Salt segmentation using deep learning

Alan Souza, Wilson Leão, Daniel Miranda, Nelson Hargreaves, Bruno Pereira Dias and Erick Talarico

PETROBRAS Petróleo Brasileiro

Abstract

The current work presents convolutional neural networks being used for semantic segmentation of salt bodies on seismic volumes. The salt body could be used to extract seismic horizons or geobodies. Our contribution is twofold. First of all, it is presented that convolutional neural networks can behave as an auxiliary tool by the geophysical interpreter. Furthermore, it is analyzed the amount of input data that is required to have a good prediction by the network.

Introduction

Convolutional Neural Networks (CNN) are being used on image classification for more than two decades (LeCun et al., 1998). There was a resurgence of these techniques after the breakthrough work of Krizhevsky et al. (2012). Since then the field has been expanding with practitioners applying these techniques in a great variety of fields and problems. One of the first works using CNN for semantic segmentation applied a modified version of a classification network to generate per pixel classification (Long, 2016; Long et al., 2014).

Recently, CNNs and others neural networks have been used for seismic classification and segmentation. One of the first of those works classified only the central pixel of a seismic sub-cube (Waldeland and Solberg, 2017; Waldeland et al., 2018). Per pixel with sliding window predictions were not considered an ideal procedure due to their high computational cost and they have been rapidly replaced by segmentation networks (Gramstad and Nickel, 2018; Karchevskiy et al., 2018).

During the course of this work, two types of segmentation network archictectures were used to study salt interpretation, analysing the amount of input data needed to have a good prediction and the quality of automatic horizons that results from our workflow. Also, different training strategies were tested in order to achieve higher predictive power and, more importantly, the amount of training data that is necessary as a start to accelerate the specialist's work.

Theory

As usual when using deep learning techniques we define a parametrized function $f(\mathbf{x}, \theta)$ with $\mathbf{x}$ being a (small) seismic image patch used as input and θ the trainable (optimizable)

-	U-Net	DeepLab
ℓ_1	binary crossentropy	focal loss with label smoothing
ℓ_2	dice loss	ternaus loss
ℓ_3	None	binary cross entropy

Table 1: Losses functions used to train the two models considered. The focal loss (Lin et al., 2017) was modified to incorporate label smoothing (Szegedy et al., 2015) that acts as regularizer (Pereyra et al., 2017). The third loss used on the Deeplab classifies if the patch has salt or not (deep supervision) and also act as regularizer (Kornblith et al., 2018)

weights (parameters) of the segmentation network. A variety of neural networks architectures exists for semantic segmentation, for instance see (Ronneberger et al., 2015; Zhao et al., 2016; Chen et al., 2018).

The training procedure is performed by optimizing 1:

$$\mathscr{L}(\theta) = \sum_{i=1}^{n_{patches}} \ell(\mathbf{y}_i, f(\mathbf{x}_i, \theta)). \qquad (1)$$

In equation 1 f is the neural network employed, $\mathbf{y}$ is the ground truth patch (segmentation mask) and ℓ is the per example loss equal to $\ell = \alpha\ell_1 + \beta\ell_2 + \gamma\ell_3$ being ℓ_i with $i \in \{1, 2, 3\}$ losses functions used by the U-Net and Deeplab networks as shown on table 1 and α, β, γ empirically determined scalars.

The use of the dice (Milletari, 2017) and ternaus (Iglovikov et al., 2018) losses are motivated by the IoU (Intersection over Union) metric. This measure is commonly used to measure the quality of semantic segmentation results as shown on figure 1. Since this coefficient is defined over the segmented mask of a discrete set of points it is not differentiable and instead of being employed directly in the optimization, some sort of smooth approximations are used as discussed in (Milletari, 2017; Berman and Blaschko, 2017; Iglovikov et al., 2018).
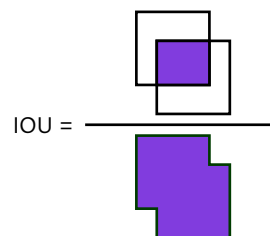

Figure 1: IoU metric ranges from 0 (no intersection) to 1 (perfect intersection).

