

International Space Station Image Extraction from a Dynamic Environment using Deep Learning

Jian-Feng Shi¹, Steve Ulrich¹, Stephane Ruel²

¹Mechanical and Aerospace Engineering, Carleton University
3135 Mackenzie, 1125 Colonel By Dr., Ottawa, Canada
jianfeng.shi@carleton.ca; steve.ulrich@carleton.ca

²Neptec Design Group Ltd.
302 Legget Drive Suite 202, Kanata, Canada
sruel@neptec.com

Abstract - This paper investigates the use of convolutional neural networks for the purpose of image foreground extraction from a dynamic environment. The proposed solution utilises the latest developments in image segmentation using pixel-wise classification to produce foreground target extraction for real-time operations. A collection of spacecraft images were assembled for network training and evaluation. The proposed technique takes advantage of transfer learning for the stable training of a convolutional neural network classifier. The image extraction software was applied to a thermal camera video, taken by an undocking spacecraft from the International Space Station. The results show the proposed deep learning-based image extraction has advantages over traditional background subtraction methods. This investigation provides evidence that semantic segmentation using convolutional neural network can be an effective tool for spacecraft image isolation and extraction from a dynamically cluttered scene.

Keywords: CNN, Semantic Segmentation, Background Subtraction

1. Introduction

Satellite robotic servicing depends on proximity sensors such as cameras and lidars for Guidance Navigation and Control (GNC). Camera systems are usually less expensive, lighter and less power intensive than their lidar counterparts. However, camera images are also less precise and less reliable partly due to sensor specifications, extreme lighting environments and a lack of image features from the scene. To this end, intelligent image recognition is an important area of research to accommodate these external and hardware shortcomings. This investigation focuses on the extraction of spacecraft image from a moving Earth background. Its main contribution is the adaptation and improvement of the latest methods in *Convolutional Neural Network* (CNN or ConvNets) for image classification and semantic segmentation of spacecrafts. Deep learning refers to a variety of deep neural networks capable of image analysis and other data processing applications [1]. ConvNets are a class of network models extending Fukushima's *neocognitron* [2] that allows the network to organise into the hierarchy model of the visual nervous system. Early ConvNet models introduced by LeCun et al. [3] was unable to reach its full potentials due to computing limitations. Twenty-three years later, Krizhevsky et al. [4] finally demonstrated ConvNet's superiority over non-network classifiers by training with millions of images using Graphic Processing Units (GPUs). Since then, Zeiler and Fergus [5] showed kernel weights learned on large datasets outperform hand-crafted ones and can be transferred to modified networks [6]. With the development of deeper [7, 8], and wider [9–11] networks, ConvNets have become the engine of choice for image recognition, object localisation [12–14], and image segmentation [15–21]. This investigation evaluates applicable networks and latest techniques in ConvNet-based image segmentation to extract the target vehicle from a dynamic environment. This paper is organised as follows: Sec. 2 provides the related work on background subtraction, image classification and segmentation. Section 3 provides the proposed method and network models. Section 4 provides the datasets and metrics used for training and evaluation. Section 5 provides the results and comparisons with other background subtraction techniques. Finally, Sec. 6 concludes this investigation.

2. Related Work

Image-based GNC algorithms rely on analysing targets and landmarks. Extracting these regions from a cluttered background will save time and improve estimation quality by minimising falsehoods. A common method of obtaining regions of interest is by removing the background. This section discusses methods for background subtraction and the alternative approach of using semantically segmented image generated by ConvNets.

2.1. Background Subtraction

There have been several studies summarising background subtraction techniques [22, 23]. A simple and effective approach is to use a first order recursive filter by Heikkilä and Silvén [24] (ADP) in the form of $B_{k+1} = (1 - \alpha)B_k + \alpha I_k$, where α is an adaptation coefficient, B_k is the adaptive background image and I_k is the incoming frame. The difference frame can be further processed by applying Otsu's threshold [25]. This adaptive scheme allows distinction of active foreground pixels from inactive background ones. While such approach is simple and effective in distinguishing a static background, it falls short in a dynamic one. The OpenCV 3.2.0 library provides several other background subtraction functions, these are the MOG method based on an adaptive Gaussian Mixture Model (GMM) [26] and two enhancements to the GMM method by Zivkovic et al. denoted by MOG2 [27] and KNN [28]. Finally, GMG denotes a method by Godbehere [29] combining statistical background image estimation, per-pixel Bayesian segmentation, and an approximate solution to the multi-target tracking problem.

