CDNet++: Improved Change Detection with Deep Neural Network Feature Correlation

K. Ram Prabhakar
Department of CDS
Indian Institute of Science
Bangalore, INDIA

Akshaya Ramaswamy
Innovations labs
Tata Consultancy Services
Chennai, INDIA

Suvaansh Bhambri
Department of CDS
Indian Institute of Science
Bangalore, INDIA

Jayavardhana Gubbi
Innovations labs
Tata Consultancy Services
Bangalore, INDIA

R. Venkatesh Babu
Department of CDS
Indian Institute of Science
Bangalore, INDIA

Balamuralidhar Purushothaman
Innovations labs
Tata Consultancy Services
Bangalore, INDIA

***Abstract*—In this paper, we present a deep convolutional neural network (CNN) architecture for segmenting semantic changes between two images. The main objective is to segment changes at the semantic level than detecting background changes, which are irrelevant to the application. The difficulties include seasonal changes, lighting differences, artifacts due to alignment and occlusion. The existing approaches fail to address all the problems together; thus, none of them achieve state-of-the-art performance in three publicly available change detection datasets: VL-CMU-CD [1], TSUNAMI [2] and GSV [2]. Our proposed approach is a simple yet effective method that can handle even adverse challenges. In our approach, we leverage the correlation between high-level abstract CNN features to segment the changes. Compared with several traditional and other deep learning-based change detection methods, our proposed method achieves state-of-the-art performance in all three datasets.**

***Index Terms*—Change Detection, Image Segmentation, Deep Learning**

I. Introduction

Change detection is one of the major pre-processing techniques used for various computer vision tasks. Generally, change regions (or foreground objects) are the regions of interest in image processing tasks. Objects like vehicles, pedestrians, etc., are of utmost importance and need to be localized and segmented in many tasks like satellite imaging, CCTV surveillance, etc. This technique is beneficial in such cases and makes the task of identification and localization more straightforward. The problem we target is the semantic segmentation of change in the scene. We need to detect changes at the semantic level rather than detecting all the changes in the background. The challenges are complex, considering the variations caused by environmental conditions that are unchanged events. Significant challenges involved are brightness difference, occlusion, seasonal changes, imperfect alignment, and occlusion.

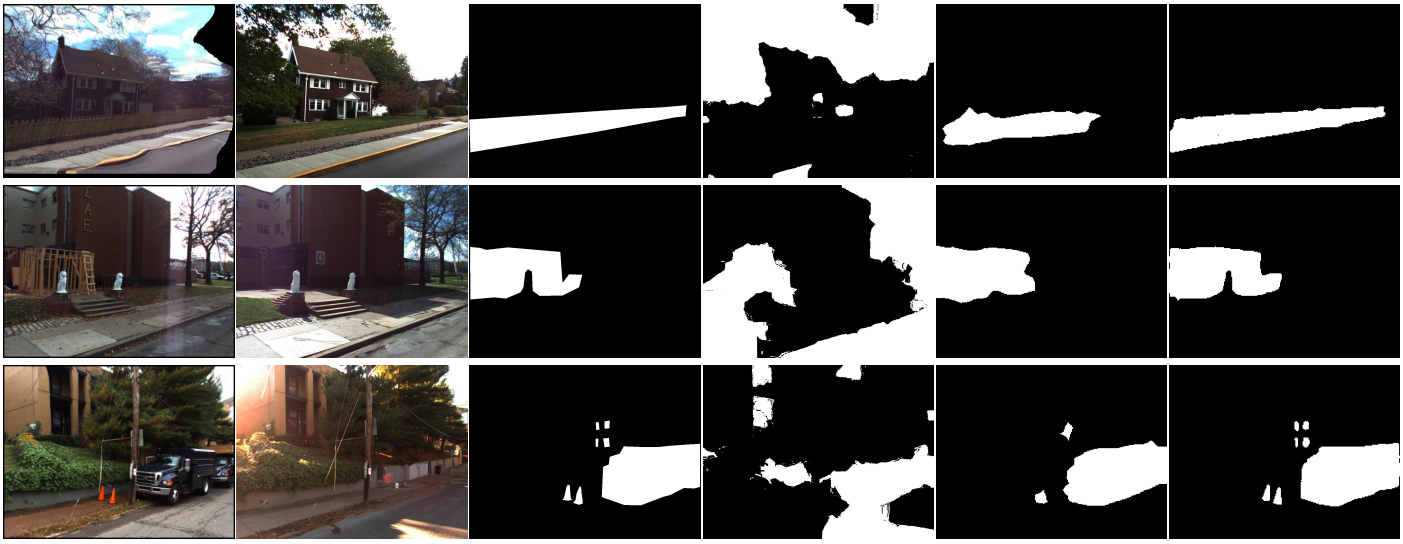
The traditional approach to this problem is to detect change regions (or foreground objects) from the difference between the test frame and reference frame, often called *background image*, or *background model* ([5]–[7]). Several methods in the literature follow a two-stage process. The first stage involves developing a model for static or background pixels. In the second stage, the developed background model is used to detect pixels (*foreground* pixels) that deviate from the estimate. The success of such methods relies on the accuracy of the estimated background model. They need to be updated for new and challenging scenes. These approaches require to observe many reference frames to build a robust background model. However, in our problem, the changes have to be segmented provided a single reference frame.