The problem determined by equation 1 is solved using stochastic optimization techniques (Bottou, 2010). These

methods approximates the gradient of 1 using small subsets (minibatches) of the data allowing a large number of weights updates for what would be the typical cost of only one update of a full batch method. In particular the presented results were trained using the ADAM optimizer (Kingma and Ba, 2014) and a variant of the Stochastic Gradient Descent with momentum (Sutskever et al., 2013).

Two architectures were tested for the presented results. A variation of the U-Net (Iglovikov et al., 2018) that uses as encoder the ResNeXt-50 model Xie et al. (2016) and also a variation of the DeepLab v3+ (Chen et al., 2018) using as encoder a modified ResNet-101 (He et al., 2015) network with output stride of 8. With both encoders being pretrained on the ImageNet dataset (Deng et al., 2009). Despite of the differences between the datasets (natural versus seismic images) there are indications that pretraining on ImageNet is useful albeit less them for similar datasets (Kornblith et al., 2018).

The U-Net architecture initially proposed on (Ronneberger et al., 2015) is widely used on segmentation problems. This architecture can generate features with a rich distribution of semantic meaning and resolution due to the use of multiple blocks that operate at different resolutions. The use of multiple skip connections among the encoder and decoder blocks is knwon to create shortcuts that facilitate the training due to the combination of low level (high resolution) features with semantically richer features. Recently, as discussed previously, there is a trend to use classification networks as an encoder for the U-Net.

The Deeplab v3+ (Chen et al., 2018) is the fourth iteration of the Deeplab family of networks for semantic segmentation. These networks use as encoder a classification network. This network is modified to use an output stride of 8 instead of the usual 32. The segmentation head is composed of one ASPP (Atrous Spatial Pyramid Pooling) block that uses atrous convolution with different dilation rates to capture information at multiple scales. To improve the quality of the segmentation mask the result of the ASPP block is combined with low level features, that have higher resolution. It is important to note that the segmentation result has one quarter of the resolution of the input image.

Methodology

A typical 3d seismic volume has thousands of points per axis. Segmenting a whole section would be challenging due to the limited amount of GPU memory typically available and also requiring a large number of already interpreted sections. Due to these constraints we have opted to use 2D overlapping patches. These patches of size 256×256 were generated from a small amount of seismic sections that were chosen randomly from the 3D volume. This is not an ideal sampling strategy since we could sample lines with labels of lower quality than the average, for instance due to interpolation problems or lines that are too similar to each other.

Two distinct procedures were used to generate the datasets. For the U-Net the patches were generated using an overlapping grid of patches that covers the chosen lines. While for the deeplab model the patches were randomly selected from the chosen lines and also subjected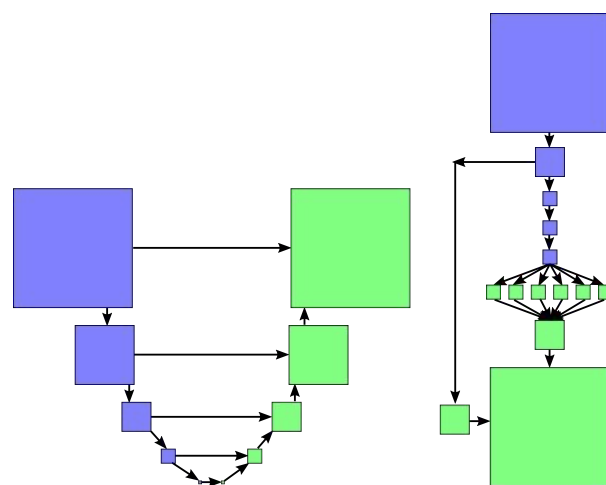 to a data augmentation procedure consisting of affine transformations and elastic distortions (smooth random warping applied to the patches).

Figure 2: Pictoric representation of the U-Net and Deeplab v3+ used to perform segmentation. Blue and green colors identify the encoder and decoder paths, respectively. Square sizes represent the feature map resolution.

Figure 3 ilustrates some of the transforms applied to the patches from the dataset.