2.2. CNN Image Classifiers

A major milestone for ConvNets was reached when Krizhevsky's *AlexNet* [4] significantly advanced classification accuracy on the Imagenet Large Scale Visual Recognition Challenge (ILSVRC)-2012 dataset [30]. Simonyan and Zisserman [7] increased network depth with the VGG model and demonstrated a 3×3 receptive field is equivalent to a larger 7×7 kernels when convolutional layers are increased. The smaller kernel reduces the number of parameters in the network allowing more convolutional layers, and the VGG results show large improvements over AlexNet on ILSVRC-2012. Ronneberger et al. [17] presents a mirrored decoding network forming a U-shape network (*UNet*) to segment cell images. Variations to this approach are adapted by this study to segment International Space Station (ISS) images. Redmond et al. [31] used *Darknet* [31] in the *You Only Look Once* (YOLO) system for object localisation. Darknet is similar to VGG but it oscillates between convolutional kernels of one and three dimensions. AlexNet [4], UNet [17], VGG [7], and Darknet [31] are manageable networks that is the primary focus of this study. ConvNets have also widened for higher performance such as the inception modules in *GoogLeNet* [9]. State-of-the-art classifier network such as ResNet [8] increased network depth to 152-residual layers in comparison to the maximum 8-layers AlexNet, 19-layers VGG network and 22-layers GoogLeNet. Future work may merge more sophisticated networks such as *Inception-ResNet* [10] and *Wide Residual Networks* (WRN) [11] for improved performance.

2.3. Image Segmentation

Image segmentation is a core problem in computer vision, early image segmentation work investigated edge contours [32], colour [33] and texture based classification [34]. Recent investigations include optical flow [35], and salient object detection [36]. *Semantic segmentation requires* the assignment of labels to each individual pixels, these labels can be used to extract foreground objects as an alternative to background subtraction; consequently, ConvNets are well suited for this task [15–21]. The approach used in this study stems from a family of works under the *Fully Convolutional Network* (FCN) scheme first proposed by Long et al. [16, 37] and later improved upon by Ronneberger et al. [17] (*UNet*), Noh et al. [18] (*DeconvNet*), and Badrinarayanan et al. [19] (*Segnet*). The FCN approach uses a decoding network to generate an annotation map that assigns class labels to every image pixel.

3. Model and Methods

This study is carried out in two phases to achieve the final goal of spacecraft image extraction. In the first phase, several network models were compared on conventional image datasets and on a space image dataset. The purpose of this phase is to gain an understanding of the network behaviour; to tune the hyper parameters used by the network; to train an initial set of encoding weights and biases for the phase two study; and to measure network performance on object classification. In the second phase, three out of the five networks from phase one were selected to generate segmentation

images. A mirrordecoding segment is added to the encoding network while the fully connected dense layers are removed; the resulting network becomes a FCN. A set of ISS images with manual segmentation masks are used for training in the second phase. The encoding and decoding network generates the semantic segmentation map during inference and saves it to disk. The full training and evaluation pipeline are shown in Fig. 1. All networks were developed using *TensorFlow 1:0:1* and *Python 3:5:2*.

3.1. Phase 1: Image Classification

Five network models were evaluated for their classification performance on conventional and space image. Namely, the networks are: AlexNet-5, AlexNet-8, UNet-8, VGG-19, and Darknet-21. The number after each network denotes the number of encoder convolutional and fully connected layers. All networks use the standard maxpooling method to downsample the feature map after each convolution groups. Batch normalisation [38] is applied after each convolutional layer followed by leaky Rectified Linear Units (ReLU) [39] activation functions using a slope of 0:2. Backpropagation is performed using stochastic gradient descent (SGD) with a initial learning rate of 0:1 (VGG and Darknet initial learning rate is 0:05) and learning rate decay of 0:1 after every 20_103 iterations. Dropout rate is set to 0:9, training and validation batch size is 128, and convolution stride is 1 unless otherwise

