INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 6)
Available online at: www.ijariit.com

Application of deep Neural Networks for object detection in satellite images

Akhilesh Kakade
akhileshkakade1995@gmail.com
Vellore Institute of Technology, Vellore, Tamil Nadu

Vikas C. Kakade
vikas.c.kakade@gmail.com
Tuljaram Chaturchand College, Baramati, Maharashtra

ABSTRACT

The rapid growth in satellite imagery has helped scientists understand the Earth better. The improved understanding of the Earth makes it possible for scientists to perform better in all activities that range from disaster management in the form of mobilizing resources to comprehending global warming by monitoring its effects. The major limitation of this achievement is the assumption that significant features in satellite images, like buildings, roads, trees, or water bodies, can be easily identified, either manually or semi-automatically, but always perfectly. In this paper to overcome this limitation, we use different convolutional neural networks with modifications such as proposed PSPNet, U-net architecture, Inverted pyramid and XGBoost algorithm for accurately detecting specified features in satellite images from Defense Science and Technology Laboratory (DSTL) database. Automation of feature detection in satellite images is not only useful in making smart and quick decisions, but also in bringing innovation in application of computer vision methodologies to satellite imagery.

Keywords— *Satellite imagery, Image classification, Convolutional Neural Networks*

1. INTRODUCTION

The phenomenal growth in the variety, the improved accessibility, and the global availability of satellite imagery has resulted in a dramatic improvement in the understanding of the planet Earth. Such an understanding is required in situations ranging from emergency operations of mobilizing resources during disasters to routine activities like monitoring effects of global warming. However, there is still a great limitation to these developments. The limitation is the assumption that detecting features or objects of interest in satellite images can be easily done either manually or with partial help of computers, that is, semi-automatically. On one hand, this assumption puts a tremendous burden on experts that are responsible for detecting and identifying such objects of interest. On the other hand, there have been spectacular improvements in processing capacities of the processing units and great advancements in computer vision with help of machine learning technologies like deep learning through deep neural network. It is then natural to think about utilizing the hardware and logarithmic advancements in automating important or significant objects in satellite images. This identification, if automatic, accurate, and quick, can be very helpful in several applications like creating and updating maps for land use and landholding information, monitoring environmental indicators, improving urban planning, and responding disaster situations.

This paper is inspired by the Kaggle competition "Dstl Satellite Imagery Feature Detection", announced and conducted more than two years ago. This paper aims at developing a Deep Neural Network using the PSPNet architecture with modifications for detecting specified objects in satellite images provided to the Kaggle competitors. The data set is not too large, and it is therefore considered manageable for supervised machine learning algorithms that are appropriate for problems of this nature. This paper consists of the following major steps.

- Adaptation of Convolutional Neural Networks (CNN) to multispectral image data and evaluation of data fusion strategies for semantic segmentation of satellite images.
- Introduction of a joint training objective for defining the desired output for the purpose of image segmentation.

2. PROBLEM STATEMENT

Satellite images contain a huge amount of data, both visible and invisible. A variety of methods have been developed for extracting information from satellite images due to different applications like environmental monitoring, urban and rural development planning, management of natural resources and many more. One of the recent trends is to extract information from image data for the purpose of security in the form of detecting and tracking vehicles, identifying illegal constructions, water bodies, roads and other tracks, and so on. The Kaggle competition required the participants to identify objects of the following types:

a) Buildings
b) Miscellaneous manmade structures
c) Road
d) Track (Cart/dirt track, footpath/trail)
e) Tree (standalone tree, group of trees, etc.)
f) Crop (grain crops like wheat, row crops like potatoes and turnips, contour ploughing, and cropland)
g) Waterway

h) Standing water
i) Large vehicle (truck, bus, trailer, etc.)
j) Small vehicle (car, van, motorcycle, etc.)

After reviewing the literature, it was decided to generate feature masks for different objects. The reason for generating masks was to only achieve a semantic segmentation by only detecting categories of different objects without identifying the objects individually.

3. DATA OVERVIEW

DSTL has provided 1km x 1km satellite images in formats, namely panchromatic, three-band (RBG) and two eight four-band formats. The total number of images provided by DSTL is 450. The number of training images is 25, the test set consists of 32 images, and the rest of the images from the validation set. Every image is available in the three versions. The following table gives more information on the three versions.

