EVALUATION OF UNet AND UNet++ ARCHITECTURES IN HIGH RESOLUTION IMAGE CHANGE DETECTION APPLICATIONS

E. Bousias Alexakis, C. Armenakis
Geomatics Engineering, GeoICT Lab, Department of Earth and Space Science and Engineering, Lassonde School of Engineering, York University, Toronto, Canada - {bousiasa, armenc}@yorku.ca

Commission I, ICWG I/II

KEY WORDS: Change detection, hi-res imagery, Deep Neural Networks, UNet, UNet++

ABSTRACT:

Change detection applications from satellite imagery can be a very useful tool in monitoring human activities and understanding their interaction with the physical environment. In the past few years most of the recent research approaches to automatic change detection have been based on the application of Deep Learning techniques and especially on variations of Convolutional Neural Network architectures due to their great representational capacity and their state-of-the-art performance in visual tasks such as image classification and semantic segmentation. In this work we train and evaluate two CNN architectures, UNet and UNet++, on a change detection task using Very High-Resolution satellite images collected at two different time epochs. We also examine and analyse the effect of two different loss functions, a combination of the Binary Cross Entropy Loss with the Dice Loss, and the Lovász Hinge loss, both of which were specifically designed for semantic segmentation applications. Finally, we experiment with the use of data augmentation as well as deep supervision techniques to evaluate and quantify their contribution in the final classification performance of the different network architectures.

1. INTRODUCTION

The application of a reliable Change Detection (CD) framework can be an invaluable tool in understanding the relationships and interactions between human activities and the physical environment, as well as for map updating and urban monitoring. In this paper we will examine and compare the use of two different Deep Neural Network (DNN) architectures for change detection applications on satellite images collected at two different time periods. The latest and most successful approaches on automatic change detection from satellite images relate mainly to using Deep Convolutional Neural Networks (DCNN) with an encoder-decoder architecture that directly produce a 'Change'-'No Change' label for each pixel of the original image. Such an architecture was first introduced by Fully Convolutional Networks (FCNs) (Long et al., 2015) and since then many successful variants have been proposed including SegNet (Badrinarayanan et al., 2017), UNet (Ronneberger et al., 2015) and UNet++ (Zhou et al., 2018).

In this work we aim to evaluate the use of both UNet and UNet++ architectures for change detection on satellite images and to compare the results produced by training an original UNet architecture on a Change Detection dataset with the results achieved by the UNet++ architecture. A notable advantage of this encoder-decoder approach is that it is an end-to-end approach, meaning that the network outputs the final change map and there is no direct need for any extra manipulation. Thus, the approach is both straightforward as well as very fast in predicting the final maps (especially when run on a GPU). We also evaluate the effectiveness of training using different loss functions as well as the benefits of data augmentation.

UNet (Ronneberger et al., 2015) is an encoder-decoder architecture that consists of two symmetrical paths: a sequence of contractive operations in order to capture context followed by a symmetrical sequence of expansive operations that allow to produce an output with a pixel-based localization. UNet builds on the Fully Convolutional Network (FCN) architecture in an attempt to create a new architecture that will require fewer training examples and will have higher segmentation accuracy. As with the FCN architecture the use of skip connections between layers that refer to the same level of abstraction is the key of the network's effectiveness. One main difference from the traditional FCN architecture is that UNet has incorporated a significantly larger number of channels to the layers of the up-sampling part of the network. This way, the network carries more information from the low resolution to the higher resolution layers.

UNet++ (Zhou et al., 2018), also known as Nested UNet, is based on the UNet architecture. The main idea behind its design is the need to bridge the semantic gap between corresponding convolutional blocks of the encoder and decoder pathways by incorporating intermediate nested convolutional blocks and redesigning the network's structure, by increasing the number of the skip connections between convolutional blocks. The architecture was based on the hypothesis that the optimization problem would become easier if the intermediate feature maps between the encoder and the decoder were semantically similar. Experimental results on tasks related to medical image segmentation support the aforementioned hypothesis as they have shown that UNet++ outperforms the traditional UNet architecture (Zhou et al., 2018).

In both cases, in order to adjust the networks to the CD application we stack all 6 channels of the two instances (RGB_1 and RGB_2) and use the concatenated array as the input to the first convolutional block of each network. Each patch consists of a pair of RGB instances captured at different time periods and a mask of the ground truth changes between the two instances.

In the original UNet paper, Ronneberger et al. (2015) highlighted the importance of using data augmentation in order to increase the generalization accuracy of the CNN by artificially increasing the number of training samples. Following on their reasoning we also explore and quantify the effect of data augmentation in our own application field in order to verify that data augmentation does increase the generalization capacity of the trained network.

2. RELATED WORK- STATE OF THE ART

In the past few years many researchers have experimented with the use of Convolutional Neural Network architectures for Change Detection applications as CNNs have become the de facto approach for image classification tasks ever since the development of AlexNet (Krizhevsky et al., 2012) and the win of the first place in the 2012 ImageNet Large Scale Visual Recognition Challenge. We could distinguish the most recent approaches into patch-based approaches that given an image patch they classify it as changed or unchanged using different variations of CNN architectures and into semantic segmentation approaches that perform semantic segmentation over the entire image. The patch-based approaches address the lack of training data, as a single training image can provide many training patches. On the other hand, patch-based approaches are run in a sliding window manner and they are very slow on inference time and inefficient as the same regions are visited multiple times (there is a great overlap between patches that correspond to adjacent central pixels).