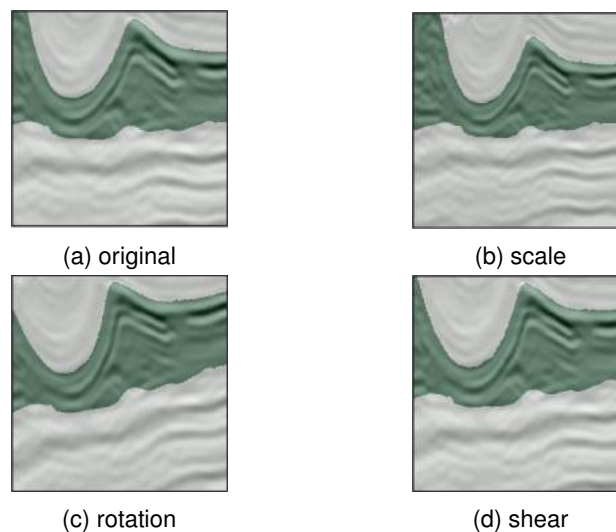(a) original

(b) scale

(c) rotation

(d) shear

Figure 3: Examples of different affine transforms to augment the data training.

The training strategies were similar for both neural networks. Both of them employed schedulings that reduce the learning rate along the training. The losses employed for the deeplab model were slightly different than the ones used for the U-Net as discussed on table 1. The training and prediction were performed in parallel using the Horovod (Sergeev and Balso, 2018) library for distributed synchronous training. The models, input pipeline, training loop were implemented using the Pytorch framework (Paszke et al., 2017).

The prediction cubes were formed by creating predictions for every line being the prediction of one of these lines ob-

tained by weighted stack of the patches that cover the line.

Discussion

The results on figure 6 show that the network prediction results has high coherence with the geology even when using only two lines for training and that adding more lines, rapidly increases the quality of the segmentation results.

It is possible to conclude that that is a diminishing returns tendency. That after some point, adding more lines to be trained does not increase noticeably the quality of the results. Indicating that the newly added data was not informative. It points out that a better estrategy to select new lines should be used.

Figures 7 and 8 show the error maps between the predicted and ground truth horizons for the top and base of salt. The segmentation results for these figures were produced using the Deeplab Model. The central region of the map presents itself with lowest errors, whereas most of the large errors are around the edges of the model.

It is instructive to analyze some of the mistakes shown on figures 7 and 8. For the top of salt (see top graph of figure 7), there is a region of discrepancy between the prediction and the ground truth labeling (shown on figure 4). This region presents a more complex setting. Looking closely, one can conclude that some mistakes are due to the lack of contrast among events which would characterize the salt/non-salt frontier. It is possible that some of those events were not in the lines used during the training of the model. In that way, there could be an iterative work with a specialist that would correct those regions and than retrain the model adding the new corrected labels and line so that the next predictions could better recognize the regions that were previously mistaken.
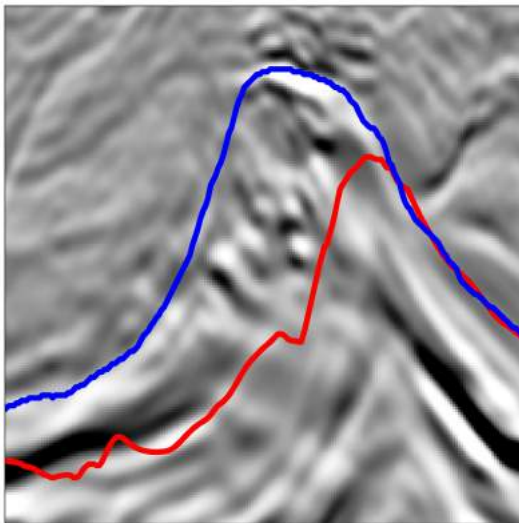

Figure 4: Complex region. Ground truth (red), predicted result (blue).

On the other hand, it is also possible to find places in which the difference between the interpretation and the prediction could actually be a correction of the ground truth label. For instance the region with largest errors on figure 8 actually corresponds to mislabeled horizon as shown on figure 5.
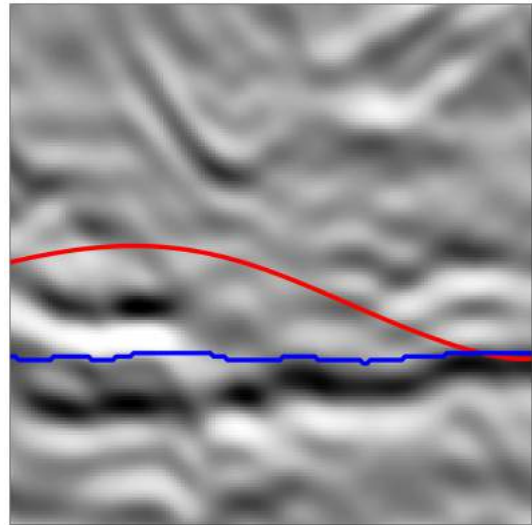

Figure 5: Ground truth (red), predicted result (blue).