TABLE I: Quantitative comparison between CDNet++ against state-of-the-art methods for binary change segmentation in VL-CMU-CD, GSV, and TSUNAMI dataset with f-score. The numbers in the bracket indicate the improvement gained by the proposed method over the individual approach.

Datasets (→) Methods (↓)	VL-CMU-CD	GSV	TSUNAMI
CDNet [1]	0.58 (+0.36)	0.61 (+0.07)	0.77 (+0.09)
Super-pixel [3]	0.15 (+0.79)	0.26 (+0.42)	0.38 (+0.48)
ChangeNet [4]	0.80 (+0.14)	0.45 (+0.23)	0.73 (+0.13)
CDNet++	**0.94**	**0.68**	**0.86**

Recently, Convolutional Neural Networks (CNN) are used to learn problem-dependent features that outperform traditional methods in change detection. Lim *et al.* [8] proposed a method to intelligently fuse multiscale CNN features with feature pooling, to learn class-specific foreground extractor. Alcantarilla *et al.* [1] proposed a new CNN based change detection method called CDnet. For training CDnet, the authors curated a new urban change detection dataset called as VL-CMU-CD. The VL-CMU-CD dataset consists of 1362 registered image pairs with 11 object classes, captured at different time instances over a year. It contains challenging changes like structural, construction, and natural seasonal changes. Sakurada *et al.* [2] use a combination of superpixel segmentation and pre-trained deep neural network weights to detect changes. Also, they have created a new dataset called

978-1-7281-6926-2/20/$31.00 ©2020 IEEE



(a) Test (b) Reference (c) Ground truth (d) SuperPixel [3] (e) ChangeNet [4] (f) CDNet++

Fig. 1: Qualitative comparison for binary segmentation with SuperPixel ([3]), ChangeNet ([4]) and proposed method for images from VL-CMU-CD dataset [1]. Our proposed method, *CDNet++*, can accurately segment the semantic changes between test and reference image. The test and reference images: have seasonal differences (first row), are captured during a different time of the day (second and third row) and alignment and/or warping artifacts (notice the distortions in the walking path of the first-row example). Irrespective of these challenges, our proposed method detects the changes with better boundary precision. Notice that, compared with the existing methods, Our method can precisely segment even the small changes (in the third-row example) with accurate boundaries.

TSUNAMI and Google Street View (GSV) for benchmarking change detection algorithms. Both TSUNAMI and GSV contain 100 panoramic image pairs each, with days or months of time gap between the pairs. These datasets cover changes on surfaces of objects (like changes in billboard signs), structural changes (such as changes in a building structure). Similarly, Gubbi *et al.* [3] proposed a multiscale superpixel method for change detection in drone imaging.