Daudt et al. (2018a) make use of the CNN architectures described by (Zagoruyko & Komodakis, 2015) to detect changes on Sentinel-2 multitemporal instances. Zagoruyko and Komodakis (2015) tested the use of Siamese, pseudo-Siamese and 2-channel networks for estimating the similarity between two different images. The pseudo-Siamese networks are networks that consist of two identical networks like the regular Siamese networks, but those two subnetworks don't share the same parameters and each of them has its own trainable parameters. The 2-channel networks are typical CNN architectures that instead of having a single image as input, they receive a multi-channel image where each channel is a different instance (in the simpler case each image consists of only one channel, but the method can be easily generalized to multi-channel images).

Zhang and Lu (2019) propose a patch-based approach for change detection of multispectral image pairs. The patches are fed to a Siamese network consisting of two identical CNNs that share the same weights. The outputs of the CNNs are unravelled onto a feature vector and the difference between the two feature vectors is used to fuse the results. The fused results are then passed through a Neural Network consisting of two hidden layers that outputs the class ("changed" or "not changed") of the central pixel.

Wiratama et al. (2018) proposed a dual dense CNN architecture for change detection in SAR images. The network consists of two independent convolutional subnetworks and each of them takes as input a 40 by 40 pixel patch from the image pair. The final product of the network is the Euclidean distance between the probability values retrieved from each of the two subnetworks. When the value of the Euclidean distance approaches 1 the central pixel of the patch is classified as change and when it is close to 0 as no change.

Dault et al. (2018b) have trained different variations of Fully Convolutional Networks (FCNs) to predict change maps given two instances of satellite images through semantic segmentation. Besides from a shortened version of the UNet Architecture they have also used a combination of the Siamese and UNet architectures, where on the encoder part of the UNet there are two identical contracting paths. The outputs of the contracting paths are fused and fed to a single expanding path, while there are two different approaches to the use of the skip connections. On the first one each expanding block is provided with two skip connections, one from each identical branch of the corresponding level and on the second one the results of the two parallel contracting blocks are first subtracted and the absolute difference is then passed on the corresponding up-sampling block through a skip connection.

Daudt et al. (2019) present a very high-resolution semantic change detection dataset comprising 291 RGB image pairs accompanied by the corresponding pixel-wise change information and land cover information. They also use multiple UNet architectures to simultaneously predict the landcover of both images of a pair as well as to directly perform change detection given the two original images.

Zhang et al. (2019) use an encoder-decoder architecture based on the Feature Pyramid Network (FPN) and on UNet to perform change detection by performing semantic segmentation on the second instance of the region. The first instance together with an already available GIS map are used for training the network and consequently they use the trained network to retrieve the prediction map for the second instance. Finally, they compare the predictions of the second instance to the GIS map to retrieve the land cover changes.

Lebedev et al. (2018) address the Change Detection problem by utilizing a Conditional Generative Adversarial Network (C-GAN) approach. Conditional GANs learn a mapping from an image and a random noise vector to some vector y. The main idea of a GAN is that we have two subnetworks, the generator G and the discriminator D, which by competing with each other learn the required representation. The discriminator takes as input the two images captured at different time instances, the ground truth changes map and the predicted changes map generated by the generator and classifies it as original or artificial. The generator takes as input solely the concatenated image and produces an artificial image in an attempt to trick the discriminator at classifying the artificially produced image as original. In theory the training process will be complete when the discriminator will classify the input as original or artificial with 50% chance.

Peng et al. (2019) provide a thorough literature review for CD and train a UNet++ network on the dataset created by (Lebedev et al., 2018), which they later compare to other Deep Learning approaches for CD. The training is enhanced by using a deep supervision training scheme and as a loss function the combination of the Binary Cross Entropy loss with Dice coefficient. Both of those suggestions were also introduced in the original UNet++ paper (Zhou et al., 2018). The comparison to other DNN methods for change detection concludes that the UNet++ architecture outperforms other state-of-the-art methods.

3. METHODOLOGY

We build on the work of Peng et al. (2019) with the goal to evaluate the effect of different architectural choices (UNet and UNet++ encoder-decoder architectures) in combination with different loss functions on the performance of the trained networks for change detection applications. In addition, we investigate how the use of deep supervision and data augmentation affects the performance of the various networks. The methodology consists of 4 sub-sections: in section 3.1 we present the two network architectures; in section 3.2 we discuss the different loss functions used to train the networks ; in section 3.3 we introduce the concept of deep supervision; and in section 3.4 we briefly discuss data augmentation.

