A Study of Deep Learning Model Performance for Remote Sensing Image Segmentation

Teodora Selea, Gabriel Iuhasz and Marian Neagul

***Abstract*—Deep Learning is an extremely important research topic in Earth Observation. Current use-cases range from semantic image segmentation, object detection to more common problems found in computer vision such as object identification. Earth Observation is an excellent source for different types of problems and data for Machine Learning in general and Deep Learning in particular. It can be argued that both Earth Observation and Deep Learning as fields of research will benefit greatly from this recent trend of research.**

In this paper we take several state of the art Deep Learning network topologies and provide a detailed analysis of their performance for semantic image segmentation for building footprint detection. The dataset used is comprised of high resolution images depicting urban scenes. We focused on single model performance on simple RGB images.

In most situations several methods have been applied to increase the accuracy of prediction when using deep learning such as ensembling, alternating between optimisers during training and using pretrained weights to bootstrap new models. These methods although effective, are not indicative of single model performance. Instead, in this paper, we present different topology variations of these state of the art topologies and study how these variations effect both training convergence and out of sample, single model, performance.

***Index Terms*—Deep learning; convolutional neural networks; remote sensing**

I. Introduction

Ever since the breakthrough of AlexNet [1] at the ImageNet classification challenge, Convolutional neural networks (CNN) dominate the field of computer vision. Since then, several new CNN architectures appeared, like VGG [2], ResNet [3] or Inception [4] that serve as base models for the further development of CNN topologies. Current computer vision tasks include image classification, semantic segmentation and instance segmentation. In image classification, one class label is attached to every image. As opposed to this, in semantic segmentation we aim to classify each individual pixel from an image and therefore identify pixels that belong to the same object.

The continuously increasing volume of acquired remote sensing data, generated the need of an automatic method for information extraction. This is also true in the case of semantic image segmentation. Valuable insight may be obtained, regarding land cover, urban planning, geographic mapping, change detection, etc. In this paper, we focus on the problem of semantic segmentation applied on satellite or aerial imagery. We analyze and compare different semantic segmentation architectures and their behaviour on satellite image data.

In this article we focus on single model predictive performance. We compare 4 state-of-the-art deep learning topologies used for semantic image segmentation. Our goal is to modify each individual topology so that we maximize the overall predictive performance. We also provide an implementation for our best performing models, along with their trained weights. In section II we present an overview of the current state of the art for the semantic image segmentation domain, focusing on the aforementioned topologies. In section III we start by describing the datasets and our experimental methodology related to data ingestion and training parameters. Section IV contains details of the topological modification of the deep learning models as well as the results of each experiment. Finally, section V contains our conclusions and plans for future work.

II. Earth Observation and Deep Learning

The increasing interest on deep learning techniques applied to remote sensing data may be seen in the amount of publication on the topic from the past few years [5].

Semantic segmentation task was also the topic of several earth observation challenges on Kaggle[1], CrowdAI[2], Spacenet[3][4] [6], Deepglobe [7], ISPRS [8]. The labeled datasets released by these competitions vary, starting from the objects aimed to be identified to the types of data provided (RGB, IRRG, RGBIR, DSM). In this paper, we focus on the building detection problem, using only RGB data, provided by the Urban 3D [9] challenge, due to the considerable size of the training and testing sets. For the testing phase our results are also validated against the ISPRS Potsdam dataset.

Semantic segmentation in Earth Observation

Fully Convolutional Networks (FCN) [10] were designed to provide a fast and accurate solution to the problem of semantic segmentation. They are based on state of the art classification networks that provide a dense pixel prediction. They build on the idea of replacing all dense, fully connected layers with convolutional layers, adding a 1×1 convolution with the channel dimension equal to the number of classes to predict. Finally, a deconvolutional layer (backwards convolution) is appended to upsample the result to its initial size. The

T. Selea, G. Iuhasz and M. Neagul are with Institute e-Austria, Timisoara, Romania and with Faculty of Mathematics and Informatics, West University of Timisoara, Romania

[1]https://www.kaggle.com/c/planet-understanding-the-amazon-from-space
[2]https://www.crowdai.org/challenges/mapping-challenge
[3]https://spacenetchallenge.github.io/Competitions/Competition2.html
[4]https://spacenetchallenge.github.io/datasets/datasetHomePage.html

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 November 2022 doi:10.20944/preprints202211.0226.v1

authors also introduce skip connections, as a way to refine the prediction. Skip connections enable feature fusion, combining information extracted by the lower layers using finer strides with the features extracted by the final coarse layers.

U-Net [11] is a semantic segmentation network based on FCN and upon the principle of skip connections and feature fusion. Another distinction when compared with FCN, is the increased number of feature channels in the upsampling part. The network is composed of a contractive part and an expansive part, having a similar number of feature channels in both parts, therefore resulting in a "U" shaped network. The feature extraction part downsamples the data and doubles the number of channels at each step. The upsampling divides the number of channels by a factor of two. Skip connections are made between each convolution block that extracted the features and the corresponding upsampling layer, fusing features at different resolutions, just like in FCN. The network uses concatenation to merge together the corresponding features.Although initially designed for bio-medical image segmentation, U-Net was successfully used on remote sensing images.

In [12] the authors separately trained a U-Net model for identifying each class from their dataset, in combination with augmenting techniques and reflectance indices. U-net architecture was also used by the winner of the Kaggle DSTL[5] competition.

W-net [13] is a medical image segmentation network, build from two U-nets, put together side by side. The authors investigate the use of activation function and different methods of bridging the two U-Net. One particularity of W-net is the mixing between two different activation functions ReLU and ELU, that provides better results as opposed to using only one of the activation functions. ELU activation is mainly used on the edges of the "U" shaped network and as it gets saturated when the network goes deeper, it is replaced by ReLU in the middle part of the network. The authors also analyze concatenation versus addition as bridging methods and conclude in favor of concatenation.

Segnet [14] is another popular semantic segmentation network, originally designed for road scene understanding. Just like U-Net, Segnet has an encoder-decoder structure, one for extracting features and the other for re-sampling the image to its original size. The encoding part of Segnet is based on the VGG [2] network, by removing the fully connected layers from the end of the network. Segnet's main contribution relies in the upsampling part of the network. The authors use knowledge from the encoding layers to refine the upsampling, but instead of transferring the whole feature maps as in U-Net, they propose to only use the pooling indices. By doing so, Segnet obtains a gain in the memory usage. Just like U-Net, Segnet also includes convolutional layers in the decoder part, in order to make the features maps obtained by the upsampling layers denser.

Bayesian Segnet [15] is an enhanced version of the original Segnet, where the authors improved the network by adding dropout [16] as a regularization technique. The authors investigated different configurations with the dropout layer: after every convolutional layer, only after the encoder/decoder blocks or combined, settling for a configuration using a dropout probability of 0.5 inserted after the max-pooling layer after the three innermost encoder blocks.

Segnet was used on the ISPRS dataset [17], in combination with an edge detection network (HED) [18]. Two networks are trained individually, one with color channels and the other with DEM data and their results are fused by concatenation at the end. The authors introduce HED in the beginning part of the training process, to extract boundary information of the objects from the input images. Then, the edge information is appended as an addition channel to both semantic segmentation models. An improvement is also obtained by using multi-scale training and use of ensemble models. Apart from Segnet, a FCN network [19] is used in the process, also trained with boundary information. In the end, the model results are averaged before the final prediction.