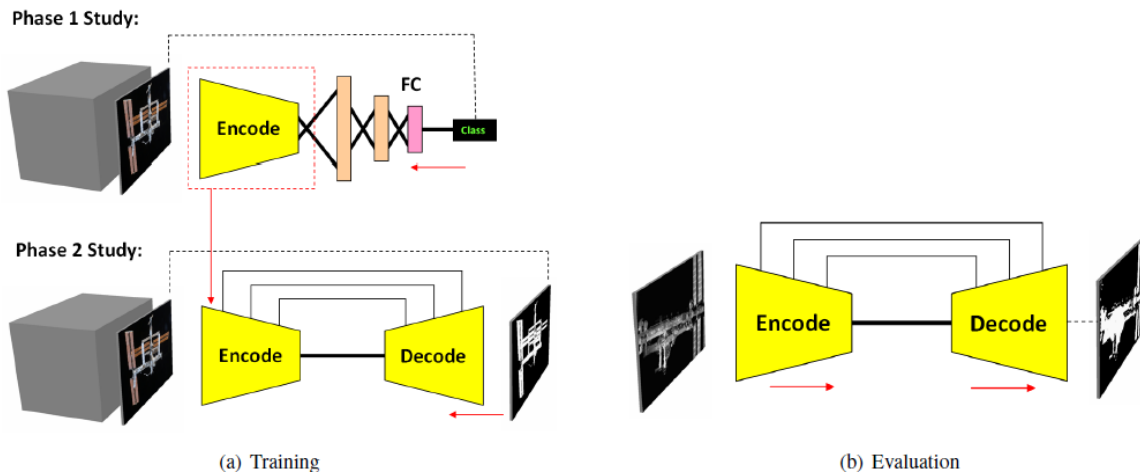


Fig. 1: Training and evaluation pipeline. For details of the encoding network see Table 1. In the phase 1 study, the network is trained for single image classification. In the phase 2 study training process, the pretrained encoding network weights and biases including the batch normalisation parameters are transferred from phase 1. The encoder weights and biases are then frozen while phase 2 focuses training of the decoder filter weights, biases, and batch normalisation parameters. A batch of 128 images forms the input tensor and is fed through the FCN with labeled mask set as targets. Class for each pixel is generated and the final loss is the sum of all pixel classification losses. Feature map tensors from the first convolutional layer are skipped across from the encoder to be concatenated with the decoder input to maintain structure. On inference evaluation, the FCN runs forward to generate the semantic segmentation map.

specified. The detail architecture of each network is provided in Table 1.

The original AlexNet [4] used ReLU [40] as activation functions instead of the traditional sigmoid function. It also used the drop-out [41] technique to minimise overfitting. In this implementation, Krizhevsky [4] splits the convolutional channels so they may run independently on two GPUs and he added Local Response Normalisation (LRN) to each convolutional layer for brightness normalisation. *AlexNet-5* is a simplified version of the vanilla AlexNet, it contains only two convolutional layers and three fully connected dense layers with max pooling and LRN implemented after each convolutional layer. This lightweight structured network has low memory needs which makes it ideal in the testing and analysis of small datasets like CIFAR-10 [42]. *AlexNet-8* is a modified AlexNet with batch normalisation and leaky ReLU activation incorporated into all convolutional layers. GPU specific convolutional layers and LRN were omitted in this network as they provide no added value to the design [7]. The five convolutional layers steps down the input image from 120×120 to 30×30 to 15×15 to 8×8 using max pooling at the end of each convolutional layer groups. *AlexNet-8* is the only network that uses a 4 pixel stride in the first convolutional layer resulting in the largest reduction of feature map dimensions after it.

Ronneberger et al. [17] presents a five group encoder-decoder FCN each with two convolutional layers and maxpooling at the end of each group for the first four groups. A modified three group UNet was developed for this study with only one convolutional layer in the first and last group to reduced memory, this resulted in 5 layers of convolutional layers with 3 fully connected layers, hence the designation of UNet-8. The UNet structure also takes advantage of skip layers to maintain feature map structure when generating the semantic segmentation map, more details on the full encoder-decoder FCN shall be discussed in Sec. 5.2. *VGG-19* [7] is one of the most popular networks used by the vision community in the past two years. Its size falls between AlexNet and GoogLeNet which allows it to enjoy sufficient accuracy and ease of use. The original *VGG-19* is made from two groups of two convolutional layers with 64 and 128 channels, and three groups of four convolutional layers with 256, 512 and 512 channels. A modified version of *VGG-19* was used in this study, the ConvNet channels were reduced to 16 – 32 – 64 – 128 – 256 for each ConvNet group respectively to reduced memory.

The initial version of *Darknet* [31] consists of seven groups with 24 convolutional layers, it contains one convolutional layer in the first two groups and max pooling in the first four groups. The YOLO-9000 Darknet [14] is simplified to six groups with first five groups ending in max pooling. The YOLO system also contains one fully connected layer and one detection tensor. This detection tensor contains bounding box information and is used to localise objects in the image. The modified Darknet used in this study omits the detection layer and use the same fully connected dense layer structure as the previously defined networks. The Darknet was inspired by GoogLeNet [9], however, it replaces the inception modules with three to five alternating 1×1 and 3×3 kernels based on the *network-in-network* concept [43]. All the evaluated networks are connected to three fully connected dense layer at the end of the convolutional groups for classification. *AlexNet-5*, *UNet-8*, and *VGG-19* using roughly 400 and 200 nodes for the first two layers while *AlexNet-8* and *Darknet-21* using roughly 200 and 100 nodes.

Due to the small datasets used in the training, the fully connected (FC) nodes was reduced by factor of 10 – 20 from the original networks. The number of nodes in the final *SoftMax* layer equals the total number of classes in the dataset.

Table 1: Network architectures. The first three numbers represents height, width, and channel of each feature layer, -sign indicate Batch Normalisation (BN) and Leaky ReLU (LR) combination layers. +sign indicate BN, LR and Drop Out (DO) combination layers. Value after "c" indicates square kernel size, value after "s" indicates the stride, value after "w" indicates the window size, FC indicates Fully Connected, LRN indicates Local Response Normalisation.