3.1 Network Architectures

3.1.1 UNet

UNet (Ronneberger et al., 2015) was built based on the idea of the Fully Convolutional Network (FCN) architecture as introduced by Long et al. (2015) with a goal to create a new type of architecture that can be trained using fewer training samples and which can produce higher segmentation accuracy. The main innovation of UNet that differentiates the architecture from FCNs is the introduction of more convolutional filters in the up-sampling path creating in this way an up-sampling (or expanding) subnetwork which is in general terms symmetric to the down-sampling (or contracting) part of the network. Thus, the final network has a U shape form, a fact highlighted by the given name of the architecture. Like the traditional FCN architecture, a very important part of the network is the "skip connections" that pass information from the down-sampling blocks to the corresponding up-sampling convolutional blocks. This flow of information is done by concatenating the high-resolution features of the down-sampling blocks to the first layer of the corresponding up-sampling block. The total number of features are then passed to the convolutional layers and in this way the previously learned contextual information flows into the convolutional layers without losing the localization accuracy caused by the down-sampling of the max pooling steps. This flow of information combined with the larger number of filters in the up-sampling path are the main advantages of the UNet architecture that makes it so successful in the localization accuracy of the segmentation task.

More specifically each down-sampling block consists of 2 successive convolutional layers with a 3 x 3 filter size and a padding of 1 pixel. The first element of each block is the down-sampled result of the previously applied max pooling layer, while in the first block the input consists of the concatenated RGB channels of both image instances. All max pooling layers use a 2x2 kernel shape and a stride of 2. Batch Normalization is applied to each convolutional layer, which was not included in the original paper, but helps the network to learn faster (Ioffe and Szegedy, 2015). The number of feature channels of each convolutional layer of every down-sampled block is doubled with respect to the number of channels in the convolutional layers of the previous level.

With regards to the expanding part of the network the blocks are almost identical to the down-sampling blocks, with the main differences being the input's origin and the extra feature channels coming from the skip connections. The input to each block is the output of the directly previous coarser block in the sequence. The up-sampling operation is performed by using a bi-linear interpolation. The convolutional layers of each expanding block have the same number of feature channels as the convolutional layers of the corresponding contracting block. When applying the skip connection, the features from the contracting block do not need to be cropped as in the original paper since we have padded the convolutional layers and kept the feature width and height constant within each block. An overview of the architecture is presented in Figure 1.

3.1.2 Nested UNet (UNet++)

As mentioned earlier, Nested UNet or also commonly known as UNet++, is an extension of the UNet architecture that was introduced by Zhou et al. (2018) in an attempt to improve the segmentation accuracy of the UNet architecture. The authors maintain the encoder decoder architecture of UNet and argue that a gradual enrichment of the feature maps of higher resolution before aggregating them with the decoder results would help the

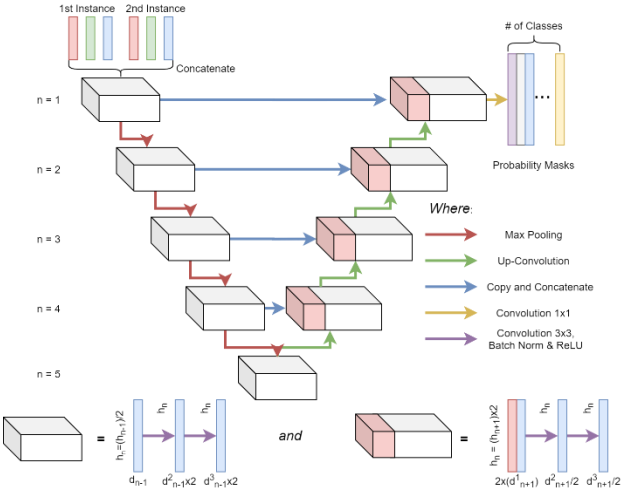

Figure 1. UNet archritecture. The notation h_n refers to the height of the convolutional layers at block level n and d_n^i refers to the number of channels of the i^{th} convolutional layer in level n. The width of each convolutional layer has an identical behavior as the height and for this reason it has been omitted. We can see that the channel depth of a convolutional layer at layer n is twice as large as the corresponding depth of the corresponding convolutional layer of block level n-1.

network capture more high-resolution details thanks to the higher semantic similarity between the concatenated features. The main hypothesis for the introduction of the UNet++ architecture is that by adding more intermediate (nested) convolutional blocks and densifying the skip connections between blocks would cause the concatenated results of each up-sampling block to be more semantically similar than the results obtained by the original UNet architecture, which would ultimately result in an easier optimization problem and thus more accurate results.

Using the same definitions regarding the contracting and expanding blocks as described in the plain UNet architecture and in Figure 1, the general overview of the UNet++ architecture is presented in Figure 2. The nested convolutional blocks $X^{n,m}$ that have been introduced to bridge the semantic gap between the contracting and the expanding blocks of the same level n in the pyramid are connected through skip connections with every convolutional block of the same level n with $m' > m$. More specifically $X^{n,m}$ can be defined as the output of Equ. 1 where $ConvBlock(x)$ is the output of a convolutional block given an input x, $\langle X^{i,j} \rangle_{k=1}^{f}$ is the concatenation operation for elements $X^{i,1}, \dots, X^{i,f}$ and $\langle x, y \rangle$ stands for the concatenation operation of elements x and y.

$$X^{n,m} = \begin{cases} ConvBlock(\langle X^{n-1,m} \rangle), & for\ m = 1 \\ ConvBlock\left(\langle \langle X^{n,k} \rangle_{k=1}^{m-1},\ X^{n+1,m-1} \rangle\right), & for\ m > 1 \end{cases} \quad (1)$$

3.2 Loss Functions

We have used and compared two different loss functions: the first one is a combination of Binary Cross-Entropy loss with the Dice coefficient function (BCE-Dice loss) (Equ. 2), and the second one is the Lovász Hinge loss (Berman et al., 2018), which is a tractable surrogate for the optimization of the intersection over union measure. The combination of the BCE-Dice loss was used in the original UNet++ paper (Zhou et al., 2018) as well as by (Peng et al., 2019) and produced state-of-the-art results.