Table 1: Three Versions ofaSatellite Image

Type	Wavebands	Pixel Resolution	No. of Channels	Size
Grayscale	Panchromatic	0.31m	1	3348 x 3392
3-band	RGB	0.31m	3	3348 x 3392
8-band	Multispectral	1.24m	8	837 x 848
8-band	Short-Wave Infrared	7.5m	8	134 x 136

The two 8-band channels have to be resized and aligned to match the 3-band channels. All channels are then concatenated to form a single 20-channel input image for processing.

The reason for utilizing all the 20 inputs is that every channel covers a unique range of the spectrum and hence records unique features that other channels are not capable of observing such as World View 3 satellite[19].

The spectral resolution of these images is also higher due to these having 11-bit and 14-bit depth for every pixel instead of the traditional 8-bit depth of earlier satellites. It is also important to note that images in different channels are captured at different time points.

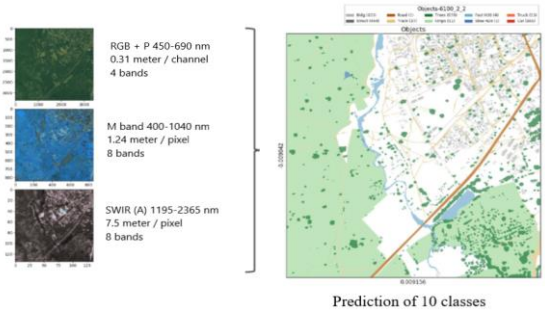


Fig. 1: The spectrum converges of the 4 versions of images and their mutual connection.

4. LITERATURE SURVEY

In semantic segmentation we take an image and divide it into meaningful parts. Each part is examined at the pixel level and classified into a predefined class. Most of the times deep learning techniques are used for such classification. One such deep learning technique use in semantic segmentation is Convolutional Neural Networks (CNN). CNN is a supervised classification method that can learn the important features of an image in an end to end manner. It can also learn optimum features very quickly and does not underperform even if the underlying image has minor variations.

Muhammad Jaleed Khan et.al [2] proposed, a target detection system for satellite imagery which uses EdgeBoxes and Convolutional Neural Network (CNN) for classifying target and non-target objects in a scene. The edge information of targets in satellite imagery contains very prominent and concise attributes. EdgeBoxes uses the edge information to filter the set of target proposals. The prediction was limited to two class objects whether it is an artifact or non-artifact. The proposed model can't be used for multi-class object detection.

The state of the art in image segmentation in DeepLab V3, which implements a ResNet model using dilated/atrous convolutions [3]. We have chosen architecture like modified PSPNet, U-net architecture over DeepLab V3 as it has familiar implementation using concepts learned in object class, also it uses comparably less parameters and trains faster than DCNN models like DeepLab and other pixel level classifiers. In real time applications, detecting an object is critical. A variant of CNN - the faster R-CNN [5] can be used in real time applications to detect objects quickly. It shares the computation of convolutional layers between proposals because of Region of Interest (RoI) Pooling. The system is trained end to end. The improvement in speed is not large in faster R-CNN because the region proposals are generated separately by another model.

Vladimir Iglovikov[16] proposed, an approach of using modified fully convolutional neural network for multispectral data processing and he is also Kaagle DSTL competition's 3rd place winner. The proposed system consists of various steps such as using multispectral U-net architecture with modifications to DSTL satellite images with joint training objective, analysis of boundary effects and use of reflectance indices.

5. METHODOLOGY

5.1 Preprocessing Steps

Preprocessing of images involve the following four steps.

Step 1: The four versions of an image, namely the panchromatic (1 band), RGB (3 bands), multispectral (8 bands), and Short-Wave Infrared (SWIR) (8 bands) are input and synchronized before concatenating them for further analysis.

Step 2: Images in training data are subjected to the scale percentile process to make them comparable to other images.

Step 3: The concatenated 20-channel image is converted to a multipolygon WKT format for creating masks for different objects for easy detection.

Step 4: Select patches of a specified size from images for training the Convolutional Neural Network (CNN).