<i>AlexNet-5</i>	<i>AlexNet-8</i>	<i>UNet-8</i>	<i>VGG-19</i>	<i>Darknet-21</i>
input(120×120×3 RGB image)				
120×120×64-c5-	30×30×48-c11-s4-	120×120×16-c11-	120×120×16-c3- 120×120×16-c3-	120×120×16-c7-
maxpool-w3s2/LRN	maxpool-w2s2			
60×60×64-c5-	15×15×128-c5-	60×60×32-c5- 60×60×32-c5-	60×60×32-c3- 60×60×32-c3-	60×60×32-c3-
LRN /maxpool-w3s2	maxpool-w2s2			
	8×8×192-c3- 8×8×192-c3- 8×8×128-c3-	30×30×64-c3- 30×30×64-c3-	30×30×64-c3- 30×30×64-c3- 30×30×64-c3-	30×30×64-c3- 30×30×32-c1- 30×30×64-c3-
	maxpool-w2s2			
			15×15×128-c3- 15×15×128-c3- 15×15×128-c3- 15×15×128-c3-	15×15×128-c3- 15×15×64-c1- 15×15×128-c3-
	maxpool-w2s2			
			8×8×256-c3- 8×8×256-c3- 8×8×256-c3- 8×8×256-c3-	8×8×256-c3- 8×8×128-c1- 8×8×256-c3- 8×8×128-c1- 8×8×256-c3-
	maxpool-w2s2			
				4×4×512-c3- 4×4×256-c1- 4×4×512-c3- 4×4×256-c1- 4×4×512-c3-
FC-384+	FC-205+	FC-384+	FC-410+	FC-205+
FC-192+	FC-102+	FC-192+	FC-204+	FC-102+

FC-(Number of Classes)/SoftMax

3.2. Phase 2: Image Segmentation

Long *et al.* [16] demonstrated semantic segmentation map can be generated using trained FCN which only consists of convolutional layers of various shapes and channel depth. In addition to the encoder downsampling of the traditional classifier networks, these FCN also includes upsampling decoder layers. Both DeconvNet and Segnet use mirror image VGG as the decoder network. The same mirroring approach was followed by this investigation. Reverse max pooling can be achieved by restoring the forward max pooling coordinates in the upsampling layer. A more efficient approach is to let the strided convolution learn its own spatial upsample [44]. The proposed method also adapts the *skip feature* concept from UNet to allow structured generation [45]. Skip features are feature maps that "skipped" across from the encoder side to concatenate with decoding map of roughly the same size. In the case where the input feature map does not match exactly with the decoder side input, the encoder side map is randomly cropped. This cropping, however, does not typically exceed ten pixels and is only one to two pixels in the deep channel layers. To train the FCN for semantic segmentation, pixel-wise class errors from the ground truth labels are summed as one loss value to be minimised during backpropagation.

4. Experimental Data

Two datasets were used to train and validate the selected networks, these are the CalTech-101 [46] and the Space-5 dataset. The CalTech dataset consists of 102 category objects provided by 9;144 images of various sizes with a typical resolution of 300×200. For this study, 10 percent of the database was randomly selected for evaluation and the rest for training. The CalTech dataset is compact and easy to handle for classification validation, although it only provides one target object per image and is not effective for segmentation evaluation. Future studies may include dataset such Pascal VOC [47] or Cityscapes [48] to gain a generic quantitative measurement of the proposed networks; however, to apply a network to spacecraft segmentation even an ImageNet [30] trained network is insufficient and requires space vehicle specific images for transfer learning.

4.1. Spacecraft Image Dataset

While there are numerous datasets available for variety of studies, there is a lack of image dataset for space system development. For the purpose of this study, a new spacecraft image dataset called *Space-5* was developed using a mixture of 4,237 synthetic and real images with five object categories: Earth (230 images), ISS (2,590 images), Spacecraft (193 images), Envisat (301 images), and the RadarSat Model (RSM) (421 images). Sample images of each class category are provided in Fig. 2. Segmentation annotation is created for ISS and the Earth using manual image segmentation techniques. To evaluate target vehicle image extraction, a Neptec TRIDAR infrared (IR) video of the Space Shuttle STS-135 mission undocking sequence was used. In this sequence, the Space Shuttle performs a flyby of the ISS while imaging it with the rotating Earth in the background. For segmentation training and validation, the Space-5 dataset is further reduced to a subset of ISS and Earth (Space-2) images. Additional evaluation images were added to Space-2, including 10 Earth only images and 141 ISS only images from the STS-135 mission docking sequence.

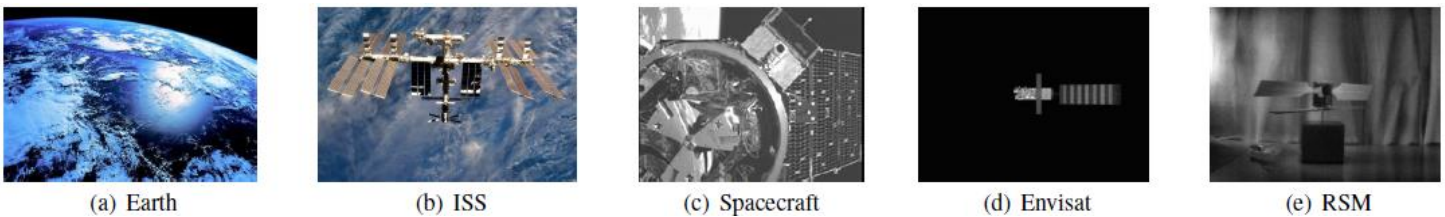


Fig. 2: Sample images from the Space-5 database.