HSN [20] is a lightweight network, designed on the encoder-decoder principles, with few trainable parameters. The network was developed for semantic segmentation on satellite images, for recognizing objects like buildings, trees, low vegetation and cars. HSN combines simple convolutional layers with residual blocks [3], deconvolution and inception modules [4]. The Inception modules allow multi-scale inference and are used both in the encoder and decoder part of the network. Skip connections are also used to transfer knowledge towards the upsampling layers, but (opposed to the previously mentioned networks) the feature maps are first passed through residual blocks and then fused in the decoding step. Deconvolutional layers are used to gradually upsample the feature maps, followed by simple convolutional layers, similar to other encoder-decoder networks.

An accuracy gain may be obtained not only by a careful choice of activation functions, but also from using regularization techniques like batch normalization [21] or Dropout [16]. In [22] the authors analyze the interaction of batch normalization layers among activation layers for classification networks trained on ImageNet dataset. The authors recommend using ELU without batch normalization or ReLU with batch normalization technique. The position of batch normalization before or after activation is also discussed, however, the results are inconsistent and therefore we aim to study the behavior of semantic segmentation networks in both cases.

Earth Observation segmentation data sets

Data sets for semantic image segmentation are gaining popularity being almost on par with regards to quantity as the ones for image classification. This trend has also been observed in the case of data sets created for semantic image segmentation meant for satellite imagery.

The Dstl[6] satellite imagery feature detection data set (segmentation) provides $1km \times 1km$ scenes in both 3 bands (RGB) and 16 bands (multispectral and shortwave infrared). The RGB resolution is $0.31m$ while the multispectral and shortwave infrared is $1.24m$ and $7.5m$ respectively. The ground truth

[5]https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection

[6]https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection/data

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 November 2022 doi:10.20944/preprints202211.0226.v1

contains 10 classes of man made objects (ranging from building to small an large vehicles) to naturally occurring objects such as waterways and trees. The total data set size is 20GB (GeoTiff lossless format) and the evaluation for the kaggle competition used the average Jaccard index.

The SpaceNet[7] data sets are semantic segmentation data sets for both building as well as roads. The area of interest covers various urban areas (Rio, Vegas, Paris, Shanghai and Khartoum) and is comprised of 3 band RGB and 8 band multispectral images provided by DigitalGlobe. The spatial resolution ranges from $30cm$ to $50cm$ the scene size also varies based on the type of segmentation it being approximately $400m \times 400m$ for road segmentation and $200m \times 200m$ for building segmentation. The total size of the available data sets combining all areas is over 110GB.

Another interesting dataset from SpaceNet is the so called SpaceNet Off-Nadir[8] data set. It contains 120.000 building footprints in 27 $450m \times 450m$ scenes. These scenes are split up into 3 categories based on their nadir value defined as: nadir $0-25$ degrees, off-nadir $26-40$ degrees and very off-nadir $40-55$ degrees. The goal of this data set was to see how well off-nadir images can be processed using current state of the art methods in earth observation. Total size of the data set is approximately $186GB$.

The Inria aerial image labeling data set[9] contains 180 color images covering $1500m \times 1500m$ each at $30cm$ resolution. There are 36 tiles in total for several regions (Austin, Chicago, Kitsap County, Western Tyrol, Vienna).

In the case of land and crop cover segmentation data sets we also have several valuable data sets. Land Cover data set of Slovenia[10] created by Sinergise comprised of a temporal stack of Sentinel 2 hyperspectral images at $10m$ resolution. The ground truth contains 10 classes, the total size of the data set is approximately $187GB$. The agricultural crop cover challenge[11] uses Landsat 8 images at a resolution of $30m$ to classify corn and soybean coverage. There is a temporal stack of images covering a period of 6 years for a total of 32 scenes (25% used for validation).

III. Experiments

As we have discussed in the previous section there are a wide range of annotated earth observation datasets geared towards the semantic segmentation problem. Most of the current work being done is to maximize by any means necessary the accuracy score achieved. This is mostly done using several models and/or transfer learning methods. For the experiments we have chosen the dataset from the Urban3D challenge. In this section we discuss the details of our experiments.

Urban 3D Dataset: The USSOCOM Urban3D Challenge [9] (sample in Figure 1)released a large-scale remote sensing dataset, containing orthorectified 2D data and 3D Digital Surface Model images, with annotated building footprints. The goal of this competition was to improve automated building footprint extraction from satellite imagery. Reducing the currently significant manual effort for obtaining high geospatial accuracy.

The dataset is composed of 174 training scenes and 62 testing scenes, each of 2048×2048 pixel resolution, obtained at approximately 0.5 meters ground sample distance. The dataset may be downloaded from the Spacenet repository mentioned in section II which contains images from Jacksonville, Florida, USA; Tampa, Florida, USA; and Richmond, Virginia, USA totaling approximately $157,000$ building footprints.

During the initial competition the question of the ground truth accuracy was raised. Some of the provided scene ground truths had missing or superfluous building footprints. Initial estimates put the corruption of the ground truth to as high as 5-10%. However, it has been found that the error is closer to 1-2% at worst in the training set while the testing set ground truth has been rectified.

ISPRS Potsdam Dataset: The ISPRS Potsdam dataset [8] is a benchmark dataset released by ISPRS, consisting of orthorectified 2D data and a 3D Digital Surface Model. The 2d is composed of 4 different channels: Red (R), Greed (G), Blue (B) and Infra Red (IR).

The dataset is composed out of 38 patches, based on a ground sampling distance of $5cm$.

Scoring: The scores presented in this paper are computed using the $F1\ score$ as seen in equation 2 which is based on the true positive (TP), true negative (TN), false positive (FP), false negative (FN). Our experiments are based solely on the provided RGB imagery.

$$Precision = \frac{TP}{TP+FP} \quad Recall = \frac{TP}{TP+FN} \tag{1}$$

$$F1\ score = 2 * \frac{Precision * Recall}{(Precision + Recall)} \tag{2}$$

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e} \tag{3}$$

$$p_o = \frac{TP+TN}{N} \tag{4} \quad p_e = \frac{1}{N^2} \sum_C n_{ki} \tag{5}$$

The Cohen Kappa coefficient is a statistical score which measures the inter-agreement. It takes into account also the possibility of the agreement occurring by chance. We can see that in equation 3 p_o represents the observed agreement (accuracy, equation 4), while p_e (equation 5) represents the hypothetical probability of chance agreements, where N is the total number of samples, C number of classes and n_{ki} is the number of times rater i predicted class C. Although we calculate this coefficient we use the F1 score as the deciding factor for the best performing models.

Training settings: Like many other such challenges, the Urban3D competition imposed no limitation on model size or the number of models that contribute to a prediction. This of course is not in itself a bad thing however, in our view it is not representative of the performance of individual models.

