

ChangeNet: A Deep Learning Architecture for Visual Change Detection

Ashley Varghese, Jayavardhana Gubbi, Akshaya Ramaswamy, and Balamuralidhar P

Embedding Systems and Robotics, TCS Research and Innovation, Bengaluru, India

Abstract. The increasing urban population in cities necessitates the need for the development of smart cities that can offer better services to its citizens. Drone technology plays a crucial role in the smart city environment and is already involved in a number of functions in smart cities such as traffic control and construction monitoring. A major challenge in fast growing cities is the encroachment of public spaces. A robotic solution using visual change detection can be used for such purposes. For the detection of encroachment, a drone can monitor outdoor urban areas over a period of time to infer the visual changes. Visual change detection is a higher level inference task that aims at accurately identifying variations between a reference image (historical) and a new test image depicting the current scenario. In case of images, the challenges are complex considering the variations caused by environmental conditions that are actually unchanged events. Human mind interprets the change by comparing the current status with historical data at intelligence level rather than using only visual information. In this paper, we present a deep architecture called ChangeNet for detecting changes between pairs of images and express the same semantically (label the change). A parallel deep convolutional neural network (CNN) architecture for localizing and identifying the changes between image pair has been proposed in this paper. The architecture is evaluated with VL-CMU-CD street view change detection, TSUNAMI and Google Street View (GSV) datasets that resemble drone captured images. The performance of the model for different lighting and seasonal conditions are experimented quantitatively and qualitatively. The result shows that ChangeNet outperforms the state of the art by achieving 98.3% pixel accuracy, 77.35% object based Intersection over Union (IoU) and 88.9% area under Receiver Operating Characteristics (RoC) curve.

Keywords: Change Detection, CNN

1 Introduction

Monitoring of public infrastructure in the context of smart cities to check for encroachments is an essential task. Encroachment can be described as anything placed in or on a public asset for e.g., a road, or a pavement that is essentially a Government property. Currently, manual methods are used where an officer visits and conducts a survey of area of interest. Manual investigation to assess the encroachment is a tedious task and the possibility of missing interesting events is high. This is a very time consuming process and affects the aesthetics of the city and results in loss to the exchequer in the form of fines.

Emerging micro unmanned aerial vehicles or commonly called drones can be employed for detecting such encroachments. According to a report from Tractica in 2017 [1], drones are expected to play a vital role in the smart city environment, providing support for a range of medical, transport and agriculture use cases. Drones have a tremendous amount of potential to provide a sustainable environment for the people who live in them. Thousands of drones are already being used to improve city life such as in documenting accident scenes and monitoring construction sites. As cameras are ubiquitous in drones, computer vision based autonomous monitoring using unmanned vehicles is picking up but is still immature. The challenges include navigation of drones autonomously, detecting the objects of interest and finally encroachment detection. The geo tagged images or videos have to be assessed for detection anomalies and their locations.

Scene understanding in real world scenario is a very challenging problem that has not reached the required maturity. However, detecting an encroachment can be viewed as a visual change detection problem. A historical image at the location and the current location of the drone can be used to find any deviation using visual processing. Identifying the deviation using images or videos is called visual change detection and is the focus of this paper.

Change detection in video analysis is often used as a stepping stone for high level scene understanding. In its conventional form, the methods are used for identifying changes in the background by comparing any two consecutive frames or limited to short term temporal analysis [2]. In remote sensing literature, change detection is referred to surface component alteration that is very useful in automatic land use analysis [3]. The fact that the satellite images are registered helps in pixel level change detection tasks that have been successfully extended to object level change analysis [3]. Some of the key challenges for visual change detection between any two images include variations in: lighting or illumination, contrast, quality, resolution, noise, scale, pose and occlusion. The first five attributes are experienced in any change detection scenario but the last three attributes are either not experienced in short term temporal analysis or it can be easily handled using frame dropping. Most of the methods in literature that models background pixels to detect change are in fact addressing the first five attributes. In the case of remote sensing, where change detection is widely used, change in scale, pose and occlusion are rarely seen and the above methods can be easily deployed with suitable pre-processing. Although these approaches are a part of decision making, it involves low level image analytics such as background foreground segmentation. In more complex inferencing using visual input, particularly in pattern recognition and category formation, higher level cognition is essential. For instance, when two images are being compared that have variations in pose, illumination, color information and occlusion, the methods in literature often fail due to unregistered images, pose and scale variations as well as occlusions.