4.2. Segmentation Metric

The standard *Jaccard index*, also known as *Intersect over Union* (IoU), was used to measure segmentation performance. The IoU for class i is defined as $J_i = |x_i \cap y_i| / |x_i \cup y_i|$, where x is the ground truth pixel labels and y is the predicted pixel labels. In terms of Receiver Operating Characteristics (ROC) [49], single class IoU is equivalent to $J_i = TP_i / (EP_i + FN_i)$ where TP is the true positive, and EP is the estimated positive which is the sum of TP and the false positive (FP). The false negative (FN) can be computed as the actual positive (AP) minus TP . The ROC accuracy, recall and precision metrics are defined respectively as $Acry_i = (TP_i + TN_i) / n_p$, $Rcal_i = TP_i / AP_i$, $Prec_i = TP_i / EP_i$, where n_p is the total

number of pixels in the image. The pixel accuracy defined by Long *et al.* [16] is equivalent to the ROC precision. In addition to the aforementioned metrics, the total and mean pixel accuracy, total and *mean IoU*, and *frequency weighted IoU* (J_{fw}) from Long *et al.* [16] are also computed. The total pixel accuracy is defined as $Prec = \sum_i TP_i / \sum_i EP_i$, the mean pixel accuracy is defined as $Prec = \sum_i Prec_i / n_{CL}$, where n_{CL} is the total number of class objects. The mean IoU is $J = \sum_i J_i / n_{CL}$ and J_{fw} is defined as $J_{fw} = \sum_i \frac{EPTP_i}{EP_i + AP_i - TP_i} / \sum_i EP_i$. Finally, the mean accuracy (*Acry*) and recall (*Rcal*) are averages over the number of classes.

5. Results and Discussions

This study was carried out in two phases. In the first phase, various networks were studied for their performance and behaviour using the CalTech-101 and Space-5/2 datasets. In the second phase, three networks were selected to perform semantic segmentation of the ISS image. The standard technique of random cropping, image flipping, brightness, saturation and contrast variation were applied on the training image to inflate the sample set and avoid overfitting. The input images were then normalised by pixel intensity, mean centered, and scaled to unit standard deviation.

5.1. Phase 1: Image Classification

Results of the network classification for the CalTech-101 and Space-5 dataset are provided in Table 2. The Space-2 dataset validation resulted in 100 percent accuracy for all five networks. The CalTech-101 and Space-5/2 dataset training were all conducted with 60e3 batch iterations. Accuracy is lower for larger class datasets, this is partly due to the low number of images available for training. Zeiler [5] propose to train deep ConvNets on high volume datasets such as ImageNet to avoid overfitting and to better develop feature extraction filters. The effects of dropout on the convolutional layers was studied, results show applying dropouts only to fully connected layers resulted in roughly 4 percent increase in accuracy. In general, Darknet outperforms the other networks for all datasets.

5.2. Phase 2: Image Segmentation

While Darknet exhibited highest performance in the image classification task, it was more complex and memory intensive to construct a mirroring decoding network. By contrast, AlexNet-5 was too shallow; therefore, *AlexNet-8*, *UNet-8*

Table 2: Classification Results.+sign indicate dropout added to convolutional layers.

Database	<i>AlexNet-5</i>	<i>AlexNet-8</i>	<i>UNet-8</i>	<i>VGG-19</i>	<i>Darknet-21</i>
CalTech-101+	56.6	58.9	48.8	57.0	56.1
CalTech-101	57.9	60.0	54.9	59.3	60.4
Space-5+	96.5	96.1	94.9	94.9	95.7
Space-5	97.3	97.1	97.3	96.5	97.7

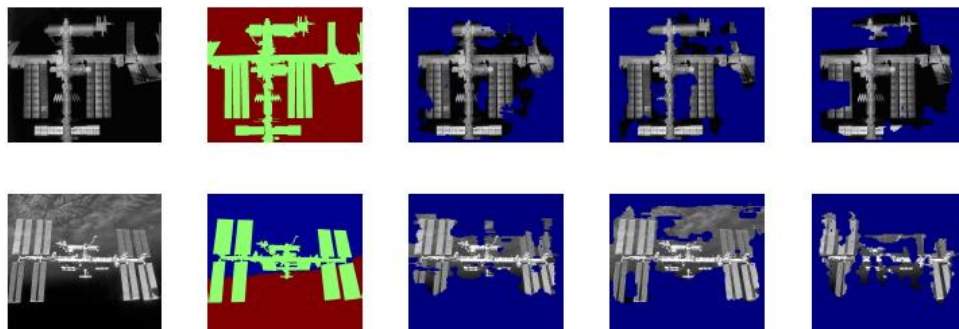


Fig. 3: Semantic segmentation of the ISS. Top: IR camera image, bottom: photo camera image. Left to right: original, ground truth (https://github.com/ai-automata/170627_CNN_Segmentation), *AlexNet-8*, *UNet-8* and *VGG-19*.