[7] https://spacenetchallenge.github.io/datasets/datasetHomePage.html
[8] https://spacenetchallenge.github.io/datasets/spacenet-OffNadir-summary.html
[9] https://project.inria.fr/aerialimagelabeling/contest/
[10] http://eo-learn.sentinel-hub.com/
[11] https://www.crowdanalytix.com/contests/agricultural-crop-cover-classification-challenge

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 November 2022 doi:10.20944/preprints202211.0226.v1

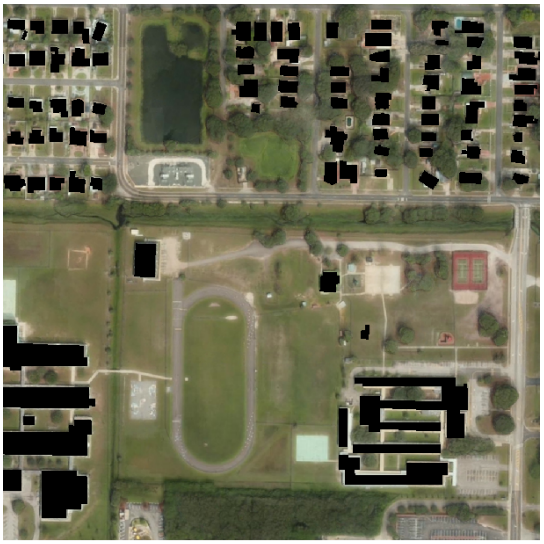


Fig. 1. Urban3D Sample of RGB image and its associated mask

Furthermore, it can lead to a *pay-to-win* type scenario where the team that has access to the more powerful infrastructure can train a variety of predictive models creating at the end an ensemble prediction.

Other solutions used a variety of training methodologies. For example training with a certain learning rate for a number of epochs after which the resulting model is retrained with a different learning rate. Another variation on this training method is that instead of using the more classic gradual deprecation of the learning rate it is made to fluctuate from a high value to a low value and than back for a fixed number of epochs.

In our experiments we wanted to highlight the performance of individual models as is, without relying on stacking, ensembling, transfer learning or retraining of models. The experiments have been implemented using the Keras[12] deep learning library, using the TensorFlow[13] backend running on NVidia Tesla V100-SXM2 (16GB RAM) hosted on IBM® Power® System AC922.

The training method chosen is Adam [23]. It is a method that uses an adaptive learning rate. It stores an exponentially decaying average of past squared gradient v_t as well as the exponentially decaying average of past gradient m_t, which is similar to momentum (i.e. first and second momentum).

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (6)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (7)$$

We can see from equations 6 and 7 that both m_t and v_t can be biased towards 0 when the decay rates β_1 and $\beta 2$ are close to 1. The biased corrected first and second momentum are:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (8) \qquad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (9)$$

The biased corrected gradient calculations from 8 and 9 are then used to update the parameters Θ as follows:

$$\Theta_{t+1} = \Theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \quad (10)$$

In [23] the authors propose the default values of 0.9 for β_1, 0.999 for β_2 and 10^{-8} for ϵ. We used a starting learning rate η of 0.0001 leaving all of the other parameters unchanged. In order to facilitate the convergence of the models we also utilized a method that reduces the learning rate. It checks after each epoch if the validation loss has decreased. If it finds that after 3 epochs the loss stagnates or is increasing it will decrease the learning rate by a factor of 0.1 from the current value.

To prevent overfitting we used early stopping based on the validation loss. Training was stopped when the model validation loss was decreasing for 10 consecutive epochs.

Considering the size of the input images (2048×2048) and the memory limitations of the accelerator hardware, we are using an sliding window mechanism (as depicted in Figure 2) where each of the input scenes is divided in multiple, partially overlapping, tiles of reduced size (256×256). We chose to overlap the tiles (using a fixed stride size) in an attempt to compensate for dimensionality reduction induced by splitting the input.

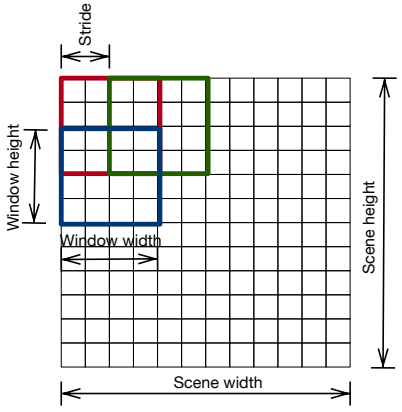


Fig. 2. Scene Tiling Example

For validation purposes the best performing models are trained again using the ISPRS Potsdam dataset, presented in Section III, using the same training methodology, particularly we use the same window size and stride. This measure was taken as a further testing of topology performance on an additional benchmark dataset.

It is important to mention that in our setup we discard the fact that the two datasets have a different spatial resolution, situation that could be partially mitigated by data augmentation (random re-sampling at training time)

For our experiments we split the training data into training and validation sets. The validation dataset represents 20% of the total available number of tiles. We use as a holdout set of the entirety of the original scoring dataset from the Urban3D challenge totaling 62 scenes. This was done to gauge the out of sample performance of the trained models.

As stated before our desire is to test single model predictive performance on RGB images only. For this we have selected a representative set of the current Deep learning network topologies currently used. We investigated different skip connections

[12]https://keras.io/
[13]https://www.tensorflow.org

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 November 2022 doi:10.20944/preprints202211.0226.v1

for *HSN* (addition and concatenation), the placement of activation functions relative to convolution and batch normalization layers from the network topologies. We also tested different activation functions inside the same topology, combining the *ReLU* (eq 11) and *ELU* (eq 12) activation.

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \tag{11}$$

$$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \tag{12}$$

The final stage of our experiments was the cross validation of the best performing models obtained. This was done both on the Urban3D and ISPRS Potsdam dataset.

We selected them based on their validation F1 score. We used 5 random folds for each model, aiming to check the predictive performance on the different training and validation splits. In the following paragraphs we detail each of the model topologies used.

Validation

In order to ensure that our best performing models selected thus far have good out of sample performance we ran 5 random sub-sampling validation experiments (denoted s1, s2, s3, s4, s5), where we randomly selected 20% of the training dataset as validation. This allows us to check for bias in the original training validation split. We have chosen this type of validation as the proportion of the training/validation split is not dependent on the number of iterations (folds) in the case of K-Fold cross validation.

IV. Results

The following section contain an in depth analysis of the topological modifications made to the selected models as well as our experimental results.

A. Concatenation and addition in HSN

U-Net and W-Net are designed to support feature fusion extracted from the encoder blocks to the decoder parts of the networks. Both in [13] and in [24] the authors analyze the usage of skip connections with concatenation for feature fusion, as opposed to common addition skip connection and they notice an improvement in their model performances. In the proposed topology, [20] uses common skip connection to transfer information from lower layers to upper layers. In this paper, we investigated this approach on the HSN network, changing the fusion method from addition to concatenation.

The HSN model with concatenation (*hsnv1*) has proven to be more stable during the training process compared to the original HSN model (*hsnv2*), with addition as feature fusion method, as depicted in Figure 3a (where the blue crosses represent the outliers, the orange line the median value and the box itself 95% of the values). In Figure 3b we plot the best epoch of both HSN models, using concatenation or addition as fusion method, considering the validation F1 score.

For the majority of the samples, the model using concatenation (*hsnv1 Concat*) as feature fusion method reaches the best F1-Score earlier, potentially starting to overfit the training data.

Interestingly, *hsnv1 Concat* also obtains higher F1, Kappa and Overall Accuracy scores on the holdout set, as seen in Table I. From Figure 3c we can see that *hsnv1 Concat* obtains a higher maximum and minimum score and obtains more consistent results than *hsnv2 Add*.