Figure 1 shows an example from the VL-CMU-CD change detection dataset [4], where higher level inferencing is required to detect the rubbish dumping on the pavement and the appearance changes are spread throughout the images. In this paper, a novel deep learning architecture is proposed for change detection that targets higher level inferencing. The new network architecture involves extracting features using ResNet [5] and combining filter outputs at different levels to localize the change. Finally, detected

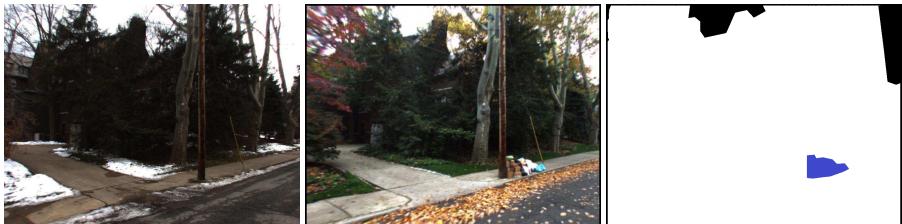


Fig. 1. Illustrative images from CMU-CD dataset: reference image, test image and ground truth (in blue)

changes are identified using the same network, and output is an object level change detection with the label. The proposed architecture is compared with the state-of-the-art using three different modern change detection dataset: VL-CMU-CD [4], Tsunami [6] and GSV [6] datasets.

2 Related work and motivation

As described in the previous section, change refers to the higher level inferencing where the appearance has substantially changed between images. The change could either be insertion or deletion of an object from the scene, or some transformation of structure of the object or scene [7]. There are numerous industrial applications that can benefit from efficient visual change detection and as a result there is plenty of work in this area, especially in satellite image processing [3]. One of the simplest approach to change detection is frame or image differencing that involves traditional pixel level analysis. However, it works if and only if both the images are registered and the variation in image attributes are relatively minimal [3].

In 2012 and 2014, Goyette *et al.* [8] and Wang *et al.* [9] developed and expanded a change detection dataset and a workshop was conducted alongside CVPR 2014. Several papers have been published using this dataset. Although the dataset is used for lower level inference tasks, these methods are highly relevant and they are reported here with advantages and pitfalls. The eleven categories of videos used in this dataset and their corresponding results throws much needed light on higher level inferences that we are repeatedly discussing in this paper. Bilodeau *et al.* [10] used local binary similarity pattern for change detection. The method is quite simple and plays with spatial neighbourhood of every pixel. Noting this, the authors have used only two of the 11 categories - baseline and thermal - for detecting change highlighting the need for a more holistic approach. Sedky *et al.* [11] propose a physics based approach called Spectral 360. They use illumination, surface spectral reflectance and spectral similarity measure to build a decision function. They report a f -Score of 67.32% with low f -score of less than 50% for four of the eleven categories. Gregorio *et al.* [12] report an improved overall accuracy of 68.12% using a weightless neural network that helps in incorporating pixel history information in decision making (very similar to background subtraction using Gaussian mixture models). Again, their performance is limited as they fail to address some of the teething issues related to scale and pose. Wang *et al.* [13] proposed a flux

tensor and split Gaussian model with a healthy f -score of 72.83%. Improving on all the above methods, a more comprehensive work has been reported by St-Charles *et al.* [2] who achieve 74.1% overall f-score using their SuBSENSE system. They propose spatio-temporal binary features to achieve the same. In line with achievements from other computer vision challenges, Bianco *et al.* [14] propose ensemble method for change detection achieving the best f -score of 78.21% by combining results of five other methods in literature. In every method discussed from basic binary features to ensemble methods, five categories out of eleven posed high challenge: PTZ, night videos, low frame rate, intermittent object motion and turbulence videos. Putting the results in perspective, small variations in illumination, contrast, quality, resolution and noise were captured by these methods quite well. Increased variations in addition to change in pose, scale and occlusion were not handled well by these methods and they are precisely the higher level inferences that are required for an object or scene level change detection method.

One interesting development in semantic change detection was reported by Gressin *et al.* [15] on satellite image processing. Although the work is on simulated data, for the first time, they have reported the perspective of change detection at different inference levels such as object, theme and database akin to our work. As far as we are aware, this is the first work alluding to different levels in change detection. In a similar work, Kataoka *et al.* [16] talks about semantic change detection by adding semantic meaning to changed area. First, they find changed area using hyper maps, and then add semantic meaning to that changed area. Since last few years, after deep learning has become the main approach in computer vision, there have been some efforts in creation of the dataset as well as in building change detection procedures. Sakurata and Okatani [6] was the first such attempt and they built two data sets with 100 image pairs known as TSUNAMI dataset and Google Street View (GSV) dataset. These are panoramic images created using street view separated temporally by several days or months. In addition to the creation of a dataset, they proposed a complex super pixel based approach that uses convolutional neural network (CNN) for feature extraction. The low resolution feature map generated from CNN network is combined with super pixel segmentation to get precise segmentation boundaries of the changes. Although deep learning is used in the pipeline, there are many other hyper parameters in the procedure that needs fine tuning for different scenarios.