and *VGG-19* were selected for the phase 2 validation. The encoder weights of phase 1 were trained to 60×10^3 iterations (*i.e.* 2,727 epochs). All attempts to simultaneously train encoder and decoder network from zero initialisation resulted in instability. The convergence in phase 2 was much slower than Phase 1 partly due to a very small learning rate of

10^{-8} (*UNet-8* learning rate was reduced to 0.5×10^{-8} at the mid-point of training to avoid instability). The learning rate magnitude must sustain a stable gradient, which in phase 2, is defined as the sum of all the pixel losses. Learning decay was omitted since the loss profile remains shallow but constant. All networks were trained for 600×10^3 iterations (*i.e.* 27,273 epochs) on separate GPUs (CPU: AMD X8 FX-8350 (125W) 8-Core Sock at AM3+ 4GHz, GPU-1: GeForce GTX-1080 Ti Founders Edition 11 GB, GPU-2: GeForce GTX-1080 AMP!Extreme Edition 8 GB). Phase 1 training was on the order of four to six hours for each network while Phase 2 training took roughly five days.

The segmentation results of two ISS images are provided in Fig. 3 and the evaluation metrics are provided in Table 3. Results show while all networks located the primary target body, *UNet-8* produces the most accurate result. Earth prediction results are much lower than ISS due to a smaller and coarse training set. To improve network accuracy one may include hand crafted annotation labels; increase image volume (*i.e.* Imagenet); refine hyper parameter tuning; and increase input resolution. Furthermore, deeper networks in combination with scale and orientation invariance can be implemented to improve accuracy.

While it is visually evident *UNet-8* outperforms the other networks in the IR image, it is not immediately intuitive for UNet’s higher performance in the photo image. Superficial observation of Fig. 3 suggest AlexNet better predicts the ISS. This discrepancy can be explained by considering all of the classes in the scene since the presented metrics are the average. As result, while the true positive of ISS in the UNet image is low, the union of space pixels are also low and cause the space IoU to be higher and the average metric in UNet to outperform AlexNet and VGG. The ISS only pixel precision of 0.63, 0.52, and 0.57 respective to AlexNet, UNet, and VGG confirms UNet as the worst ISS predictor. The above discussion highlights the importance in the correct interpretation of the metric results. In actual operations, the ISS prediction is far more important than Earth and space; therefore, it is more suitable to develop a weighted accuracy and IoU metric based on operational needs. Timing wise, *AlexNet-8* is the fastest while the relatively deeper *VGG-19* is the most computationally expensive. GPU inference can be 12, 37 and 55 times faster than CPU for AlexNet, UNet, and VGG respectively. Qualitative results of ISS extraction is provided in Fig. 4. This figure shows the original video sequence, a manual segmentation based on edge detection and dilation, all *OpenCV* 3.2.0 background subtraction methods, the adaptive method, and using ConvNets. Contrary to ConvNets, background subtraction is mostly dependent on relative motion which can restrict their use. The best ConvNet results are frame 1, 164, and 219; the worst is frame 55. In frame 55, the pixel intensity of the ISS and Earth are very close, so a deep network with more distinctive features will perform better. Evidently, Fig. 4 shows the deeper *VGG-19* model outperforms the other networks.

Table 3: Forward inference timing and semantic segmentation metric comparisons. Timing is an average of all evaluation images, semantic segmentation metric is based on the IR and the photo ISS image only.

Hdwr.	\bar{t} (ms)	Network Model	Image	\overline{Acry}	\overline{Rcal}	\overline{Prec}	\overline{Prec}	\overline{J}	J_{fw}
CPU	11.80	<i>AlexNet-8</i>	IR	0.68	0.67	0.71	0.73	0.50	0.50
	37.15	<i>UNet-8</i>		0.78	0.75	0.82	0.82	0.63	0.63
	61.93	<i>VGG-19</i>		0.59	0.58	0.61	0.64	0.40	0.40
GPU-1	0.58	<i>AlexNet-8</i>	Photo	0.68	0.58	0.53	0.50	0.33	0.49
	0.77	<i>UNet-8</i>		0.72	0.64	0.59	0.71	0.38	0.56
	1.13	<i>VGG-19</i>		0.63	0.48	0.44	0.41	0.25	0.38