Model	F-1			Kappa Score			Overall accuracy		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
hsnv1 Concat	**0.8191**	**0.8350**	**0.8265**	**0.7929**	**0.8102**	**0.8009**	**0.9544**	**0.9573**	**0.9558**
hsnv2 Addition	0.8090	0.8289	0.8222	0.7819	0.8037	0.7963	0.9525	0.9564	0.9550

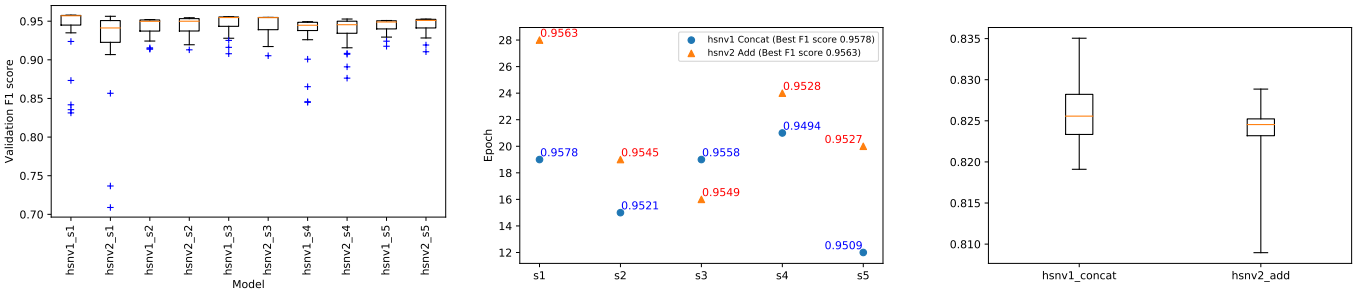
TABLE I
HSN MODEL F1-SCORE RESULTS

For the rest of the presented experiments, we selected the HSN model with concatenation skip connections, as it exhibits higher scores on the holdout set than the original proposed topology.

B. Activation and Normalization

The authors in [22] investigate the impact of Batch Normalization layers inserted before or after the activation layer on two different architectures. Since their results are inconclusive, we compare the impact of Batch Normalization positioning relative to the activation layer and adapt our models to the results.

1) U-Net: The original U-Net topology does not include a Batch Normalization (BN) layer after the convolutional layers, still it is widely used in the available U-Net implementations.



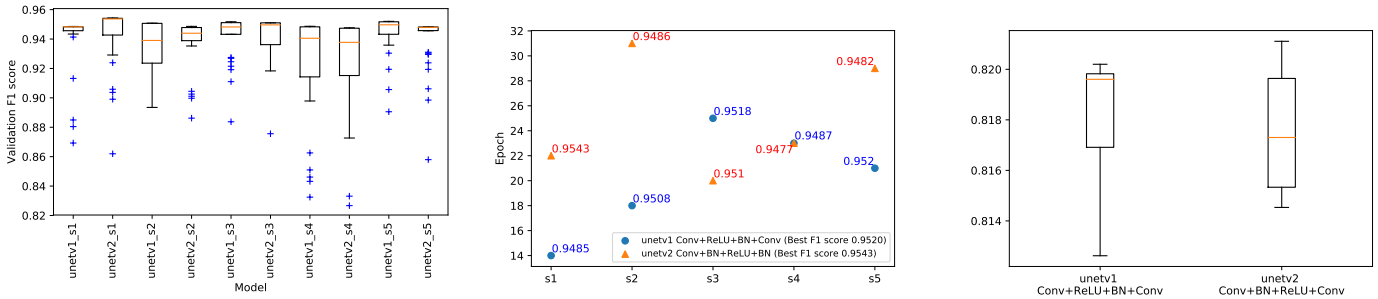
(a) Training

(b) Best epoch during training

(c) F1 score on holdout set

Fig. 3. HSN models with different fusion methods

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 November 2022 doi:10.20944/preprints202211.0226.v1



(a) Training (b) Best epoch during training (c) F1 score on holdout set

Fig. 4. U-Net models with different positioning of the BN layer

In Figure 4 we present results of our experiments with two different positions for the BN layer, before or after the ReLU activation function (non-linearity). In the first configuration, we apply the BN layer after non-liniarity and use BN before in the second setting.

In Figure 4a, the U-Net which positions the BN layer before the activation, *unetv2*, has a better score during training and a more consistent score on all most all training validation splits used.

When applying our models on our holdout set (see Figure 4c), we can see that BN before activation obtained a higher maximum and minimum score while the other configuration obtained a more consistent score. Table II shows the F1, Kappa and Overall Accuracy scores on the holdout set.

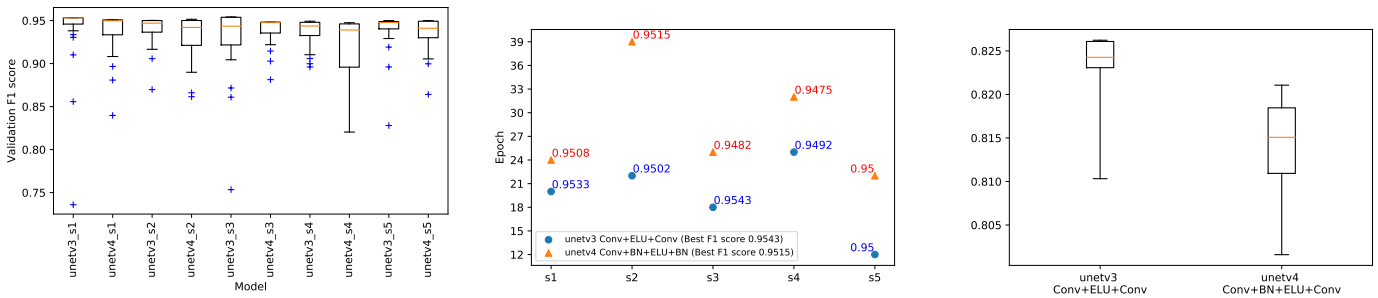
In Figure 4b we can see that on most of the samples used for training, the models which place BN after the activation layer reach their best score on a much earlier training epoch, essentially being faster when it comes to obtaining their maximum training score. It can be argued that this can mean that they also overfit much earlier, however the difference between the best performing models appears to be small. This difference in convergence rates requires further research.

Model	F-1			Kappa Score			Overall accuracy		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
unetv1 Cnv+ReLU+BN+Cnv	0.8126	0.8202	**0.8178**	0.7853	0.7942	**0.7914**	0.9526	0.9548	**0.9541**
unetv2 Cnv+BN+ReLU+Cnv	**0.8145**	**0.8211**	0.8176	**0.7877**	**0.7952**	0.7912	**0.9533**	**0.9549**	**0.9541**
unetv3 Cnv+ELU+Cnv	**0.8103**	**0.8262**	**0.8220**	**0.7823**	**0.8004**	**0.7955**	0.9513	**0.9554**	**0.9542**
unetv4 Cnv+ELU+BN+Cnv	0.8016	0.8211	0.8134	0.7741	0.7946	0.7866	**0.9515**	**0.9544**	0.9533

TABLE II
U-NET MODEL F1-SCORE RESULTS

In the original U-Net, the authors propose using ReLU as a non-liniarity function for the network. We have investigated the placement of BN layer for U-Net topology and choose BN before non-liniarity for our next experiments (as seen in *unetv2*) as it provides an increase in the predictive performance of the model and is more stable during training. In Figure 5 we plot the results obtained with U-Net with ELU activation functions. Authors of [22] suggest not using a BN layer together with an ELU activation and we investigate this approach on U-Net topology. As seen in Figures 5a and 5c, *unetv3* is more consistent during the training process on almost all training validation splits and during the testing phase. In addition to this, in Figure 5b, *unetv3* reaches a better performance in an earlier epoch than the U-net version with BN and ELU (*unetv4*). Compared to the results obtained when using ReLU activation function, we notice an improvement in the F1, Kappa and Overall Accuracy scores on *unetv3*, as depicted in Table II.