It's important to point out that the final decision should be made by the specialist that would conclude whether there was a mistake on his part or not. Following the previously cited iterative process, the specialist could correct the label and then retrain the network so that it would be clearer for the model to learn better patterns without ambiguities.

Conclusion

Convolutional Neural Networks are a promising tool to perform segmentation of seismic datasets. The numerical experiments showed that with a small amount of interpreted lines (less than 1% of the volume) one could obtain reasonable salt segmentation results. Indicating that for some tasks the outlined work could be used to accelerate the construction of horizons.

During the seismic interpretation, the specialist could start with a few lines, for instance two, train the network and obtain a first draft of what could be the salt target in the whole volume. Then, he or she could analyze and correct some of the lines in which the model's prediction were way off of what could be the right event. After that, the model could be fine-tuned with the new corrected lines given as labels. The next prediction of the whole volume should be more accurate than the previous one. That iterative process could be repeated until an acceptable salt prediction were obtained.

Acknowledgments

(a) 2 lines.

(b) 4 lines.

(c) 8 lines.
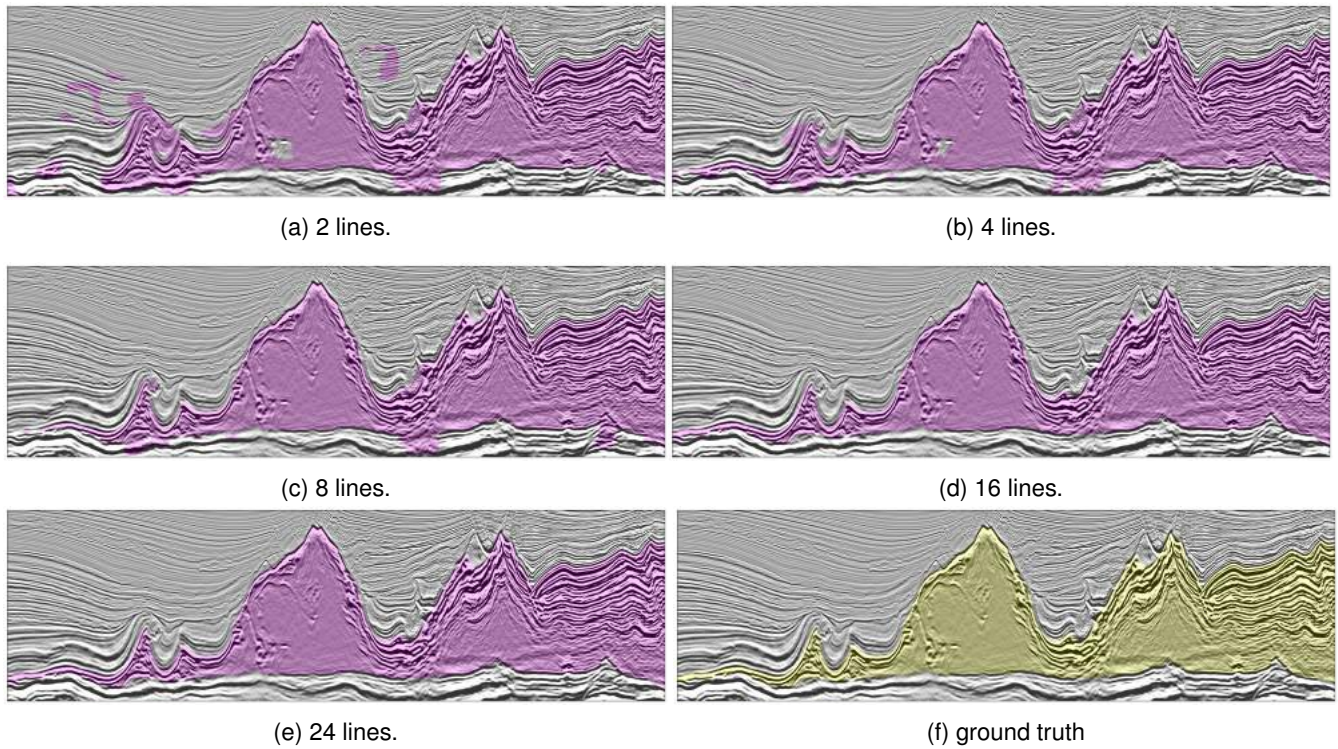
(d) 16 lines.