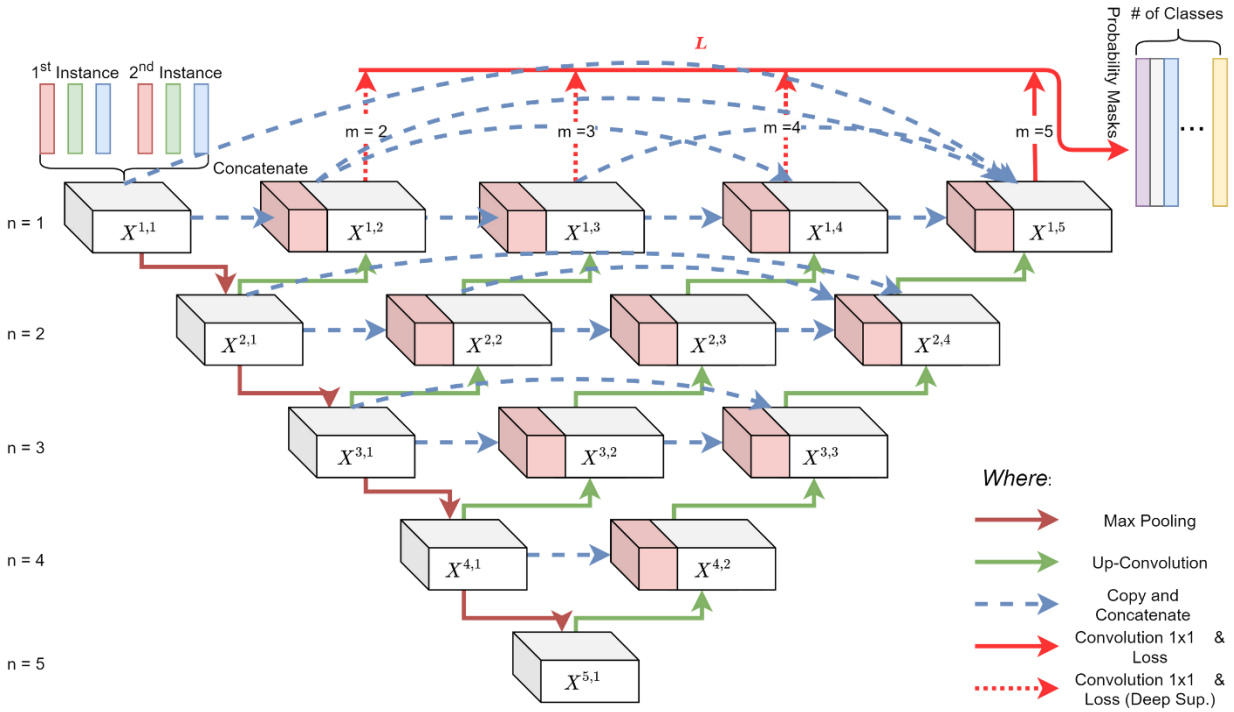

Figure 2: Nested UNet Architecture.

$$L(Y,\hat{Y}) = -\frac{1}{N}\sum_{b=1}^{N}\left(\lambda \cdot Y_b \cdot log(\hat{Y_b}) + \frac{2 \cdot Y_b \cdot \hat{Y_b}}{Y_b + \hat{Y_b}}\right) \quad (2)$$

The parameter λ in Equ. 2 is set to 0.5 as this is the value being used by Zhou et al. (2018) in the original UNet++ paper and it was also experimentally shown to produce the best results on a change detection application (Peng et al., 2019). Y_b and $\hat{Y_b}$ are the flattened ground truth and predicted probability maps respectively of image b and N is the batch size (number of images per batch).

Regarding Lovász Hinge loss, Berman et al. (2018) are introducing a tractable surrogate of the Jaccard index, which is defined as the Intersection over Union $J_C(y,\hat{y})$ (Equ. 3) with the convention that $\frac{0}{0} = 1$. The parameter c is the class value for each pixel and given a set A the operator $|A|$ returns the total number of elements in the set.

$$J_C(y,\hat{y}) = \frac{|\{y=c\} \cap \{\hat{y}=c\}|}{|\{y=c\} \cup \{\hat{y}=c\}|} \quad (3)$$

In our case we are dealing with a binary problem and so the parameter c of the Jaccard index will be equal to 1 (Equ. 4) (we are only considering the foreground object, meaning pixels that are mapped as changed).

$$J_1(y,\hat{y}) = \frac{|\{y=1\} \cap \{\hat{y}=1\}|}{|\{y=1\} \cup \{\hat{y}=1\}|} \quad (4)$$

The corresponding loss is given by (Equ. 5), which is not differentiable as the parameters y and $\hat{y}$ can only take binary values $\in \{-1, 1\}$. Berman et al., (2018) use the Lovász extension of a set function and develop an algorithm that computes an optimization surrogate of (Equ. 5).

$$\Delta_{J_1}(y,\hat{y}) = 1 - J_1(y,\hat{y}) \quad (5)$$

A SoftMax version of the extension has been used by Rakhlin et al. (2018) in combination with the UNet architecture for a land cover classification task and produced promising results.