The last step is necessary because the original images are too large for training the CNN. This study has used 224 as the input size for CNN. This size has been arrived at after carrying out experiments.

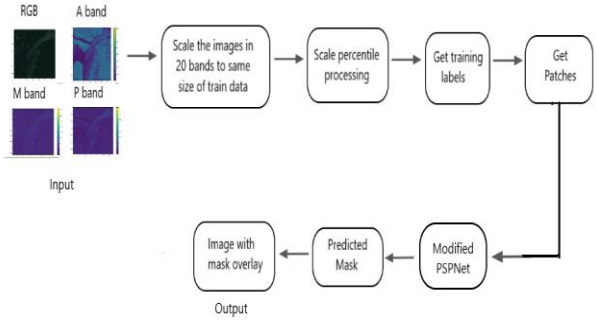


Fig. 2: Proposed Pipeline

5.2 Scale Percentile Processing

The 3-band (RGB) images have mostly 8-bit spectral resolution. That is, each pixel has 256 levels, from 0 to 255. For example, a completely red pixel has the spectral value (255, 0, 0), a completely green pixel (0, 255, 0), and a completely blue (0, 0, 255), while a white pixel has the spectral value (255, 255, 255) and the perfectly black pixel has the spectral value (0, 0, 0). More recent satellite sensors have 11-bit or 14-bit images and hence have a larger range of spectral pixel values. Of course, these images can store more information, but they also require larger storage spaces. This feature may also have compatibility issues with software.

The images used in this study are converted from 14-bit to 8-bit spectral resolution. Even through software tools like NumPy are available for this conversion, this study has made use of the gdal library. The scale percentile process normalizes the image luminance and resizes the input image to a square image that has side length 112. The resizing does not affect the aspect ratio and preserves it. This process specifies a percentile range of 1 to 99 and pixels having spectral values outside this range (that is, below the 1st and above the 99thpercentiles) are removed since they are declared to be outliers. The cleaned image is then rescaled to 8-bit spectral resolution.

Let X_{in} be an 8-bit input band whose value range from 0 to 255, say a=0 and b=1 and $X_{out}$is output band. For scale percentile processing we choose lower percentile and higher percentile as c and d to be 1st and 99th percentile respectively.

The scale percentile processing can be represented using the following function:

$$X_{out} = (X_{in} - c)\frac{(b-a)}{(d-c)} + a$$

We clip the values to minimum and maximum such as

$$X_{out}\,[X_{out} < a] = a \text{ and } X_{in}\,[X_{in} > b] = b$$

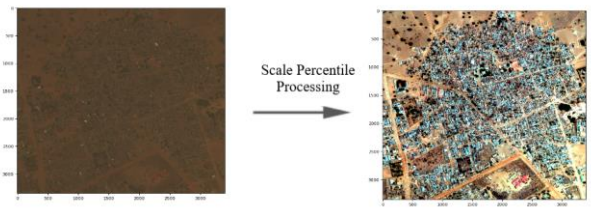


Fig. 3: Preprocessing step

5.3 Pateches and Input

The literature on satellite imagery mentions the panchromatic band as p-band, the 8-channel multi-spectral band as M-band, and the 8-channel short wave infrared band as A-band. The four bands do not have the same resolution and have therefore to be resized for spatial synchronization. It is also found that the frequency distributions of the 10 objects to be detected are skewed in the images in the training set together. The number of images in the training set is 25 and is not enough for training. Further, the size of every image is too large for processing. Considering all issues related to images, every image is divided into square parts of size 112 □ 112 and these parts are called patches. Object detection is then carried out on these patches rather than original images. This has allowed the Deep Neural Network to train properly due to large training data, while enhancing the processing speed due to reduced size of every individual input data element.

6. DEEP NEURAL NETWROK ARCHIECTURES

Object detection in the given satellite images is the objective of this study and training data has been provided to develop the classification rules. However, there is no evidence in the literature that one particular Deep Neural Network architecture is optimal. This study has deployed four different architectures such as multispectral U-net architecture, Inverted pyramid, modified PSPNet and XGBoost algorithm, so that the best can be identified at the end of the study. These four architectures are briefly described here

6.1 Multispectral U-NET architecture model

The literature indicates that most of image analysis and classification problems are solved with Deep Convolutional Neural Network (CNN). The U-net architecture [7] gives a fully connected CNN. It was developed by a German research team at University of Freiburg for biomedical image segmentation. It has shown a good performance at image segmentation for the problem of nerve detection in ultrasound images [18].

