. As in our previous work, we applied CRF to the models, however, it showed no improvement in the LSTM-based models. This can be explained in that these models already consider information from neighboring zones along their timesteps.

6 Conclusions

In this work we implemented new models based on LSTM to observe its performance in the semantic segmentation of seismic images for facies classification, since it is characterized by a good performance in problems involving samples with temporal behavior. The results of our best LSTM-based model were very close to the Atrous UNet architecture, using fewer parameters. These results were achieved thanks to the combination of

[4] `https://github.com/mkl04/Semantic_Segmentation-Seismic_Images`



(a) Seismic image

(b) Ground Truth

(c) UNet

(d) Atrous UNet

(e) BiUNetConvLSTM

(f) BiAtrousUNetConvLSTM

Figure 3. Results on inline 200 from test set #1.

different techniques such as the use of atrous convolutions in the bottleneck, the application of zoom as data augmentation and the implementation of focal loss. As part of the continuous improvement, we tried using CRF for the correction of pixel inconsistencies, but it showed no improvement. We can conclude that this is because the near pixel information was already considered when using LSTM cells as part of the training.

After the implementation of different types of architectures, we can say that there are still a variety of configurations of deep learning techniques that will help us to better model the facies classification.

Acknowledgements. The authors would like to thank at the Applied Computational Intelligence Laboratory (ICA) and Cenpes / Petrobras, partners for 21 years in the research and development of artificial intelligence projects for oil and gas sector.

References

[1] T. Zhao. *Seismic facies classification using different deep convolutional neural networks*, pp. 2046–2050. Society of Exploration Geophysicists, 2018.

[2] J. S. Dramsch and M. Lüthje. *Deep-learning seismic facies on state-of-the-art CNN architectures*, pp. 2036–2040. Society of Exploration Geophysicists, 2018.

[3] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Y. Bengio and Y. LeCun, eds, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[5] A. Waldeland and A. Solberg. Salt classification using deep learning. In *79th eage conference and exhibition 2017*, volume 2017, pp. 1–5. European Association of Geoscientists & Engineers, 2017.

[6] D. Civitarese, D. Szwarcman, E. V. Brazil, and B. Zadrozny. Semantic segmentation of seismic images. *ArXiv*, vol. abs/1905.04307, 2019.

[7] R. Silva, L. Baroni, R. S. Ferreira, D. Civitarese, D. Szwarcman, and E. V. Brazil. Netherlands dataset: A new public dataset for machine learning in seismic interpretation. *ArXiv*, vol. abs/1904.00770, 2019.

[8] L. Baroni, R. M. Silva, R. S. Ferreira, D. Civitarese, D. Szwarcman, and E. V. Brazil. Penobscot dataset: Fostering machine learning development for seismic interpretation, 2019.

[9] Y. Alaudah, P. Michałowicz, M. Alfarraj, and G. AlRegib. A machine-learning benchmark for facies classification. *Interpretation*, vol. 7, n. 3, pp. SE175–SE187, 2019.

[10] A. Bridi Guazzelli, M. Roisenberg, and B. B. Rodrigues. Efficient 3d semantic segmentation of seismic images using orthogonal planes 2d convolutional neural networks. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2020.

[11] M. Q. Nasim, T. Maiti, A. Shrivastava, T. Singh, and J. Mei. Seismic facies analysis: A deep domain adaptation approach. *ArXiv*, vol. abs/2011.10510, 2020.

[12] E. A. Trindade and M. Roisenberg. Multi-view 3d seismic facies classifier. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, SAC '21, pp. 1003–1011, New York, NY, USA. Association for Computing Machinery, 2021.

[13] M. J. Campos Trinidad, S. W. Arauco Canchumuni, and M. A. Cavalcanti Pacheco. Towards a benchmark for sedimentary facies classification: Applied to the netherlands f3 block. In J. A. Lossio-Ventura, J. C. Valverde-Rebaza, E. Díaz, and H. Alatrista-Salas, eds, *Information Management and Big Data*, pp. 211–222, Cham. Springer International Publishing, 2021.

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, vol. 9, n. 8, pp. 1735–1780, 1997.

[15] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pp. 802–810, Cambridge, MA, USA. MIT Press, 2015.

[16] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam. Pyramid dilated deeper convlstm for video salient object detection. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds, *Computer Vision – ECCV 2018*, pp. 744–760, Cham. Springer International Publishing, 2018.

[17] J. A. Chamorro Martinez, L. E. Cué La Rosa, R. Q. Feitosa, I. D. Sanches, and P. N. Happ. Fully convolutional recurrent networks for multidate crop recognition from multitemporal image sequences. *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 171, pp. 188–201, 2021.

[18] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, eds, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham. Springer International Publishing, 2015.

[19] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, n. 4, pp. 834–848, 2017.

[20] S. Piao and J. Liu. Accuracy improvement of UNet based on dilated convolution. *Journal of Physics: Conference Series*, vol. 1345, pp. 052066, 2019.

[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, vol. abs/1412.6980, 2015.