Going one step further, Alcantarilla *et al.* [4] propose a network called CDnet for finding structural changes in street view video. They create a new dataset of 152 categories with 11 unique object classes called VL-CMU-CD dataset. In order to create nearly registered images, they use visual simultaneous localization and mapping (SLAM) to get the 3D point cloud and then project the points onto a 2D reference plane after determining the reference pose. It is a pixel level change detection approach and uses contraction and expansion layers for pixel level classification. The contraction block creates data representation. In this process it stores max pooling output for later use in the expansion network. The expansion block has been used for improving change localization. The proposed ChangeNet architecture is different from CDnet approach. Our network determine category of change in addition to change localization. We use parallel weight tied networks for feature extraction. It ensures both the network

learn same features from the two images. Therefore, the features from both the images can be compared easily. In addition to this, we combine output from different levels of convolution layers so that the model captures the sparse and finer details of the object. Bilinear interpolation is used in ChangeNet for upsampling the data and the filter parameters are learned in network itself. Another feature of ChangeNet is that it combines predictions from different levels of convolution layer. Such approach helps the model to capture both coarse and fine details of the object. Apart from deep learning approaches, a multi-scale super pixel approach for drone image analysis has been proposed by Gubbi *et al.* [17] with limited success on VL-CMU-CD dataset. The focus of this paper is to implement the change detection system on a computationally challenging environment.

In the recent past, there has been quite a good amount of success in pixel level image analysis using deep architecture. Bansal *et al.* [18] proposed a new architecture for predicting surface normal that is useful in 2D-3D alignment. They use pre-trained VGG-16 network for feature extraction followed by three layers of fully connected layers for predicting surface normal for every pixel. Bansal *et al.* [19] generalized their earlier work and created PixelNet architecture and demonstrated semantic segmentation and edge detection in addition to surface normal estimation using an extended VGG-16 network. Such work has demonstrated that CNN is able to learn pixel level information in addition to their success in image categorisation. Similar work have been extended for regions of interest where they propose a new network for simultaneously predicting human eye fixations and segmenting salient objects. In addition to single image pixel analysis, there has been some recent work in finding similarity between two images or signal pairs. Du *et al.* [20] proposes a Siamese CNN network for checking whether two hand written texts are written by the same person or not. Both the inputs are encoded with the same network and then concatenated output is fed to a two class classifier to determine whether handwriting is same or not.

With the developments in change detection and pixel level analysis using deep learning, we are motivated to solve the hard problem of change detection using a deep network. VL-CMU-CD dataset is our target as the scene pairs are complex and taken at different view angle, illumination and seasons as well. It has 11 different class of structural changes like construction-maintenance, bin on pavement, new sign boards, traffic cone on road, vehicles, etc., including background. To the best of our knowledge, it is a novel architecture for visual change detection particularly resulting in scene labels that can be viewed as semantic change detection. We further train the network to determine the category of change in addition to the changed area. Both the tasks happen within the network and involves single training. Most of the background information are irrelevant in our case since those changes could be due to season, illumination or view point variation. It mainly looks for changes at object level as compared to Alcantarilla *et al.* [4]. The model inputs are test and reference images. Output is detection, localization and categorization of changed region. It mainly answers the following three questions in the presence of seven variations, which have been discussed earlier: is there any change? if yes, what is the change? and where is the change in the image? Section 3 gives details about our architecture and experimental details are provided in Section 4. Result and discussion are presented in Section 5.

3 ChangeNet architecture

We propose a deep learning architecture for detecting changes between image pairs. We adapt ideas from siamese network [21, 22] and fully convolutional network (FCN) [23] to map features from image pair to visual change. Convolutional neural networks are known for its performance in object detection. Especially, networks like GoogLeNet [24], Alexnet [25], VGGNet [26], and ResNet [5] are powerful deep trained models for feature representation. The learned representation can be transferred to another domain instead of training it from scratch. Transfer learning approach is followed here for feature extraction. In this architecture, two inputs are required: a test I_{test} and a reference image I_{ref} . Both are having same dimension of $w \times h \times d$ where w and h are spatial dimension and d is the number of channels. The change detection problem can be formulated as: find a way to compare features from I_{test} and I_{ref} to assign a change class label from label set of $l = 1, 2, \dots, N$ to each element of change map $I_{w \times h}$. N is the number of defined semantic change class and it ensures environmental changes are neglected during change detection.

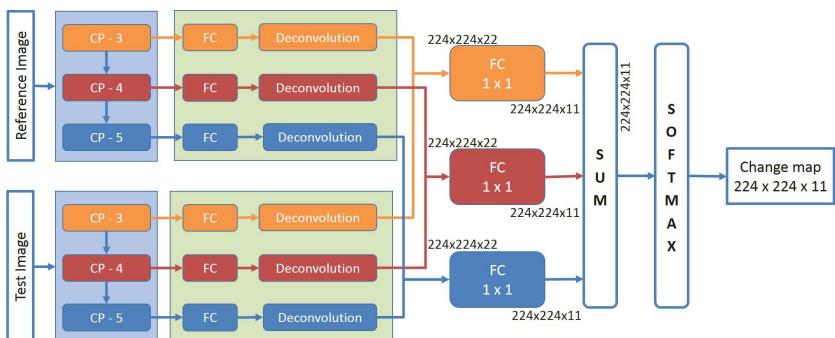


Fig. 2. Architecture for the proposed visual change detection. CP represents ResNet residual block and FC represents fully convolutional layer with kernel size of 1×1