The U-net architecture has contractive and expansive paths. The architecture of contractive path is usually convolution neural network. Batch normalization has been used in this study for accelerating convergence during training. Also, the primary activation function is exponential instead of linear (ELU) [9]. The following figure shows the multispectral U-net architecture used in this study.

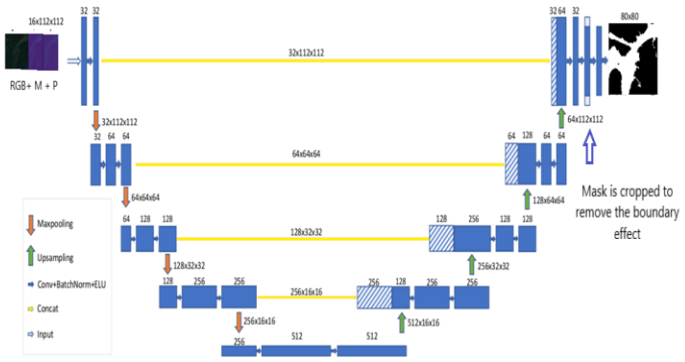


Fig. 4: Multispectral U-net Architecture

Hyperparameters used during training U-net model, batch size is set to 16, primary activation function used is exponential linear unit (ELU) [9], learning rate is set to 0.00001, optimizer used is Adam, and loss used is 'binary_crossentropy'

6.2 Inverted Pyramid Model

Inverted pyramid architecture is an experimental model by Danzelmo[17] and it has been used in this study as the second option. The image decreases in size as they pass through the network. Every path of the network uses different parts of the image while the output size is fixed. Dilated convolution is used in the early part of the network for decreasing the image size while retaining larger receptive field for output neurons. Dilated convolutions are particularly popular in the field of real-time segmentation. Cropping2D layer is used for 2D input layer. It crops along spatial dimensions, i.e. height and width. Different paths are combined later in the network. Dropout is used at final layers for added regularization as no max pooling is used. All Conv2D layers are represented as conv->batch norm->elu but the extra layers are suppressed to make viewing slightly easier. This architecture is still at the experimental stage. It is shown in the figure 5.

Hyperparameters used during training inverted pyramid model, batch size is set to 16, primary activation function used is exponential linear unit (ELU) [9], learning rate is set to 0.00001, optimizer used is Adam, and loss used is 'binary_crossentropy'.

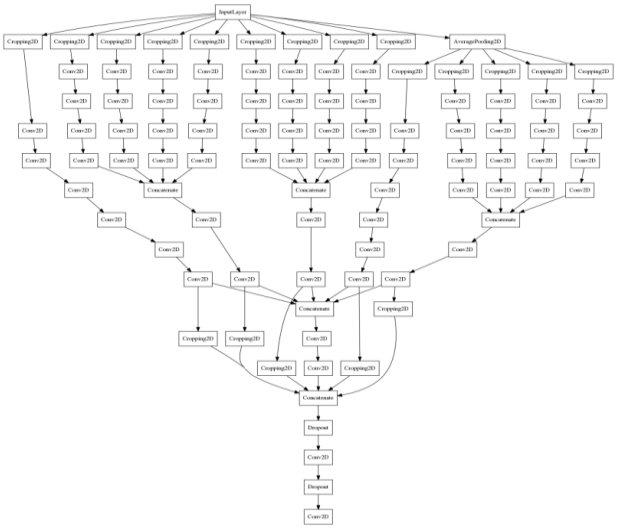


Fig. 5: Inverted Pyramid Model

6.3 Modifiedpyramid Scene Parsingnetwork- PSPNET

Zhao et al. (2017) [11] introduced this state-of-the-art deep learning model for semantic image segmentation. This network model focuses on exploring global information at different scales when compared with other convolution neural network (CNN) models. PSPNet employs a pyramid pooling module on the feature map that is generated by ResNet[13] for creating pooled feature maps at different levels. These features are merged for use in further analysis.