(e) 24 lines.

(f) ground truth

Figure 6: Segmentation results for different numbers of inlines used to train the network using the U-Net network.
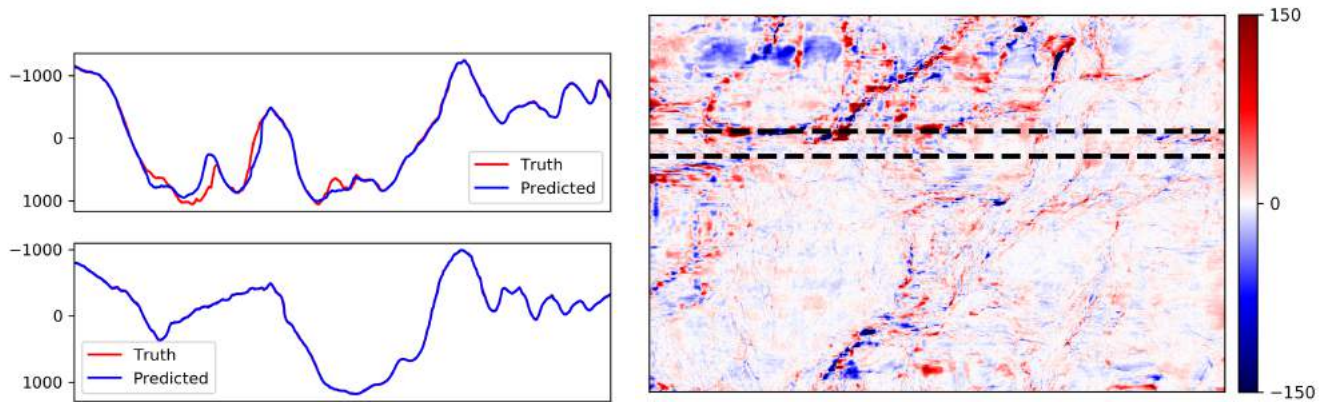

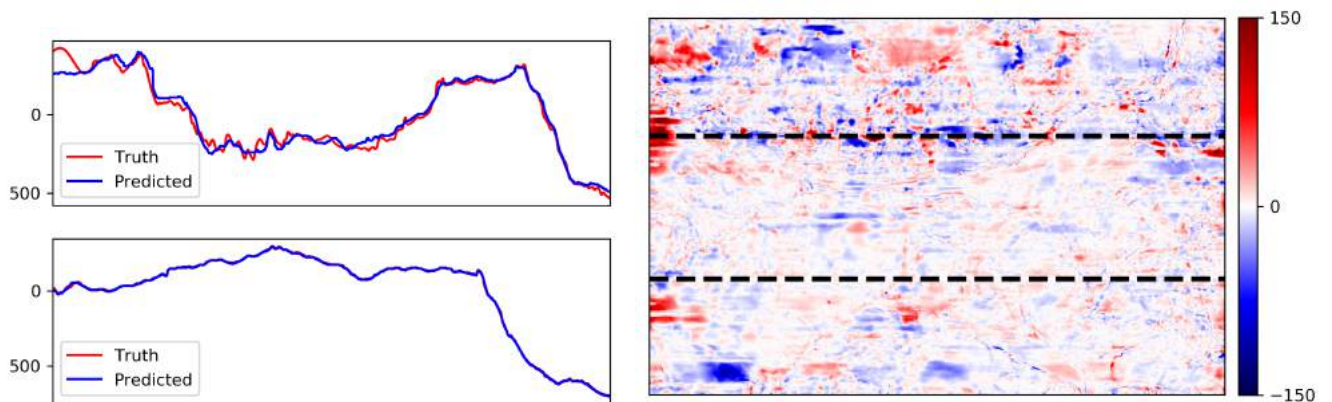Figure 7: Extracted Top of salt horizon for the indicated lines.