3.3 Training - Deep Supervision

UNet++ can also be trained using Deep Supervision (Lee et al., 2015), where the overall loss is computed by aggregating the loss of the output layer (the output of the convolutional block $X^{1,5}$after applying a 1x1 convolutional layer) with the "companion"[1] losses of all the intermediate layers of the first level of the pyramid $X^{1,k}$ for $k = 1,..,4$. The companion losses are computed by applying the loss function to each output $X^{1,k}$. The overall loss is then computed as the average of all five losses. In Zhou et al. (2018) the use of deep supervision gave both better and slightly worst results depending on the dataset being used (all datasets related to medical segmentation tasks) compared to using only the output layer loss of the network.

3.4 Data Augmentation

Data augmentation techniques have been shown to improve the performance of similar networks (Ronneberger et al., 2015; Zhou et al., 2018; Peng et al., 2019) especially when training using relatively small datasets and to reduce overfitting to the training data. Thus, we evaluate and quantify the effect of data augmentation on the training of different architectures with different loss functions and in combination to deep supervision. We have augmented the original dataset by occasionally (with a 50% probability) performing a horizontal flip to the input training images, a technique that even though very simple proved to be quite effective as will be shown in the results section.

[1] The term companion loss is used by (Lee et al., 2015)

Design and Training Choices												
Architecture	UNet++	UNet++	UNet++	UNet++	UNet++	UNet++	UNet++	UNet++	UNet	UNet	UNet	UNet
Loss Function	Lovász	Lovász	Lovász	Lovász	BCE Dice	BCE Dice	BCE Dice	BCE Dice	Lovász	BCE Dice	Lovász	BCE Dice
Deep Supervision	✓	×	✓	×	✓	×	✓	×	×	×	×	×
Data Augmentation	×	✓	✓	×	×	✓	✓	×	✓	✓	×	×
Evaluation Metrics												
Precision	0.8504	0.8683	0.8770	0.8602	0.9565	**0.9668**	0.9575	0.9599	0.8455	0.9572	0.8667	0.9512
Recall	0.8807	0.9031	**0.9076**	0.8804	0.8831	0.9034	0.8977	0.8781	0.8967	0.8989	0.8604	0.8656
F1	0.865299	0.885342	0.892046	0.870208	0.918334	**0.934011**	0.926671	0.917212	0.870368	0.927132	0.863575	0.906409
Accuracy	0.9638	0.9680	0.9710	0.9661	0.9842	**0.9870**	0.9856	0.9841	0.9638	0.9858	0.9700	0.9822

Table 1: Results Summary. Evaluation metrics for various choices regarding the architecture of the network, the loss function being used for training and the use of deep supervision and data augmentation during training.

4. EXPERIMENTS

4.1 Test Dataset

The training, validation and test dataset we used was created and introduced by Lebedev et al. (2018). The dataset consists of 10000 training, 3000 validation and 3000 test satellite image pairs and their corresponding change masks. The satellite images were retrieved from Google Earth (GE) and refer to 3 channel RGB images of size 256x256 pixels, whose ground resolution varies from 30cm to up to 100cm per pixel according to the authors. The two instances of each image pair were generally captured during different seasons and the number of changes has been occasionally augmented by the manual addition of objects. The masks consider only changes that correspond to the appearance or disappearance of objects between the two instances of the pair and do not consider any seasonal variations.

4.2 Training Implementation

For the training we have trained all networks for 260 epochs applying in all cases the Adam optimization algorithm using the default parameters for the coefficients of the running average (0.9 and 0.999) and with no weight decay. The batch size consisted of eight images and the learning rates used were set to 0.0003 when training using the BCE-Dice Loss function and 0.0005 which was gradually reduced to 0.0001 when training using the Lovász-Hinge loss function. The training was performed using the pytorch framework on an NVIDIA GeForce GTX 1080Ti GPU. We have built on the UNet++ pytorch implementation available at: https://github.com/4uiiurz1/pytorch-nested-unet .

4.3 Evaluation Metrics

The evaluation metrics that have been used are Precision, Recall, F1 Score and Accuracy (Equs. 6 to 9), where TP (True Positive) is the number of pixels that were correctly classified as changes, TN (True Negative) is the number of pixels that were correctly classified as unchanged, FP (False Positive) is the number of pixels that were classified as changed while they were not actually changed and FN (False Negative) is the number of pixels that were mistakenly classified as unchanged. Change Detection algorithms have to deal with highly unbalanced data with respect to the proportion of changed regions compared to the area that has not been changed. In many cases the changed area can cover less than 5% of a change map, which means that a Network that never detects any change and classifies the whole image as unchanged would score an accuracy higher than 95%. Thus, the measure of accuracy can be widely misleading in a Change Detection task since it does not distinguish between changed and unchanged regions and the measures of Precision, Recall and F1 score represent much more realistic evaluation metrics.