This study has used a modified version of the original PSPNet. As part of the modification, an encoder-decoder is added before the CNN for converting the 20-channel input image in to a 3-channel image for input to the CNN. The pyramid pooling module features four levels with varying bin sizes, and these are appended to the feature map generated by ResNet. A 1×1 convolutional layer is added to each pyramid level to reduce its dimension to a specified depth. The total of channels of the four levels is equated to the dimension of original feature map. The resulting five feature maps are concatenated to get one feature map that can be used for the final stage of analysis, namely object detection.

In figure 6 given an input image (a), we first pass the input layer of 20-channel input image data to (b) i.e. encoder decoder to convert it into output layer of 3 channels. This helps to preserve the maximum features during prediction and avoids the heavy computations to be carried for huge 20-channel input layer. Then we pass the 3-channel layer as an input to CNN i.e. ResNet-101 to get (c) i.e. feature map of the last convolutional Layer. Then a pyramid parsing module (d) is applied to harvest different sub-region representations, followed by up-sampling and concatenation layers to form the final feature representation, which carries both local and global context information in final prediction i.e. (e). Finally, the representation is fed into a convolution layer to get the final per-pixel prediction.

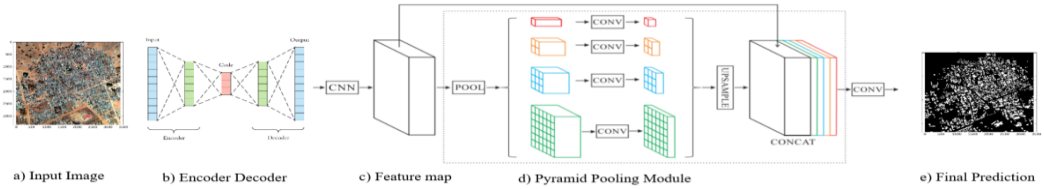


Fig. 6: Overview of out proposed PSPNET

Hyperparameters used during training PSPNet model, batch size is set to 16, primary activation function used is rectified linear unit (ReLu) [8], learning rate is set to 0.00001, optimizer used is Adam, and loss used is 'binary_crossentropy'.

6.4 XGB Classifier

XGBoost decision tree [14] [15] is a more traditional machine learning algorithm that uses aggregate image features for training. The algorithm is used as the fourth model in this study for comparing its performance with that of other three methods.

The data in training images consists of locations of objects that are to be detected. Every image is divided in several grids and the best grid is chosen with a view to improve the classification score. Jaccard index is used for these comparisons. The feature vector is formed for every grid, where the included features are mean, variance, skewness, and kurtosis. These four features are extracted for every channel. Since there are 10 objects of interest, 10 trees are created to determine if a label should be attached to an image. The trees have had a maximum depth of 5 with 100 estimators per label.

After carrying out all analysis, it is found that training one separate CNN for every object class achieves much higher accuracy than training a single CNN for all the 10 object classes.

7. EVALUATION METRICS

For evaluating our classification result, we are using two metrics as follows:

7.1 Pixel Accuracy

The pixel accuracy assesses our outcomes by simply identifying the number of pixels which were effectively classified in an image. The pixel accuracy is ordinarily revealed per class and for overall classes.

When per-class pixel is taken into consideration we're basically evaluating the binary mask; where true positive outcomes pixels of an image that are correctly classified for the given class when it is compared with ground truth mask and true negative outcomes pixels of an image that are correctly classified but does not belong to the given class.

$$Pixel\ Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

The evaluation metric used can give wrong results sometimes when the class object present is small in the image. Model efficiency is tested well, when there is negative case i.e. when class is not present in an image.

7.2 Jaccard Index

The Jaccard index is defined as similarity measure between a limited number of sets which is also known as Intersection Over Union. Intersection Over Union is a statistic measure

used for comparing the similarity and diversity measure between two sets A and B which can defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

$$0 \leq J(A, B) \leq 1$$

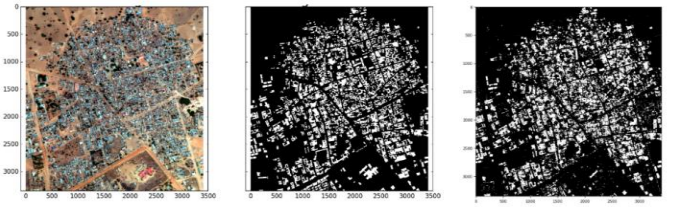
To evaluate the performance on our algorithm of all the labels, we must calculate Jaccard index of each labels and take the average, which is

$$\text{Score} = \sum_{i=1}^{10} Jaccard\ i$$

Overall, the problem can be viewed as a classic supervised image classification and object recognition problem with multispectral input image channels and score function with Jaccard index.