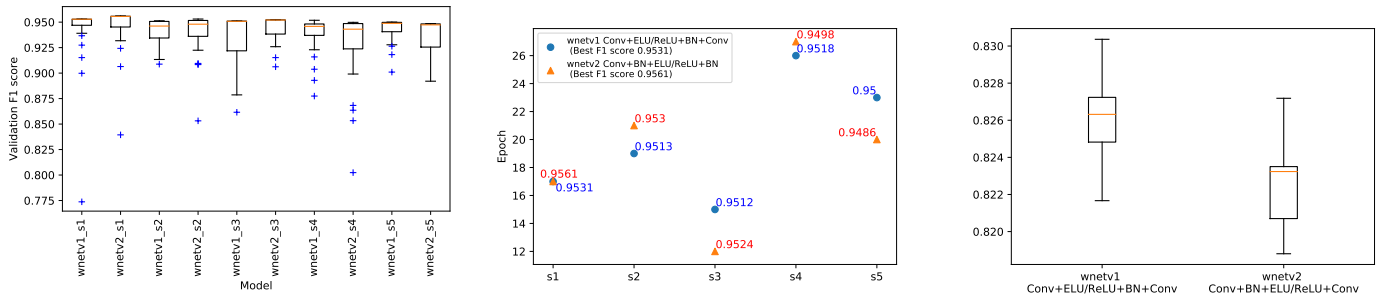
2) W-Net: W-Net is composed of two bridged U-Nets, having a combination of ReLU and ELU activation functions, proposing a BN layer before the non-liniarity (*wnetv2*). It follows the same downsampling and upsampling factors as in U-Net. As seen in the validation F1 score statistics from Figure 6a and the results on the holdout set, depicted in Figure 6c we can observe that using the BN layer after the non-liniarity (*wnetv1*) leads to an improvement on both the validation and testing F1 scores, resulting also in a much more stable training process. From Table III, we can also see higher F1, Kappa and



(a) Training (b) Best epoch during training (c) F1 score on holdout set

Fig. 5. U-Net models with ELU activation function

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 November 2022 doi:10.20944/preprints202211.0226.v1



(a) Training (b) Best epoch during training (c) F1 score on holdout set

Fig. 6. W-Net models with different positioning of the BN layer

Overall Accuracy score obtained by *wnetv1* on the holdout test.

This is in contrast with the results obtained with U-Net, where the experiments suggest using BN before activation layer. However, it might be due to the combination of the ELU activation function and BN, problem identified by [22].

Interestingly we also observe in Figure 6b that the relative positioning of the BN layer does not significantly impact training convergence (training epochs needed for reaching the maximum validation F1 score), in contrast to the other analyzed architectures.

Model	F-1			Kappa Score			Overall accuracy		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
wnetv1 Cnv+ELU/ReLU+BN+Cnv	**0.8217**	**0.8304**	**0.8261**	**0.7959**	**0.8055**	**0.8007**	**0.9553**	**0.9570**	**0.9560**
wnetv2 Cnv+BN+ELU/ReLU+Cnv	0.8188	0.8272	0.8227	0.7928	0.8020	0.7969	0.9546	0.9564	0.9552
wnetv3 Cnv+ELU+Cnv	0.8252	**0.8361**	**0.8324**	0.7990	**0.8115**	**0.8073**	0.9547	**0.9576**	**0.9567**
wnetv4 Cnv+ReLU+BN+ELU	**0.8264**	0.8349	0.8295	**0.8007**	0.8099	0.8041	**0.9554**	0.9570	0.9561
wnetv5 Cnv+BN+ReLU+ELU	0.8235	0.8359	0.8311	0.7975	0.8112	0.8059	0.9549	0.9575	0.9565

TABLE III
W-NET MODEL F1-SCORE RESULTS

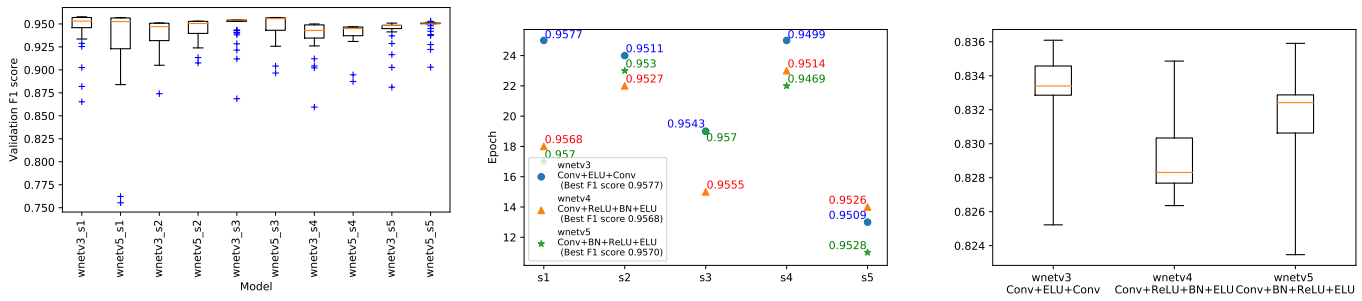
Figure 7 presents the results obtained by removing the BN near the ELU activation layers in W-Net. Compared to the original proposed topology, we do notice a perfomance improvement when using ELU only, with no BN layer, as in *wnetv3* (Table III). In *wnetv4* and *wnetv5* we experiment with topologies using a combination of ELU and ReLU, as originally proposed. However, we do not use a BN layer near ELU, but keep the BN when using ReLU: after ReLU (*wnetv4*) and before ReLU (*wnetv5*). As depicted in Figure 7b *wnetv5* converges faster in reaching the maximum validation F1 score. During testing, we notice an improvement in placing BN before ReLU (Figure 7c), where *wnetv5* is more stable than *wnetv4*, having also a higher maximum and average F1, Kappa and Overall Accuracy score (Table III). In accord with U-net and the original topology, we obtain the best results by placing BN before ReLU. We do notice an improvement on holdout set, in both *wnetv4* and *wnetv5*, as opposed to *wnetv1* and *wnetv2*, by eliminating the BN layer near ELU.

3) HSN: We analyze the impact of the position of the BN layer in the best performing HSN model resulted from the analysis in section IV-A. Namely we used the HSN with concatenation for feature fusion, which may be seen as a U-Net model with inception and residual blocks. The computed F1-score on the holdout set is shown in Table IV.

Although positioning the BN before non-liniarity (*hsnv3*), as in the original HSN proposed topology, brings a higher minimum F1 value, the model with BN after the activation (*hsnv1*) obtains an improvement if we look at the maximum and mean values. This is not in accord with the results obtained by our U-Net experiments.