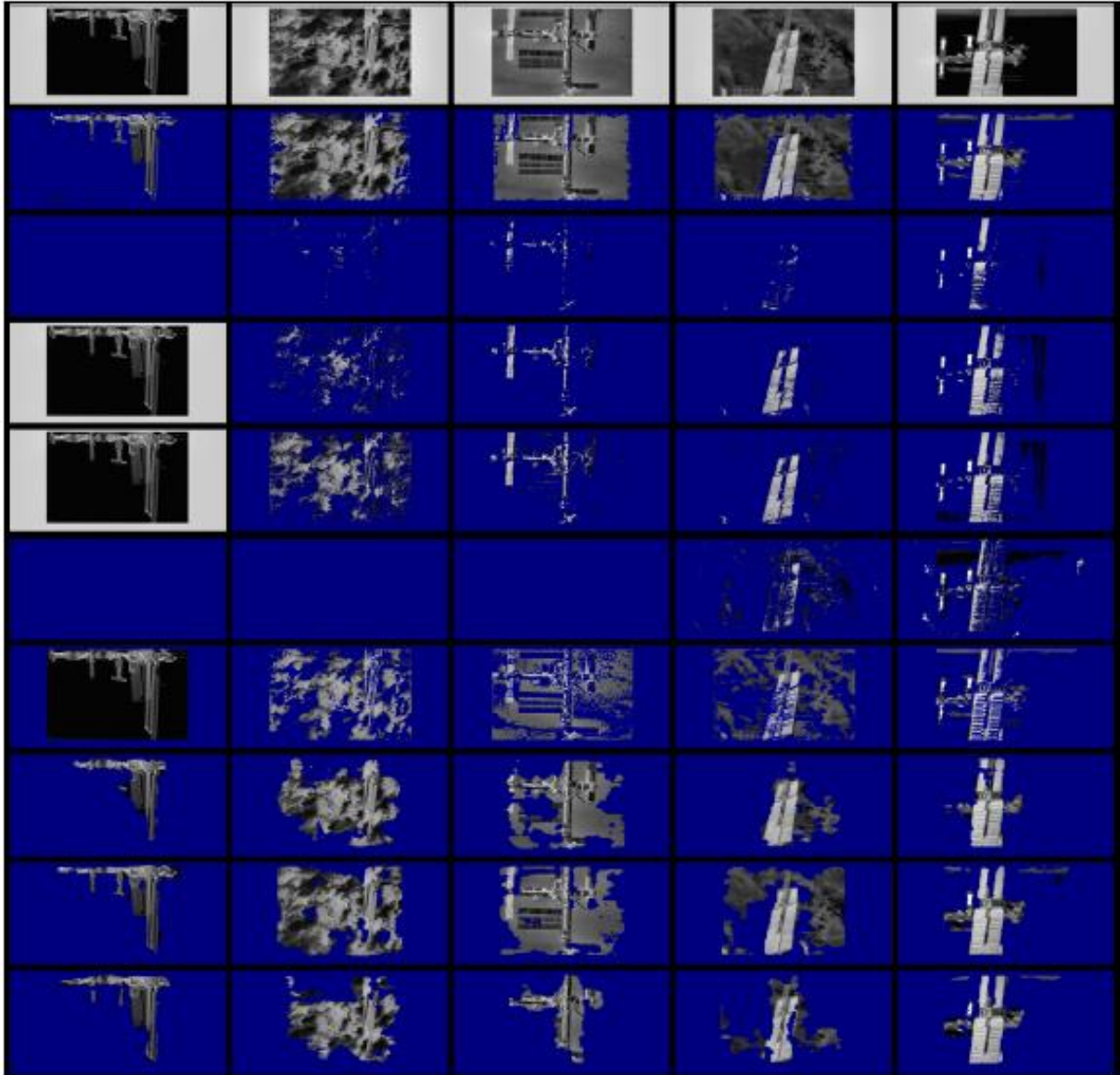


Fig. 4: Background subtraction results, images are taken from STS-135 mission undocking segment with dynamic Earth background environment. Frames selected for presentation: start (frame 1), quarter way (frame 55), half way (frame 110), three-quarter way (frame 164), and end of video sequence (frame 219). Method as follows: level 1-original , level 2-manual , level 3-MOG [26] , level 4-MOG2 [27] , level 5-KNN [28] , level 6-GMG [29] , level 7-ADP [24] , level 8-AlexNet-8 , level 9-UNet-8 , level 10-VGG-19 MOG, MOG2, KNN, and ADP methods require initialisation on the first frame, GMG method requires 120 frames to initialise. (<https://youtu.be/iBZKGy-6yJE>).

6. Conclusion

In conclusion, five CNN networks have been evaluated for ISS image extraction from a dynamic background. The encoder-decoder mirror image network with skip layers is effective in providing general shape of the target vehicle. Future work includes a more comprehensive study of established datasets such as Pascal VOC [47] and ImageNet [30].

Acknowledgments

This research was jointly funded by the NSERC CGSD3-453738-2014, CSA STDP and the OCE VIP-II Award 24053.