The detailed architecture diagram for visual change detection is shown in Figure 2. There are two weight tied convolutional neural networks $CNN1$ and $CNN2$ for extracting features from I_{test} and I_{ref} respectively. In change detection, model learns representation from image pair; and then it tries to find relationship between them. However, task and input type of both the sub-networks are the same. Therefore similar type of features are expected from both the images. This can be achieved with a siamese network [22]. Siamese network is a weight tied network having same number of parameters and weight values. In addition, this approach optimizes the memory and training time without compromising on the performance. One key difference between a standard siamese and our network is that the weights are not tied in deconvolution layer. This resulted in significantly better performance of up to 5%. The reason could be due to the nature of siamese network where the convolution stage and the deconvolution stage have the weights tied that will force the network to work in constrained space. In

our network, the convolution stage will be weight tied but during deconvolution stage they work independently.

Since our training data are natural images and the number of training images are limited, transfer learning performs better than creating a new model from scratch. A Residual network (ResNet)-50 is used as pre-trained model [5]. ResNet uses residual blocks so that it can handle very deep architectures. ResNet block mainly consists of convolution layer, batch normalization and a rectified linear unit as shown in Figure 3 that contributes to feature extraction. The features from $CNN1$ can be represented as $f_1 = g(I_{test})$ and $CNN2$ as $f_2 = g(I_{ref})$. The output of three layers $f_{l^1}, f_{l^2}, f_{l^3}$ are tapped from the feature block in order to capture changes at different scales. l^1 is the CNN layer before fully connected layer and l^2 and l^3 are the residual blocks before l^1 .

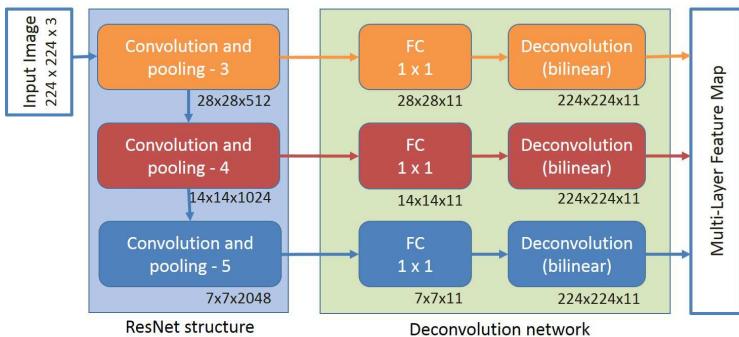


Fig. 3. Architecture for feature extraction.

Generally ResNet learns representation, and generates score for object classification. In this process, it uses max pooling layers and fully connected layers. However, it causes loss of spatial information. In semantic change detection, prediction should happen at input image spatial dimension in order to localize the changed area. Therefore, a mechanism is required to map learned representation onto input image dimension. Similar issue is already tackled in semantic segmentation. We adapted a similar approach to restore back the learned representation onto test image. There are different semantic segmentation approaches in literature like FCN [23], U-Net [27], PSPNet [28], SegNet [29], etc. Semantic segmentation uses both global and local information for encoding both semantics and location. As per [23], global information resolves semantic and local information resolves location.

A deconvolution layer is used to upsample output to image spatial dimension. It maps features f in high dimensional space to change map of $I_{w \times h \times N}$. Upsampling is done with bilinear interpolation filter. Bilinear filter interpolation predicts values from nearest four inputs. The filter parameters for upsampling is learned in the network itself. In order to incorporate both coarse and finer details, the convolution layer output from previous layer is also upsampled to input spatial dimension. Subsequently, upsampled output from both the parallel network are concatenated for comparison. Again, same

layers from both the networks are concatenated. The filter outputs f_{1l_1} and f_{2l_1} from layer l^1 of parallel network are concatenated. The same way, filter outputs f_{1l_2} and f_{2l_2} from layer l^2 ; filter outputs f_{1l_3} and f_{2l_3} from layer l^3 are concatenated. This ensures that we compare the representation at same degree and scale. Finally, all the concatenated outputs are added up together and shared to a *softmax* classifier to classify to one of the N classes. A convolutional layer with kernel size of 1×1 is used before softmax layer for reducing dimensionality to N classes. The changed area will be highlighted with class label of structural change. The network details are as follows: Fully convolution and up-sampling are performed on ResNet 7×7 , 14×14 , 28×28 convolution layers; and changed the dimension to $224 \times 224 \times 11$, where 224×224 is the input image dimension and 11 is the number of classes. Then, concatenation of subsequent layers and fully convolutional network resulted in three $224 \times 224 \times 11$ dimension. The three outputs are summed together (tensor addition) and given to a *softmax* classifier. It predicts a class for each pixels and generates a prediction of dimension $224 \times 224 \times 11$.