8. RESULTS AND ANALYSIS

8.1 Multispectral U-Net Architecture



Input Image Train Mask Test Predicted Mask

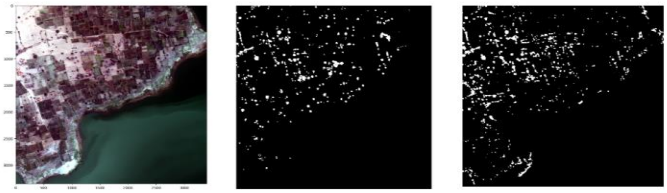
Fig. 7: Predicted mask for class 1

8.2 Inverted Pyramid Model

Input Image Train Mask Test Predicted Mask

Fig. 8: Predicted mask for class 7

8.3 XGB classifier



Input Image Train Mask Test Predicted Mask

Fig. 9: Predicted Output for class 5

7.4 Modified pyramid Scene Parsing Network - PSPNET

Input Image Train Mask Test Predicted Mask

Fig. 10: Predicted output for class 7

7.5 Comparison of predicted masks for all architectures in class 7

Fig. 11: The output performance of all architectures on an image containing class 7

Figure 11 shows the comparison of predicted masks shows output performance of all architectures on an image containing class 7 i.e. waterway, where a) Input Image, b) Train Mask, c) U-net architecture, d) Inverted Pyramid, e) PSPNet,f) XGBoost. PSPNet gives good results in terms of intersection over union when compared with ground truth mask.

The Table II shows results per class which reflects that Modified PSPNet architecture outperforms the other approaches in terms of pixel accuracy as well as mean IOU. Apart from the addition of encoder-decoder, we also changed the kernel size in the convolution layer after pyramid pooling from original 3×3 to 1×1; this saves on computational performance whilst no major difference in accuracy measure. We have also tweaked the hyperparameters viz. batch size is set to 16 and the base learning rate is set to 0.00001, rectified linear unit (ReLu) [8] is used as primary activation function instead of exponential linear unit (ELU) [9], which proved beneficial for the overall train and validation scores. The Table III shows that our approach gives better results for overall classes as compare to Vladimir Iglovikov's approach.

9. CONCLUSION

In this research, our method of introducing encoder-decoder to PSPNet improves overall computation performance and accuracy; additional changes like 1x1 convolution kernel instead of 3x3 convolution kernel is used. Rectified linear unit (ReLu) is used as primary activation function instead of exponential linear unit (ELU)which also helped in overall computation and accuracy respectively. The accuracy of training each class separately with 1 CNN is much higher than training all 10 classes at a time with 1 CNN. U-net architecture already has a heavy-weight decoder since it has the same number of parameters as its encoder. The Modified PSPNet module seems to have a better encoder- decoder, and U-Net would need additional decoder capacity.

Finally, we would like to add that successful approach to above-mentioned problems allows to significantly improve the quality of final models. Our approach includes several steps, such as the adaptation of fully convolutional networks to multispectral satellite images and evaluation of several data fusion strategies on semantic segmentation task of satellite images with joint training objective.

10. ACKNOWLEDGMENT

We take this opportunity to thanks Dr. Sharad D Gore, XHead, Department of Statistics, Savitribai Phule PuneUniversity, for his valuable guidance and providing us hissubject thought on this research topic.