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

$$F1 = 2\frac{Precision \cdot Recall}{Precision+Recall} \quad (8)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

4.4 Results

The evaluation metrics over the test set for UNet and UNet++ architectures trained using either the BCE - Dice Loss or the Lovász Hinge Loss functions are summarized in Table 1. The table also presents the perceived changes of the performance on the test set caused by the incorporation of data augmentation and deep supervision (when applicable) during training. The comparison of the results suggests that the UNet++ architecture when trained using the BCE-Dice Loss function with data augmentation produces the best results over the test set. Figures 3 to 8 illustrate the effects of the different design and training choices on the performance of each trained network on the test set.

Figure 3 shows the effect of data augmentation and deep supervision on the prediction of the UNet++ architecture trained using the Lovász Hinge loss function. The use of data augmentation has a positive effect on all the metrics, while the use of solely deep supervision has a negative effect on Precision (of about 1%) and on F1 score. However, the best results were obtained when using both deep supervision and data augmentation. Similarly, Figure 4 presents the effect of data augmentation and deep supervision on the prediction of the UNet++ architecture trained using the combination of the Binary Cross Entropy loss with the Dice loss (BCE-Dice loss) function. The results suggest that the use of deep supervision and of data augmentation produce higher metric values than the base

scenario, where none of the two techniques was used, with the use of data augmentation having a greater positive effect. The best results for all metrics are retrieved when applying only data augmentation (and no deep supervision) when training the network.

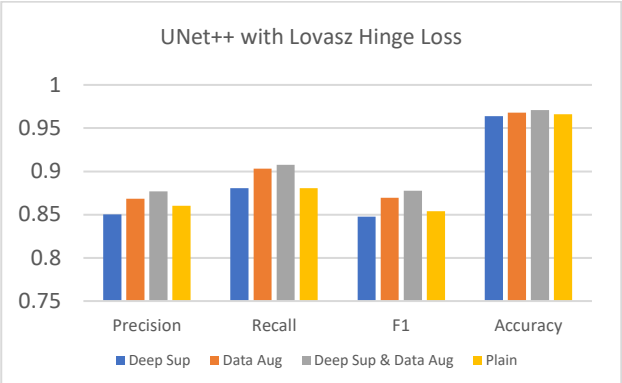

Figure 3: Effect of Data Augmentation (Data Aug) and Deep Supervision (Deep Sup) on UNet++ architecture when training using the Lovász Hinge Loss.

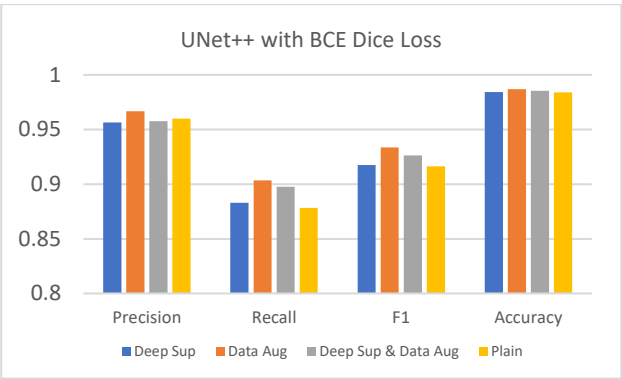

Figure 4: Effect of Data Augmentation (Data Aug) and Deep Supervision (Deep Sup) on UNet++ architecture when training using the BCE Dice Loss.

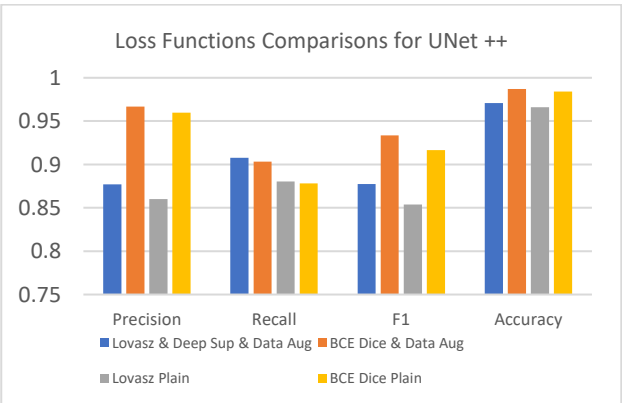

Figure 5: Comparison of the effects of using different loss functions on the UNet++ test results.

Figure 5 illustrates a direct comparison of the retrieved metrics when using different loss functions on the same UNet++ architecture. The combination of BCE-Dice coefficient achieves better precision (by more than 8%) and slightly worst recall than the use of the Lovász Hinge loss. This means that the network trained with the BCE Dice loss produces less pixels classified falsely as changed, but at the same time it detects slightly fewer changes than the network trained with the Lovász Hinge loss. The F1 score, as the geometric mean of the precision and recall rates, is 7% higher for the BCE Dice loss. Likewise, the comparison of the effects of the different loss functions on the UNet architecture is presented in Figure 6, where the effects are similar to the ones derived from Figure 5 and the precision gap is even larger (higher than 11%).