Figure 8a, shows *hsnv1* to have a more consistent validation score during training as well as a better overall score during testing (8c).

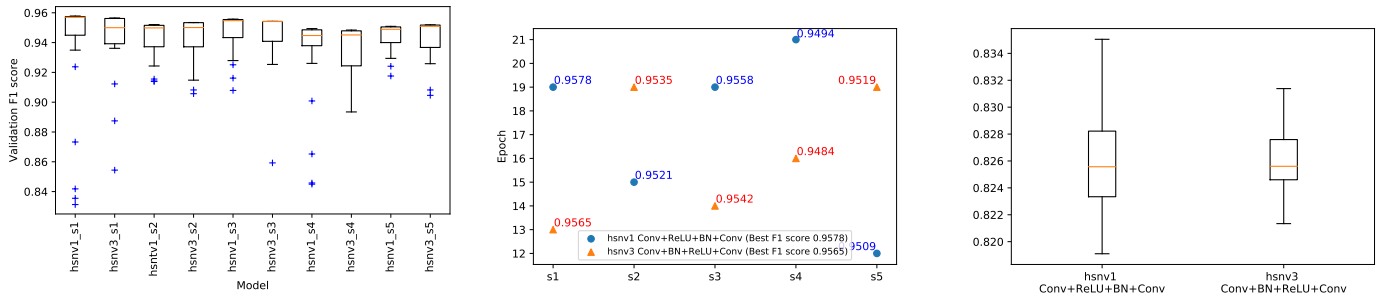
An interesting difference arises when we compare the results of HSN best epochs from figure 8b to the ones obtained by U-Net models from figure 4b. In contrast to U-Net our HSN model that uses BN layer after the activation layer requires more epochs to reach its best score when compared to the model using activation layer before BN layer. The difference in behavior when it comes to the speed of convergence between



(a) Training (b) Best epoch during training (c) F1 score on holdout set

Fig. 7. W-Net models with ELU activation function

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 November 2022 doi:10.20944/preprints202211.0226.v1



(a) Training (b) Best epoch during training (c) F1 score on holdout set

Fig. 8. HSN models with different positioning of the BN layer

these two topologies requires further research.

Model	F-1			Kappa Score			Overall accuracy		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
hsnv1 Cnv+ReLU+BN+Cnv	0.8191	**0.8350**	**0.8265**	0.7929	**0.8102**	**0.8009**	0.9544	**0.9573**	**0.9558**
hsnv3 Cnv+BN+ReLU+Cnv	**0.8213**	0.8314	0.8261	**0.7954**	0.8062	0.8006	**0.9549**	0.9565	**0.9558**
hsnv4 Cnv+ELU+Cnv	**0.8275**	**0.8350**	**0.8323**	**0.8015**	**0.8102**	**0.8071**	**0.9551**	**0.9573**	**0.9566**
hsnv4 Cnv+BN+ELU+Cnv	0.8191	0.8266	0.8239	0.7929	0.8012	0.7982	0.9544	0.9559	0.9553

TABLE IV
HSN MODEL F1-SCORE RESULTS

Just as the U-Net model, using the ELU activation function without BN (*hsnv4*), brings stability during training (Figure 9a), the networks reaching their best score in an earlier epoch (Figure 9b), than combining BN and ELU, by placing the BN before non-liniarity as proposed in the original HSN (*hsnv5*). Also, as depicted in Figure 9c, *hsnv4* has better results in the testing phase. ELU only version of HSN also distinguish itself by yielding the best predictive performance as can be seen in the results presented in Table IV.

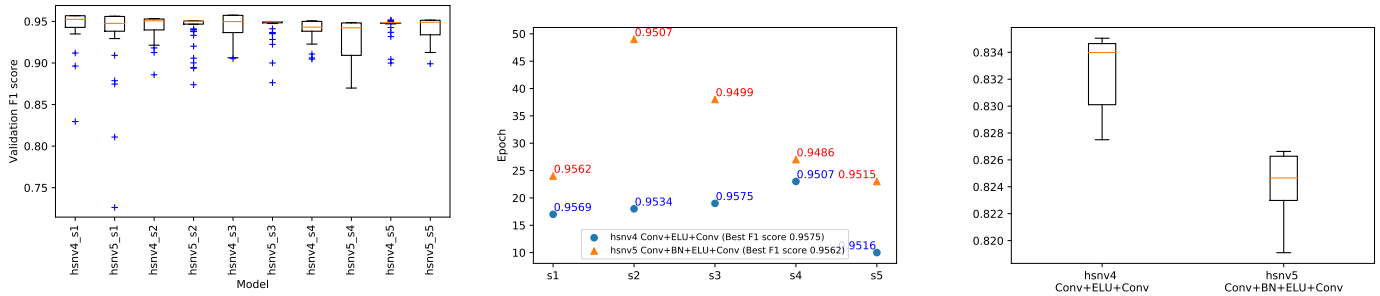
Model	F-1			Kappa Score			Overall accuracy		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
segnetv1 Cnv+ReLU+BN+Cnv	**0.8198**	**0.8273**	**0.8249**	**0.7941**	**0.8021**	**0.7996**	**0.9551**	**0.9565**	**0.9559**
segnetv2 Cnv+BN+ReLU+Cnv	0.7957	0.8182	0.8069	0.7690	0.7924	0.7806	0.9518	0.9550	0.9534
segnetv3 Cnv+ELU+Cnv	0.0000	**0.8265**	0.3243	0.0000	**0.8007**	0.3136	0.8645	0.9555	0.8999
segnetv4 Cnv+ELU+BN+Cnv	**0.8131**	0.8249	**0.8189**	**0.7869**	0.7995	**0.7932**	**0.9541**	**0.9559**	**0.9550**

TABLE V
SEGNET MODEL F1-SCORE RESULTS

4) Segnet: For the following analysis we have used the bayesian version of the Segnet topology, originally build and tested for *224x224* pixel images. The authors propose a Batch Normalization layer after every convolutional layer and before the activation function (*segnetv2* in our experiments). However, as seen in Figures 10a and 10b, considering the validation F1 score, positioning after non-linearity (*segnetv1* in our experiments) stabilizes the network, with the model converging faster to maximum score in all samples. As can be seen in Table V and Figure 10c *segnetv1* also shows an improvement in the holdout set scores.

Our next step is examining the effects of ELU activation positioning with respect to the BN layer. The results of these experiments where surprising. We can see from Figure 11a that during training, *segnetv3* (using only ELU activation) has a markedly poor performance. Scores obtained during training for *segnetv4* (BN after ELU activation) are much more consistent while at the same time are higher than *segnetv3*. This difference is maintained in the convergence rate of the best score obtained during training as seen in Figure 10b. We can see that *segnetv4* is consistently reaching its best score at approximate the same epoch (24 to 30) during training, while *segnetv3* is fluctuating from a maximum of 73 to a minimum of 1.