Table 2: Results for each class in terms of Pixel Accuracy and Mean IOU on test data

Architectures	**Multispectral U-net**		**Inverted Pyramid**		**Modified PSPNet**		**XGBoost**	
Classes	**Pixel Accuracy**	**Mean IOU**	**Pixel Accuracy**	**Mean IOU**	**Pixel Accuracy**	**Mean IOU**	**Pixel Accuracy**	**Mean IOU**
Buildings	0.9418	0.6484	0.8957	0.6597	0.8317	0.7500	0.5296	0.4150
Structures	0.9699	0.5046	0.2095	0.0175	0.9692	0.4118	0.3434	0.0140
Road	0.9560	0.7639	0.7190	0.5450	0.8392	0.7694	0.5276	0.3440
Track	0.9528	0.5305	0.5580	0.1790	0.9311	0.5451	0.1657	0.0380
Trees	0.9644	0.6713	0.9095	0.6008	0.8644	0.7713	0.6447	0.5109
Crops	0.8734	0.8260	0.9176	0.8865	0.8566	0.8033	0.6919	0.6485
Waterway	0.9610	0.81110	0.8521	0.7709	0.9205	0.8913	0.8016	0.5739
Standing Water	0.9238	0.7087	0.8688	0.5268	0.9352	0.7123	0.6830	0.4769
Vehicle Large	0.6550	0.4000	0.3807	0.2248	0.9307	0.6069	0.24062	0.0265
Vehicle Small	0.6958	0.4978	0.3217	0.1056	0.9158	0.5940	0.2666	0.0038
	Average Pixel Accuracy	**Average IOU**	**Average Pixel Accuracy**	**Average IOU**	**Average Pixel Accuracy**	**Average IOU**	**Average Pixel Accuracy**	**Average IOU**
All Classes	0.8893	0.6362	0.6632	0.4516	0.8994	0.6855	0.4894	0.3051

Table 3: Comparison of multispectral U-net for different classes of DSTL dataset in terms of intersection over union with our approach and Vladimir Iglovikov[16] approach.

Class	**Test data (Vladimir Iglovikov's approach)**	**Test data (Our approach)**
Buildings	0.7453	0.6484
Structures	0.1905	0.5046
Road	0.8005	0.7639
Track	0.3281	0.5305
Trees	0.5018	0.6713
Crops	0.8251	0.8260
Waterway	0.9697	0.81110
Standing Water	0.6081	0.7087
Vehicle Large	0.2964	0.4000
Vehicle Small	0.0186	0.4978
	Average IOU	**Average IOU**
All classes	0.52841	0.63623

11. REFERENCES

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

[2] Muhammad JaleedKhan,"Automatic Target Detection in Satellite Images using Deep Learning", Article in Journal of Space Technology, July 2017.

[3] Chen,L., Papandreou, G., Schroff, F., Adam, H., "Rethinking Atrous Convolution for Semantic Image Segmentation", Dec 2017

[4] Ciresan, D., Giusti, A., Gambardella, L., Schmidhuber, J., "Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images"

[5] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", Jan 2016.

[6] C. Lee, K. Won oh, H. Kim, "Comparison of faster R-CNN models for object detection", 2016 16th International Conference on Control, Automation and Systems, 16–19, 2016 in HICO.

[7] Olaf Ronneberger, Philipp Fischer, Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", May 2015.

[8] Abien Fred M. Agarap, "Deep Learning using Rectified Linear Units (ReLU)", Mar 2018

[9] Djork-ArnéClevert, Thomas Unterthiner, Sepp Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)", Feb 2016.

[10] RyuheiHamaguchi, Ajito Fujita, "Effective Use of Dilated Convolutions for Segmenting Small Object Instances in Remote Sensing Imagery".

[11] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, Jiaya Jia, "Pyramid Scene Parsing Network", April 2017.

[12] https://www.doc.ic.ac.uk/~js4416/163/website/autoencoders/

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", Dec 2015.

[14] Tianqi Chen, Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System", Jun 2016.

[15] Anne He, Jennifer He, Richard Kim, "An Ensemble-Based Approach for Classification of High-Resolution Satellite Imagery of the Amazon Basin", July 2017.

[16] Vladimir Iglovikov, Sergey Mushinskiy, Vladimir Osin, "Satellite Imagery Feature Detection using Deep Convolutional Neural Network: A Kaggle Competition"

[17] https://github.com/danzelmo

[18] Akhilesh Kakade, JaysidhDumbali, "Identification of nerve in ultrasound images using U-net architecture", 2018 International Conference on Communication information and Computing Technology (ICCICT), Feb 2018.

[19] https://www.satimagingcorp.com/satellite-sensors/worldview-3/