References

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 52, pp. 436-444, 2015.
- [2] K. Fukushima, "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Neural Computation*, vol. 36, pp. 193-202, 1980.
- [3] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541-551, 1989.
- [4] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.
- [5] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *arXiv preprint. arXiv:1311.2901*, 2013.
- [6] M. Oquab and et al., "Learning and transferring mid-level image representations using cnn," in *CVPR*, 2014.
- [7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *arXiv preprint. arXiv:1409.1556*, 2014.
- [8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *ICCV*, 2015.
- [9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, D. Reed, S. Anguelov, D. Erhan, V. Vanhoucke, and Rabinovich, "Going deeper with convolutions," in *CVPR*, pp. 1-9, 2015.
- [10] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *arXiv preprint. arXiv:1602.07261*, 2016.
- [11] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *BMVC*, 2017.
- [12] S. Ren, K. He, and et al., "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.
- [13] W. Liu, E. C. S. C. Anguelov, D., and S. Reed, "Ssd: single shot multibox detector," in *CoRR*, vol. abs/1512.02325, 2015.
- [14] J. Redmon and A. Farhadi, "Yolo9000: Better, faster stronger," *arXiv:1612.08242*, 2016.
- [15] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *CVPR*, 2016, pp. 3150-3158.
- [16] J. Long, E. Shelhamer, and T. Darrel, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431-3440.
- [17] O. Ronneberger and et al., "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234-241.
- [18] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *ICCV*, 2015.
- [19] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," in *CoRR*, vol. abs/1505.07293, 2015.
- [20] A. Dosovitskiy and et al., "Learning to generate chairs, tables and cars with convolutional networks," *PAMI*, 2016.
- [21] J. Yang, B. Price, and et al., "Object contour detection with a fully convolutional encoder-decoder network," in *CVPR*, 2016.
- [22] A. McIvor, "Background subtraction techniques," in *Proc. of Image and Vision Computing*, vol. 4, pp. 3099-3104, 2000.
- [23] D. Tripathy and K. Guru Raghavendra Reddy, "Adaptive threshold background subtraction for detecting moving object on conveyor belt," *Intl. Journal of Indestructible Mathematics and Computing*, vol. 1, no. 1, pp. 41-46, 2017.
- [24] J. Heikkila and O. Silven, "A real-time system for monitoring of cyclists and pedestrians," *JIVC*, pp. 563-570, 1999.
- [25] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, 1975.
- [26] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Proc. of the 2nd European Workshop on Advanced Video Based Surveillance Systems*, 2001.
- [27] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *ICPR*, 2004.
- [28] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773-780, 2006.
- [29] A. Godbehere, A. Matsukawa, and K. Goldberg, "Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation," in *American Control Conference*, pp. 4305-4312, 2012.
- [30] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

- [31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.
- [32] P. Arbelaez and et al., "Contour detection and hierarchical image segmentation," *PAMI*, vol. 33, no. 5, pp. 898-916, 2010.
- [33] S. Belongie, C. Carson, H. Greenspan, and M. J., "Color- and texture-based image segmentation using em and its application to content-based image retrieval," in *ICCV*, pp. 675-682, 1998.
- [34] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *IJCV*, vol. 81, no. 1, pp. 2-23, 2009.
- [35] Y. Tsai, M. Yang, and M. Black, "Video segmentation via object flow," in *CVPR*, 2016.
- [36] W. Tu, S. He, and et al., "Real-time salient object detection with a minimum spanning tree," in *CVPR*, 2016, pp. 2334-2342.
- [37] E. Shelhamer and et al., "Fully convolutional networks for semantic segmentation," *PAMI*, vol. 39, no. 4, pp. 640-651, 2017.
- [38] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *arXiv preprint. arXiv:1502.03167*, 2015.
- [39] A. Maas, A. Hannun, and A. Ng, "Rectifier nonlinearities improve neural network acoustic models," *ICML*, vol. 30, no. 1, 2013.
- [40] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines," *ICML*, 2010.
- [41] G. Hinton, N. Srivastava, A. Krizhevsky, and R. Sutskever, I. Salakhutdinov, "Improving neural networks by preventing coadaptation of feature detectors," in *arXiv preprint. arXiv:1207.0580*, 2012.
- [42] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," in *Univ. of Toronto Tech. Report*, 2009.
- [43] M. Lin, Q. Chen, and S. Yan, "Network in network," in *CoRR. abs/1312.4400*, 2013.
- [44] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *arXiv preprint. arXiv:1511.06434*, 2015.
- [45] P. Isola, J. Zhu, T. Zhou, and A. Efros, "Image-to-image translation with conditional adversarial networks," in *arXiv preprint. arXiv:1611.07004*, 2016.
- [46] L. Fei-fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *PAMI*, vol. 28, no. 4, pp. 594-611, 2006.
- [47] M. Everingham and et al., "The pascal visual object classes (voc) challenge," *IJCV*, vol. 88, no. 2, pp. 303-338, 2010.
- [48] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.
- [49] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, pp. 861-874, 2006.