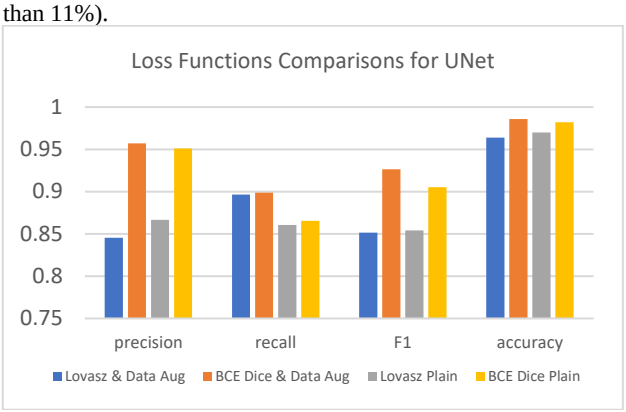

Figure 6: Comparison of the effects of using different loss functions on the UNet test results

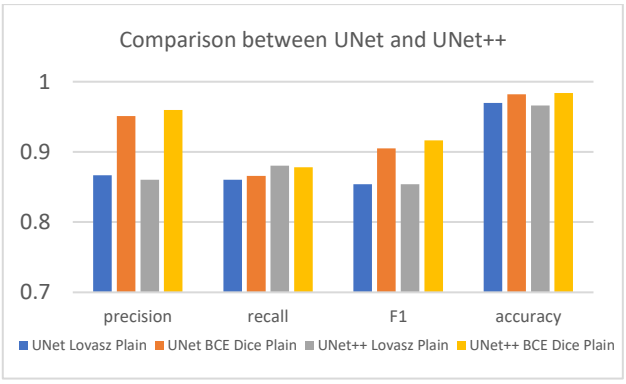

Figure 7: Comparison of the effects of different architectures and different loss functions.

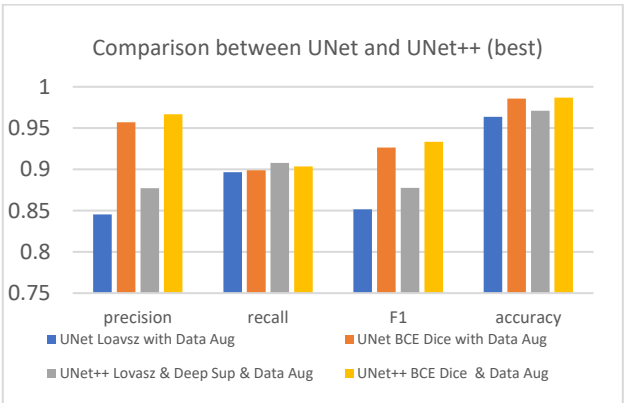

Figure 8: Comparison of the best results achieved with each architecture and each loss function.

The effects of using different architectures and different loss functions without using data augmentation nor deep supervision are presented in Figure 7. The choice of the loss function seems to affect the performance of the network more than the choice of UNet or UNet++ architecture with the BCE Dice loss yielding higher metric values. Also, the use of the UNet++ architecture produces slightly better results than UNet given the same loss function, with the differences being less than 1% for all metrics.

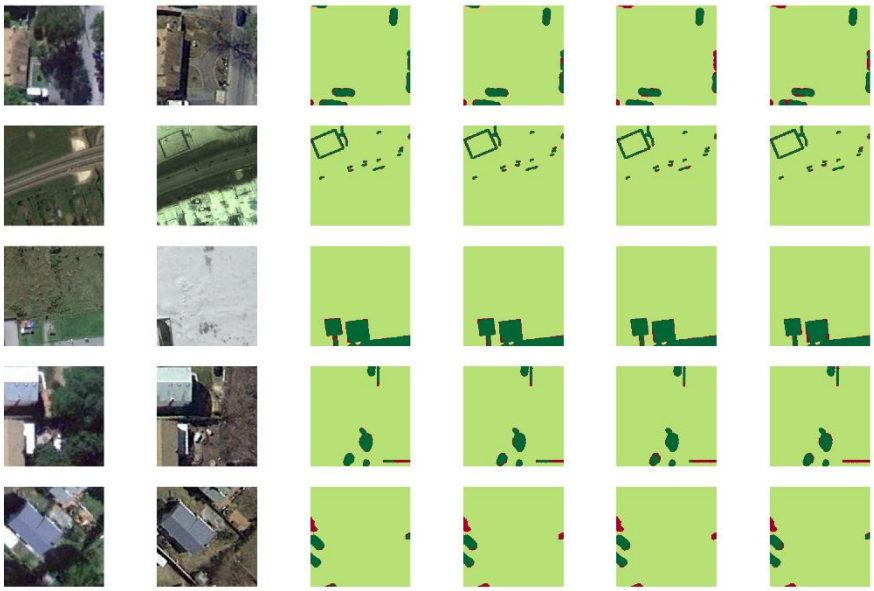

Figure 9: Examples of change masks for various models. The color scale is the same as in Figure 9. From left to right the models are: (a) the UNet++ trained with BCE Dice loss and data augmentation, (b) the UNet++ trained with Lovasz Hinge loss deep supervision and data augmentation, (c) the UNet trained with BCE Dice loss and data augmentation and (d) the UNet trained with Lovasz Hinge loss and data augmentation.