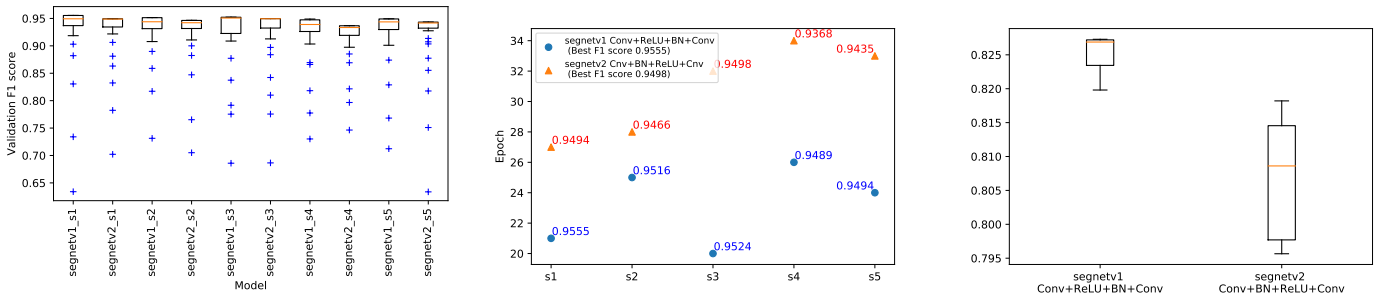
The most surprising result can be found in Figure 11c which shows out of sample predictive performance. The version of Segnet using only the ELU activation has poor performance to the point in which some of the models have obtained an F1 score of 0. We think that this is due to several factors. First, Segnet is one of the larges topologies we have tested having in excess of 30 million trainable parameters, this coupled with the ELU activation has likely resulted in an exploding gradient which is noticeably improved by the addition of BN layer



(a) Training (b) Best epoch during training (c) Testing

Fig. 9. HSN models with ELU activation function

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 November 2022 doi:10.20944/preprints202211.0226.v1



(a) Training (b) Best epoch during training (c) F1 score on holdout set

Fig. 10. Segnet models with different positioning of the BN layer

after the ELU activation in *segnetv4*. We have also considered the possibility that in the case of *segnetv3* our method of using randomly sampled training and validation sets might be unrepresentative for the holdout set. In order to check this hypothesis we generated a new training/validation splits, obtaining similar results, some of the models yielding again an F1 score of 0. This has lead us to strongly favor the exploding gradient theory as being the most likely one. Further experimentation into this issue is needed to pinpoint the exact causality of this behavior.

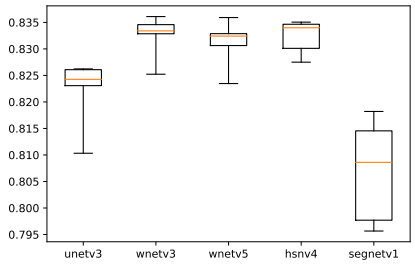


Fig. 12. Best model comparison F1 score on Urban3D holdout set

Overall comparison

Finally we select the best performing models on the Urban3D dataset and analyze how they perform on the ISPRS Postdam dataset, using the same training and testing methodology. The dataset contains multiple classes out of which we select only the building class in order to ensure compatibility with the Urband3D dataset. In Figure 12 and Figure 13 we can see a performance overview for all of the models presented in this paper, on both datasets. From all of the models tested HSN and W-Net have the highest scores on the holdout sets. In Table VI and Table VII we can view a detailed breakdown of the scores obtained.

We can see that the *W-Net* based topologies have a consistently superior score on both the Urban3D and ISPRS Potsdam datasets, despite the differences in spatial resolution ($0.5m$ for Urband3D and $5cm$ for ISPRS Potsdam).

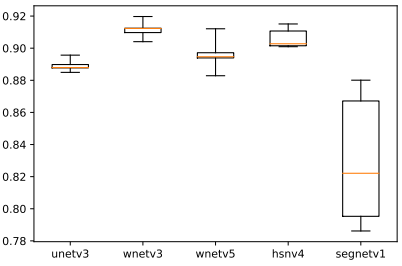
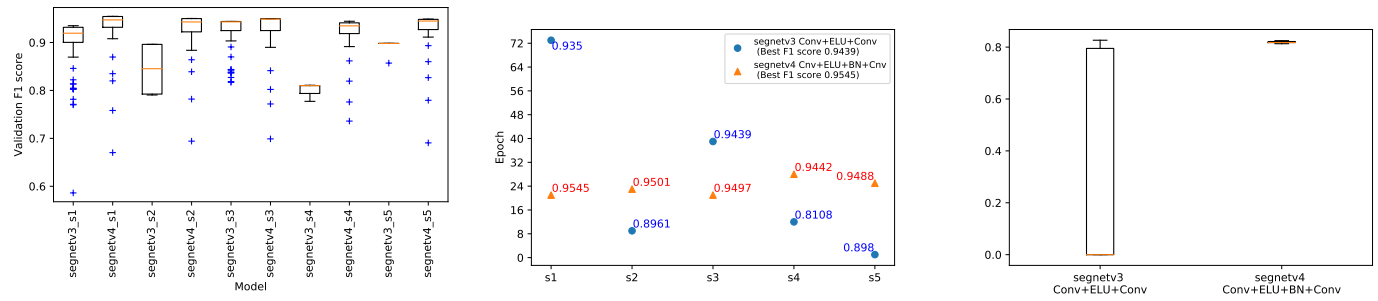


Fig. 13. Best model comparison F1 score on Postdam holdout set



(a) Training (b) Best epoch during training (c) F1 score on holdout set

Fig. 11. Segnet models with ELU activation function

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 November 2022 doi:10.20944/preprints202211.0226.v1

Model	F-1			Kappa Score			Overall accuracy		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
unetv3 Cnv+ELU+Cnv	0.8103	0.8262	0.8220	0.7823	0.8004	0.7955	0.9513	0.9554	0.9542
wnetv3 Cnv+ELU+Cnv	0.8252	**0.8361**	**0.8324**	0.7990	**0.8115**	**0.8073**	0.9547	**0.9576**	**0.9567**
wnetv5 Cnv+BN+ReLU+ELU	0.8235	0.8359	0.8311	0.7975	0.8112	0.8059	0.9549	0.9575	0.9565
hsnv4 Cnv+ELU+Cnv	**0.8275**	0.8350	0.8323	**0.8015**	0.8102	0.8071	**0.9551**	0.9573	0.9566
segnetv1 Cnv+Relu+BN+Cnv	0.7957	0.8182	0.8069	0.7690	0.7924	0.7806	0.9518	0.9550	0.9534

TABLE VI
BEST MODEL F1-SCORE RESULT ON URBAN3D HOLDOUT SET

Model	F-1			Kappa Score			Overall accuracy		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
unetv3 Cnv+ELU+Cnv	0.8849	0.8957	0.8892	0.8503	0.8634	0.8555	0.9475	0.9517	0.9491
wnetv3 Cnv+ELU+Cnv	**0.9041**	**0.9197**	**0.9116**	**0.8746**	**0.8951**	**0.8844**	**0.9557**	**0.9630**	**0.9590**
wnetv5 Cnv+BN+ReLU+ELU	0.8828	0.9121	0.8961	0.8486	0.8851	0.8651	0.9475	0.9595	0.9529
hsnv4 Cnv+ELU+Cnv	0.9009	0.9151	0.9062	0.8708	0.8888	0.8774	0.9545	0.9606	0.9566
segnetv1 Cnv+Relu+BN+Cnv	0.7862	0.8801	0.8302	0.7345	0.8440	0.7861	0.9139	0.9452	0.9288

TABLE VII
BEST MODEL F1-SCORE RESULT ON ISPRS POSTDAM BINARY BUILDING

V. CONCLUSION AND FUTURE WORK

In this paper, we showed how different deep neural network topologies perform for semantic image segmentation. We have focused on single model performance , proposing changes to the network topologies, regarding the placement of the BN layer and alternating between ReLU and ELU activation functions.