Fig. 4. Qualitative performance of change detection versus other approaches. Our approach gives change area as well as class label of structural change in the scene. Different classes are overlaid with different colors. Here green color represent vehicle class, purple for sign board, navy blue for rubbish bin and orange color for construction maintenance. Our approach can detect and localize changes in the scene at semantic level. Multiple colors in ChangeNet and ground truth images indicate separate class labels.

4 Experiments

For our experiments, three different datasets are used. The VL-CMU-CD dataset [4] is one of the most complex datasets available for change detection. It has 152 different scene changes. Each category has 2 to 41 pairs of test and reference images. Image pairs are taken at different view angle, seasonal and lighting condition as shown in figure 1. Out of 152 categories, we have chosen 103 categories, which are having more than 5 image pairs. This will ensure that we have enough samples from each category for training and testing. This results in a total of 1187 image pairs over 103 categories. The other two datasets - TSUNAMI and GSV - were developed by Sakurata and Okatani [6] and contains 100 pairs of images each. The definition of change in these two datasets are different compared with VL-CMU-CD. All the changes including variations in the background are considered as change. Each of the three datasets is divided it into train, validation and test in the ratio of 7:1.5:1.5. We evaluate our network on the test data by computing standard performance evaluation metrics such as precision, recall, f -score, ROC curve, Area under ROC (AUC) and Intersection over Union (IoU) measure [6, 4]. In addition to this, a five fold cross-validation is conducted on the VL-CMU-CD dataset to assess the network performance. The three datasets and related methods in literature focus on binary classification, that is the final output is to detect change or no-change. We refer to this scenario as *binary* for the rest of the paper. We are also interested in labeling the object after the change is detected. We call this scenario multi-class. Currently, the system is built for *multi-class* classification of 10 commonly appearing objects in VL-CMU-CD dataset: barrier, bin, construction, person/bicycle, rubbish bin, sign board, traffic cone, and vehicle. An Ubuntu based workstation with the following

Table 1. Analysis of ChangeNet results at class level on VL-CMU-CD data set. Miscellaneous class has been excluded from the table as all the values were 0.

Classification	Metric	Barrier	Bin	Cons- truction	Other objects	Person/ Bicycle	Rubbish bin	Sign board	Traffic -cone	Vehicle
Pixel based	Precision	0.55	0.80	0.88	0.92	0.83	0.87	0.77	0.53	0.91
	Recall	0.63	0.71	0.80	0.84	0.74	0.82	0.61	0.42	0.83
	f -score	0.59	0.76	0.84	0.88	0.79	0.84	0.68	0.47	0.87
Object based	Precision	0.50	0.97	0.86	1.00	1.00	0.96	1.00	1.00	1.00
	Recall	0.87	1.00	1.00	0.87	1.00	1.00	0.75	0.50	0.91
	f -score	0.63	0.98	0.92	0.93	1.00	0.97	0.85	0.66	0.95

configuration is used for training and testing purpose: Intel core i7 @3.4Gx8, 32GB RAM and NVIDIA GM204GL [Quadro M4000] GPU card. Tensorflow, a deep learning library with python support is used for implementing deep learning network.

5 Results and Discussions

The ChangeNet architecture was specifically designed keeping VL-CMU-CD dataset in mind due to its complexity. In order to validate the architecture, a 5 fold cross validation

was conducted. The results are as shown in Table 2 and a healthy average f -score of 86.9% was obtained for binary classification and 73.87% for multi-class classification. Figure 5 shows the boxplot of cross validation performance for binary classification scenario. As it can be seen, the variation across different folds in marginal reflecting good generalization of the proposed architecture.

Table 2. Average results of 5-fold cross validation for binary and multi-class scenarios

	Accuracy	Precision	Recall	f -score
Binary	98.3	87.98	85.85	86.90
Multi-class	82.58	77.14	71.43	73.87

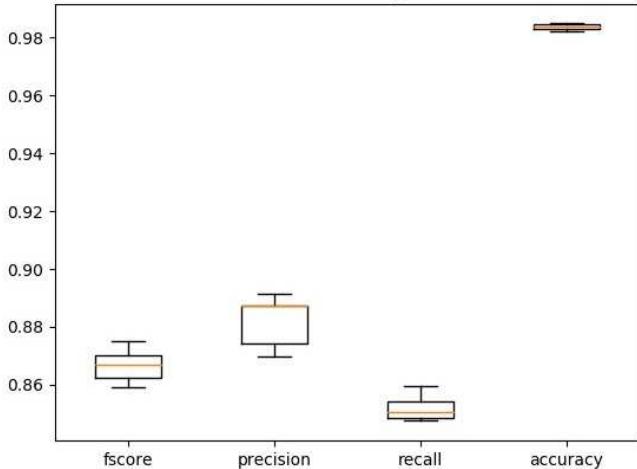


Fig. 5. Boxplot of ChangeNet performance in 5-fold cross validation for binary classification