Figure 8: Extracted Base of salt horizon for the indicated lines.

References

Berman, M., and M. B. Blaschko, 2017, Optimization of the jaccard index for image segmentation with the lovász hinge: CoRR, **abs/1705.08790**.

Bottou, L., 2010, Large-scale machine learning with stochastic gradient descent: Proceedings of COMPSTAT'2010, Physica-Verlag HD, 177–186.

Chen, L., Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, 2018, Encoder-decoder with atrous separable convolution for semantic image segmentation: CoRR, **abs/1802.02611**.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, 2009, ImageNet: A Large-Scale Hierarchical Image Database: Presented at the CVPR09.

Gramstad, O., and M. Nickel, 2018, *in* Automated interpretation of top and base salt using deep-convolutional networks: 1956–1960.

He, K., X. Zhang, S. Ren, and J. Sun, 2015, Deep residual learning for image recognition: CoRR, **abs/1512.03385**.

Iglovikov, V. I., S. S. Seferbekov, A. V. Buslaev, and A. Shvets, 2018, Ternausnetv2: Fully convolutional network for instance segmentation: CoRR, **abs/1806.00844**.

Karchevskiy, M., I. Ashrapov, and L. Kozinkin, 2018, Automatic salt deposits segmentation: A deep learning approach: CoRR, **abs/1812.01429**.

Kingma, D. P., and J. Ba, 2014, Adam: A method for stochastic optimization: CoRR, **abs/1412.6980**.

Kornblith, S., J. Shlens, and Q. V. Le, 2018, Do better imagenet models transfer better?: CoRR, **abs/1805.08974**.

Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012, Imagenet classification with deep convolutional neural networks, *in* Advances in Neural Information Processing Systems 25: Curran Associates, Inc., 1097–1105.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner, 1998, Gradient-based learning applied to document recognition: Proceedings of the IEEE, 2278–2324.

Lin, T., P. Goyal, R. B. Girshick, K. He, and P. Dollár, 2017, Focal loss for dense object detection: CoRR, **abs/1708.02002**.

Long, J., 2016, Understanding and designing convolutional networks for local recognition problems: PhD thesis, Berkeley.

Long, J., E. Shelhamer, and T. Darrell, 2014, Fully convolutional networks for semantic segmentation: CoRR, **abs/1411.4038**.

Milletari, F., 2017, Hough voting strategies for segmentation, detection and tracking: PhD thesis, TU München.

Paszke, A., S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, 2017, Automatic differentiation in pytorch: Presented at the NIPS-W.

Pereyra, G., G. Tucker, J. Chorowski, L. Kaiser, and G. E. Hinton, 2017, Regularizing neural networks by penalizing confident output distributions: CoRR, **abs/1701.06548**.

Ronneberger, O., P.Fischer, and T. Brox, 2015, U-net: Convolutional networks for biomedical image segmentation: Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, 234–241. ((available on arXiv:1505.04597 [cs.CV])).

Sergeev, A., and M. D. Balso, 2018, Horovod: fast and easy distributed deep learning in tensorflow: CoRR, **abs/1802.05799**.

Sutskever, I., J. Martens, G. Dahl, and G. Hinton, 2013, On the importance of initialization and momentum in deep learning: Proceedings of the 30th International Conference on Machine Learning, PMLR, 1139–1147.

Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, 2015, Rethinking the inception architecture for computer vision: CoRR, **abs/1512.00567**.

Waldeland, A., and A. Solberg, 2017, Salt classification using deep learning: Presented at the EAGE Annual Conference Proceedings.

Waldeland, A. U., A. C. Jensen, L.-J. Gelius, and A. H. S. Solberg, 2018, Convolutional neural networks for automated seismic interpretation: The Leading Edge, **37**, 529–537.

Xie, S., R. B. Girshick, P. Dollár, Z. Tu, and K. He, 2016, Aggregated residual transformations for deep neural networks: CoRR, **abs/1611.05431**.

Zhao, H., J. Shi, X. Qi, X. Wang, and J. Jia, 2016, Pyramid scene parsing network: CoRR, **abs/1612.01105**.