In our first experiments we showed that for HSN we obtained better performance using concatenation instead of addition for feature fusion. This difference in predictive performance was observable for both training and holdout set. The most likely explanation for this is that concatenation preserves the original form of the previous convolutional layer.

In our second set of experiments we investigated the impact of different activation functions and BN layer positioning. For U-Net we have found that placing BN before ReLU activation yields better predictive performance however, convergence during training is faster if BN is placed after ReLU. The best performance was attained by using ELU activation without any BN layers (*unetv3*). This also reduces substantially the memory requirements of U-Net. On both Urban and Postdam holdout sets, *unetv3* ranks 4th amongst our best performing models.

W-Net has been found to perform better by positioning BN after the ReLU/ELU layer for both training and holdout sets, instead of the initial version with BN before non-liniarity. Furthermore, we have found that when using ELU activation by removing BN layer from its vicinity we obtained some performance improvements. The best results were obtained only by using ELU without any BN layer (*wnetv3*), just like in the case of U-Net. However, alternating ReLU and ELU activation functions (*wnetv5*) also achieves close results to ELU only W-Net and ELU HSN, but only by using BN before ReLU and no BN near ELU activation. From Figures 12 and 13, we can see that *wnetv3* has the most consistent results, on both datasets.

The results obtained for HSN show that it has better performance when placing BN after ReLU activation. It is important to note that performance of HSN using ELU activation is similar to what we observed for U-Net and W-Net, having best performance when using no BN near ELU activation (*hsnv4*). ELU only HSN ranks on the 2nd position, in our comparison, on both Urban and Postdam datasets, after *wnetv3*.

Segnet experiments resulted in some performance impact when it comes to positioning of the BN layer relative to ReLU activation. The Segnet models which had ReLU after BN layer (*segnetv1*) resulted in more consistent training scores and higher F1 score on the holdout set, as opposed to the original Segnet with BN before non-liniarity. The impact of using ELU activation in the case of Segnet was quite significant. Contrary to the topologies tested until this point ELU activation resulted in significant performance degradation to the point that some models had an F1 score of 0.0. Adding BN after ELU activation resulted in a much more consistent training score and overall predictive performance. Out of our 5 best models, *segnetv1* has the lowest performance on the Urban3D and ISPRS Postdam holdout sets.

We provide a Keras based implementation[14] of our best performing model topologies. We have also publish pretrained weights for our models, to serve as initialization for further training of deep learning networks applied on Remote Sensing data.

Our experiments have shown that further research is required regarding the positioning of activation functions relative to the BN layer and its impact not only on overall out of sample predictive performance but also on training convergence rate. As our experiments where conducted on heterogeneous hardware using bare metal, AWS and Google cloud resources we are unable to clearly say what the impact on per epoch training time is with certainty. We were only able to gauge the number of training epochs it took for a model to reach it's maximum convergence during training.

An interesting observation is that topologies based around skip connections and alternating activation functions obtained some of the best results. Future research will focus on implementing an improved topology based on this observation.

ACKNOWLEDGMENT

This work was primarily supported by a grant of the Romanian Ministry of Education and Research, CNCS-UEFISCDI, project number PN-III-P2-2.1-PED-2019-4878, within PNCDI III. This work was partially supported by a grant of the Romanian Ministry of Education and Research, CNCS-UEFISCDI, project number PN-III-P4-ID-PCE-2020-0407, within PNCDI III.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] 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*, 2016, pp. 770–778.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[5] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: a comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[14]Private Zenodo Dataset: https://zenodo.org/record/2611283#.XJ30EyB10s

 doi:10.20944/preprints202211.0226.v1

[6] G. Christie, N. Fendley, J. Wilson, and R. Mukherjee, "Functional map of the world," in *CVPR*, 2018.

[7] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar, "Deepglobe 2018: A challenge to parse the earth through satellite images," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[8] F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf, "The isprs benchmark on urban object classification and 3d building reconstruction," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 1, no. 3, pp. 293–298, 2012.

[9] H. Goldberg, M. Brown, and S. Wang, "A benchmark for building footprint classification using orthorectified rgb imagery and digital surface models from commercial satellites," in *Proceedings of IEEE Applied Imagery Pattern Recognition Workshop 2017*, 2017.

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[12] V. Iglovikov, S. Mushinskiy, and V. Osin, "Satellite imagery feature detection using deep convolutional neural network: A kaggle competition," *arXiv preprint arXiv:1706.06169*, 2017.

[13] W. Chen, Y. Zhang, J. He, Y. Qiao, Y. Chen, H. Shi, and X. Tang, "W-net: Bridged u-net for 2d medical image segmentation," *arXiv preprint arXiv:1807.04459*, 2018.

[14] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[15] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, "Classification with an edge: Improving semantic image segmentation with boundary detection," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 135, pp. 158–172, 2018.

[18] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[19] D. Marmanis, J. D. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla, "Semantic segmentation of aerial images with an ensemble of cnns," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3, p. 473, 2016.

[20] Y. Liu, D. Minh Nguyen, N. Deligiannis, W. Ding, and A. Munteanu, "Hourglass-shapenetwork based semantic segmentation for high resolution aerial imagery," *Remote Sensing*, vol. 9, no. 6, p. 522, 2017.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[22] D. Mishkin, N. Sergievskiy, and J. Matas, "Systematic evaluation of convolution neural network advances on the imagenet," *Computer Vision and Image Understanding*, vol. 161, pp. 11–19, 2017.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[24] Z. Miao, K. Fu, H. Sun, X. Sun, and M. Yan, "Automatic water-body segmentation from high-resolution satellite images via deep networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 4, pp. 602–606, 2018.

Teodora Selea received her B.S. from the West University of Timisoara, Romania, in 2015, and is a 2^{nd} year Ph.D. student at the West University of Timisoara, Romania. Starting from 2015 she participated in multiple research projects in the area of distributed systems, machine learning and remote sensing. Her research interest include cloud computing, machine learning, computer vision and remote sensing analysis.

Gabriel Iuhasz received his Ph.D. degree in Machine Learning and Distributed systems from the West University of Timisoara, Romania, in 2014. Starting from 2017 he is an Assistant Professor in the Department of Computer Science, West University of Timisoara. From 2013 he is also a researcher at Institute e-Austria, Romania. He was involved in multiple research projects dealing with distributed systems, machine learning and cloud computing. Recently he is focusing on research in the fields of remote sensing, image processing and deep learning.

Marian Neagul received the Ph.D. degree in distributed systems from West University of Timisoara, Romania, in 2015. From 2016 he is Assistant Professor with the Department of Computer Science, West University of Timisoara. From 2015 he is a Research Scientist at Institute e-Austria, Romania. He participated in multiple research projects in the area of distributed systems, machine learning and remote sensing. His research interests include cloud computing, machine learning, remote sensing image processing, self-organizing systems and high performance computing