To further confirm its performance, multi-scale super pixel [17] and CDnet [4] were compared to the proposed architecture. In order to make a fair comparison, the results of binary classification (change or no change) of all the methods are compared by converting our class based output into binary form. The predicted change map of baseline approaches and our method are shown in Figure 4. Each sample exhibits different lighting and seasonal condition. The first column is the test image, which is compared against reference image in the second column. The third and the fourth columns are the change detection results of multi-scale super pixel method and CDnet. The changed area is highlighted with red color. CDnet result images are taken from [4] for comparison purpose. The results of ChangeNet is shown in the fifth column. The changed area is highlighted with corresponding class label. The ground truth is given in the last column. It should be noted that different colored labels for ChangeNet and Ground truth

indicate multi-class classification that is unique to our work. ChangeNet achieves this in a single shot and a single network is able to detect change and label them. As shown in Figure 4, ChangeNet performs better than other approaches both in terms of the output as well as in terms of change class labelling. It gives a better performance in terms of accuracy and precision. Compared to other approaches, it gives additional information like what is the structural changes in the scene. In other words, our approach is able to tell *where* is the change in the scene as well as *what* the change is. ChangeNet performs well even though the background between image pair is different due to seasonal alterations and lighting conditions. For example, image pair in row 2 are taken at different lighting condition and ChangeNet was able to detect the changed area. An example of multiple changes in the same scene is depicted in row 4 where vehicle and a sign board are depicted as change. ChangeNet is able to identify both of them accurately. However, small objects such as sign board are mis-classified as background. One of the reasons is that the sign board object is less dominant in test image. This is due to the features that we tap out at different levels whose receptive fields cover certain minimum area. Results in row 5 shows performance of ChangeNet when images are captured at different seasonal conditions. The reference image shows the presence of snow in the background. The same case applies to row 6 as well. Model performed well in this case, and it could detect and locate the rubbish bin. Since we approached the change detection problem at semantic level, we could mitigate irrelevant background information and reduce false alarms, if any.

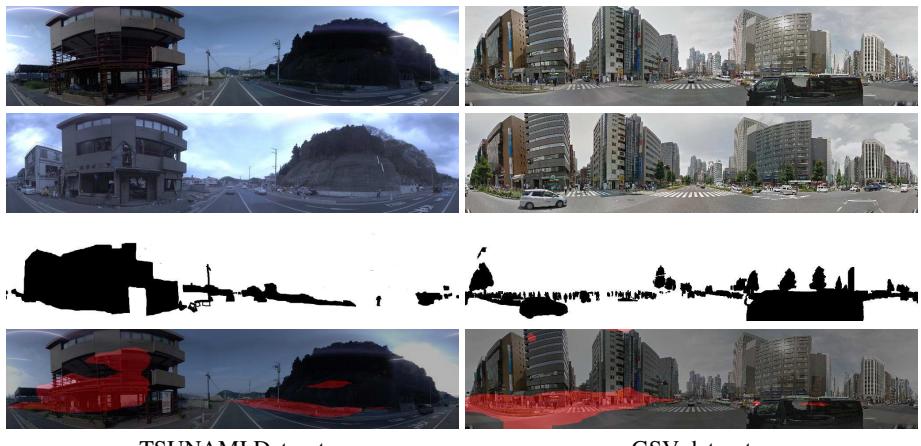


Fig. 6. Qualitative performance of ChangeNet on Tsunami and GSV dataset. From top to bottom: Reference image, Test image, Ground truth mask and ChangeNet

Quantitative performance of our method is evaluated in two aspects. First aspect is how accurately it localized the change. Once it localized the change, what is the pixel labeling accuracy. Mainly, Intersection over Union (IoU) and pixel accuracy metrics are

used for evaluating the performance. We considered 11 classes including background for this performance measurement. Model is evaluated with 177 image pairs and the results are generated. The performance metric for ChangeNet is given in Table 3. We achieved 98.3% pixel level accuracy and 82.58% mean pixel accuracy. In other words, 98.3% pixels are classified as change correctly. In that, 82.58% pixels are classified correctly per class basis. Also, we achieved 77.35% IoU. It compares the ground truth and predicted changed area on a per class basis. IoU is changed to 96.96% once we assigned the weights to class IoU based on their appearance frequency. Table 1 shows the results of ChangeNet in identification of class based change. Other than barrier and traffic cone, all other classes resulted in a f -score of over 0.8 for object level change detection. At pixel level, small objects including traffic cone, barrier and sign board resulted in lower f -scores. Table 4 shows quantitative comparison of ChangeNet with CDnet [4] and Super-pixel [17] methods for two different false positive rates of 0.1 and 0.01. As it can be seen, ChangeNet outperforms both the methods with impressive f -scores. Figure 7 shows the Receiver Operator Characteristic (ROC) curve for binary classification of ChangeNet, CDnet [4] and super-pixel [17] based methods. All the classes except background is considered as logical one. ChangeNet resulted in steep ROC curve with maximum true positive rate and minimum false positive rate. The area under ROC curve, i.e., AUC is 89.2%.