Similarly, the comparison of the best results achieved for each architecture and each loss function are shown in Figure 8. Once again the choice of the loss function seems to affect the results more than the choice of architecture with the best results being produced by the UNet++ trained with BCE Dice Loss and data augmentation followed closely (less than 1% difference on every metric) by the UNet network trained with BCE Dice loss and data augmentation.

An example of change detection prediction using UNet++ is shown in Figure 10. A set of exemplary results retrieved from different combinations of architectures, loss functions and the use of data augmentation and deep supervision are presented in Figure 9. By visually examining the results, TP (change) and TN (no change), we can argue that all networks perform reasonably well on the CD task and can learn to ignore seasonal effects like snow and seasonal vegetation changes.

Figure 10: Change Detection Prediction Example Using UNet++.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have experimented with two encoder-decoder CNN architectures for change detection applications using high resolution satellite images. We have also compared two different loss functions for the training of the CNNs and evaluated the contribution of data augmentation and deep supervision techniques on the performance of the networks. All networks produced state-of-the-art results with the network using the UNet++ architecture and being trained with the BCE Dice Loss and data augmentation performing the best on the test data. Future work will involve the testing of similar architectures on different datasets as well as the evaluation of the robustness of the semantic segmentation results in the presence of higher misregistration errors.

ACKNOWLEDGEMENTS

This work is financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC Discovery and CREATE grants) and York University.

REFERENCES

Badrinarayanan, V., Kendall, Alex, & Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Berman, M., Rannen Triki, A., & Blaschko, M. B. (2018). The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4413–4421.

Daudt, R. C., Saux, B. L., Boulch, A., & Gousseau, Y. (2018a). Urban Change Detection for Multispectral Earth Observation Using Convolutional Neural Networks. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*. https://doi.org/10.1109/IGARSS.2018.8518015

Daudt, R.C, Le Saux, B., & Boulch, A. (2018b). Fully Convolutional Siamese Networks for Change Detection. *IEEE International Conference on Image Processing (ICIP)*.

Daudt R., C., Rodrigo, Le Saux, B., Boulch, A., & Gousseau, Y. (2019). Multitask learning for large-scale semantic change detection. *Computer Vision and Image Understanding*, *187*, 102783. https://doi.org/10.1016/j.cviu.2019.07.003

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *ArXiv:1502.03167 [Cs]*. http://arxiv.org/abs/1502.03167

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, 1097–1105.

Lebedev, M. A., Vizilter, Y. V., Vygolov, O. V., Knyaz, V. A., & Rubis, A. Y. (2018). Change Detection in Remote Sensing Images Using Conditional Adversarial Networks. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLII–2*, 565–571. https://doi.org/10.5194/isprs-archives-XLII-2-565-2018

Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z., & Tu, Z. (2015). Deeply-Supervised Nets. *Artificial Intelligence and Statistics*, 562–570. http://proceedings.mlr.press/v38/lee15a.html

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965

Peng, D., Zhang, M., & Wanbing, G. (2019). End-to-End Change Detection for High Resolution Satellite Images Using Improved UNet++. *Remote Sensing*, *11*, 1382. https://doi.org/10.3390/rs11111382

Rakhlin, A., Davydow, A., & Nikolenko, S. (2018). *Land Cover Classification From Satellite Imagery With U-Net and Lovász-Softmax Loss*. 262–266. http://openaccess.thecvf.com/content_cvpr_2018_workshops/w4/html/Rakhlin_Land_Cover_Classification_CVPR_2018_paper.html

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (pp. 234–241). Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28

Wiratama, W., Lee, J., Park, S.-E., & Sim, D. (2018). Dual-Dense Convolution Network for Change Detection of High-Resolution Panchromatic Imagery. *Applied Sciences*, *8*(10), 1785. https://doi.org/10.3390/app8101785

Zagoruyko, S., & Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4353–4361. https://doi.org/10.1109/CVPR.2015.7299064

Zhang, C., Wei, S., Ji, S., & Lu, M. (2019). Detecting Large-Scale Urban Land Cover Changes from Very High Resolution Remote Sensing Images Using CNN-Based Classification. *ISPRS International Journal of Geo-Information*, *8*(4), 189. https://doi.org/10.3390/ijgi8040189

Zhang, W., and Lu, X. (2019). The Spectral-Spatial Joint Learning for Change Detection in Multispectral Imagery. *Remote Sensing*, *11*(3), 240. https://doi.org/10.3390/rs11030240

Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., & Liang, J. (2018). UNet++: A Nested U-Net Architecture for Medical Image Segmentation. In D. Stoyanov, Z. Taylor, G. Carneiro, T. Syeda-Mahmood, A. Martel, L. Maier-Hein, J. M. R. S. Tavares, A. Bradley, J. P. Papa, V. Belagiannis, J. C. Nascimento, Z. Lu, S. Conjeti, M. Moradi, H. Greenspan, & A. Madabhushi (Eds.), *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support* (pp. 3–11). Springer International Publishing. https://doi.org/10.1007/978-3-030-00889-5_1