Table 3. Performance metrics for ChangeNet

Pixel Accuracy	Mean Pixel Accuracy	Mean IoU	Frequency weighted IoU
98.3	82.58	77.35	96.96

Table 4. The quantitative comparison of our method with other approaches for FPR=0.1 and FPR=0.01. Pr -Precision, Re -Recall and $F1$ - f -score

	FPR=0.1			FPR=0.01		
	Pr	Re	$F1$	Pr	Re	$F1$
Super-pixel [17]	0.17	0.35	0.23	0.23	0.12	0.15
CDnet [4]	0.40	0.85	0.55	0.79	0.46	0.58
ChangeNet	0.79	0.80	0.79	0.80	0.79	0.79

Detailed results of ChangeNet on the three datasets tested is presented in Table 5. For TSUNAMI and GSV datasets, the performance measures are calculated in the region of interest as well as for the whole image (within parenthesis). As it can be seen, the results on VL-CMU-CD dataset are very high with nearly good performance on TSUNAMI dataset. There is a drop in GSV performance. The drop in performance on GSV dataset is attributed to the way the ground truth is created in these datasets. ChangeNet focuses on structural changes but GSV ground truth represents cars on the road as change. Hence, the overall performance seems to dip. The results of ChangeNet

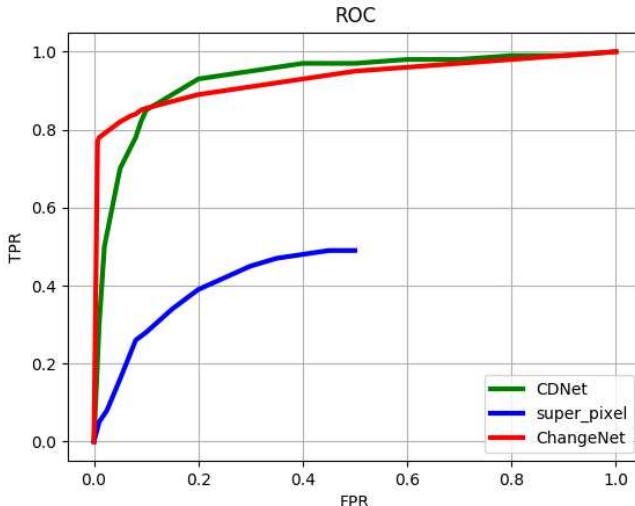


Fig. 7. ROC and FPR-TPR curve for binary class

on TSUNAMI and GSV dataset are shown in Figure 6. It can be clearly seen that we have been able to detect dominant objects such as houses and trees as changes but movement of cars and small signboards are not detected.

Table 5. Performance metrics of ChangeNet for binary classification (change or no-change) on different datasets. The values for TSUNAMI and GSV datasets are in the following format: calculated in the region of interest (calculated on whole image)

Dataset → Metric ↓	CMU-CD	Tsunami	GSV
Precision	0.88	0.73 (0.82)	0.51 (0.67)
Recall	0.80	0.74 (0.82)	0.45 (0.66)
<i>f</i> -score	0.84	0.74 (0.82)	0.48 (0.67)
Accuracy	0.97	0.85 (0.85)	0.77 (0.77)
IoU	0.64	0.55 (0.69)	0.27 (0.5)
Area overlap	0.89	0.71 (0.71)	0.44 (0.44)
AuC	0.89	0.82 (0.82)	0.66 (0.66)

Finally, the performance of the three different methods on three different datasets are presented in Table 6. It should be noted that the definition of change detection is evolving. The new datasets and methods that can detect change at higher levels of inference is becoming possible. This paper is one of the early works in the direction and hence there are very few methods in literature that can be compared, which is presented in Table 6. For VL-CMU-CD dataset, ChangeNet results in the best performance. For TSUNAMI dataset, CDnet gives the best result but ChangeNet is not too far behind.

However, for GSV dataset, CDnet outperforms other methods. The lack of robustness in detecting small changes is found to be the main drawback of ChangeNet. In the future, we plan to extend the network for detecting small object change detection in the presence of occlusion.

Table 6. Comparison of f -score values of ChangeNet with other methods on different datasets. For ChangeNet, two separate f -scores are calculated based on false positive rates

Method→ Dataset↓	Super-pixel	CDnet	ChangeNet (0.1)	ChangeNet (0.01)
TSUNAMI	0.38	0.77	0.73	0.47
GSV	0.26	0.61	0.45	0.20
VL-CMU-CD	0.15	0.58	0.80	0.83

ChangeNet does not depend on the objects trained in the network for change detection. Change is first detected and then object label (semantic information) is inferred using a single network. Figure 8 shows the reference image, test image and change detected image (from left to right). The bin in both the picture is in the object classes but it is not detected as change as it is present in both the images. However, bicycle is highlighted as the system detects change in that region and labels the class correctly.



Fig. 8. ChangeNet performance demonstration: reference (left), test (middle) and change detected (right). The result in the figure demonstrates that ChangeNet is focused on change rather than the object category. In spite of both bin and bicycle being present in our category labels, only bicycle is highlighted as a change, which is along expected lines.

6 Conclusion

A deep learning architecture called ChangeNet is proposed for detecting structural changes between an drone captured image pair. The new architecture is comprised of two parallel weight tied networks that act as image feature extractors for detecting change. Features at different layers are merged and a fully convolutional network is used to detect change. ChangeNet is experimented and evaluated with VL-CMU-CD dataset, which is very challenging. ChangeNet detects and localizes the changes of the same scene captured at different lighting, view angle and seasonal condition. Further, for the first time, a network that can detect and report change in semantics (scene labels) is demonstrated. 98.3% pixel accuracy, 77.35% class based IoU and 88.9% AUC was achieved.

References

1. Sahi K., M., Wheelock, C.: Drones for commercial applications. Tractica Research Report (2017)
2. St-Charles, P.L., Bilodeau, G.A., Bergevin, R.: Subsense: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing* **24**(1) (2015) 359–373
3. Hussain, M., Chen, D., Cheng, A., Wei, H., Stanley, D.: Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing* **80** (2013) 91–106
4. Alcantarilla, P.F., Stent, S., Ros, G., Arroyo, R., Gherardi, R.: Street-view change detection with deconvolutional networks. In: *Robotics: Science and Systems*. (2016)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2016) 770–778
6. Sakurada, K., Okatani, T.: Change detection from a street image pair using cnn features and superpixel segmentation. In: *BMVC*. (2015) 61–1
7. Rensink, R.A.: Change detection. *Annual review of psychology* **53**(1) (2002) 245–277
8. Goyette, N., Jodoin, P.M., Porikli, F., Konrad, J., Ishwar, P.: Chagedetection.net: A new change detection benchmark dataset. In: *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. (June 2012) 1–8
9. Wang, Y., Jodoin, P.M., Porikli, F., Konrad, J., Benezeth, Y., Ishwar, P.: Cdnet 2014: An expanded change detection benchmark dataset. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. (June 2014) 393–400
10. Bilodeau, G.A., Jodoin, J.P., Saunier, N.: Change detection in feature space using local binary similarity patterns. In: *Computer and Robot Vision (CRV), 2013 International Conference on*, IEEE (2013) 106–112
11. Sedky, M., Moniri, M., Chibelushi, C.C.: Spectral-360: A physics-based technique for change detection. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. (June 2014)
12. De Gregorio, M., Giordano, M.: Change detection with weightless neural networks. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. (June 2014)
13. Wang, R., Bunyak, F., Seetharaman, G., Palaniappan, K.: Static and moving object detection using flux tensor with split gaussian models. *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2014) 420–424
14. Bianco, S., Ciocca, G., Schettini, R.: How far can you get by combining change detection algorithms? *CoRR* **abs/1505.02921** (2015)
15. Gressin, A., Vincent, N., Mallet, C., Paparoditis, N. In: *Semantic Approach in Image Change Detection*. Springer International Publishing, Cham (2013) 450–459
16. Kataoka, H., Shirakabe, S., Miyashita, Y., Nakamura, A., Iwata, K., Satoh, Y.: Semantic change detection with hypermaps. *arXiv preprint arXiv:1604.07513* (2016)
17. Gubbi, J., Ramaswamy, A., NK, S., Varghese, A., P, B.: Visual change detection using multiscale super pixel. In: *Digital Image computing: Techniques and applications*. (2017)
18. Bansal, A., Russell, B.C., Gupta, A.: Marr revisited: 2d-3d alignment via surface normal prediction. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. (2016) 5965–5974
19. Bansal, A., Chen, X., Russell, B.C., Gupta, A., Ramanan, D.: Pixelnet: Representation of the pixels, by the pixels, and for the pixels. *CoRR* **abs/1702.06506** (2017)

20. Du, W., Fang, M., Shen, M.: Siamese convolutional neural networks for authorship verification
21. Mueller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In: AAAI. (2016) 2786–2792
22. Koch, G.: Siamese neural networks for one-shot image recognition. Master’s thesis, University of Toronto, Canada (2015)
23. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3431–3440
24. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 1–9
25. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
27. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer (2015) 234–241
28. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. arXiv preprint arXiv:1612.01105 (2016)
29. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561 (2015)