More recently, Varghese *et al.* [4] proposed ChangeNet, a deep neural network-based approach that uses pre-trained CNN features to detect changes. Currently, ChangeNet is state-of-the-art in the VL-CMU-CD dataset. However, they have shown to perform lower than [1] in GSV and TSUNAMI dataset. As a whole, there is no single method that outperforms in all the three datasets. Such a technique would be able to detect both pixel level and semantic level changes irrespective of the challenges posed in real-life conditions.

The challenges present in the VL-CMU-CD dataset is shown with a few examples in Figure 1. As can be seen from the examples, the dataset has a huge variety of challenges. The existing SuperPixel [2] approach segments the seasonal changes also as part of changes (For the first-row example, the trees are also segmented). Comparatively, ChangeNet [4] model performs better. However, the boundaries predicted by ChangeNet is inaccurate. Additionally, it fails to segment small change regions in the third-row example.

Motivated by these issues, we propose a CNN based change segmentation model, CDNet++, that has better localization as well as accurate boundary prediction capability. The proposed CDNet++ architecture extracts semantic features from multiple depths of a CNN feature extractor. Then, we use the correlation between extracted features to segment the change regions. As highlighted in Figure 1, our proposed method can accurately segment changes irrespective of the size of the change regions.

In summary, our contributions are:

- We propose a CNN-based change detection method that is robust to various environmental, alignment and warping challenges,
- Through experimental evaluation, we show the efficacy of the proposed method in VL-CMU-CD, GSV, and TSUNAMI datasets.
- We show extensive ablation experiments and analysis on various choices of model architecture and parameters.

The rest of the paper is organized as follows. The proposed method is described in Section II. The implementation details, evaluation protocols, and results are discussed in Section III. Finally, we conclude the paper in Section IV.

II. Proposed approach

Overview: Given a test (I_t) and a reference image (I_r), the objective is to segment change regions between them and label each pixel into one of the following ten classes: barrier, bin, construction, person/bicycle, rubbish bin, signboard, traffic cone, vehicles, other objects, and background. I_t and I_r need not be captured sequentially; the images in VL-CMU-CD, TSUNAMI, and GSV are captured at different seasons

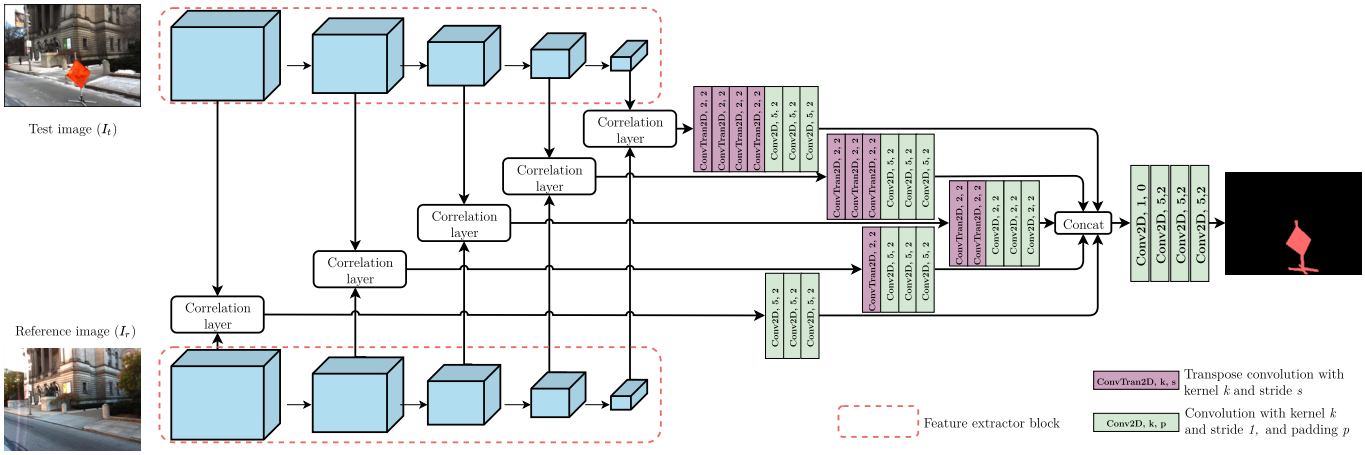


Fig. 2: Overview of our proposed CDNet++ model. The test image (I_t) and reference image (I_r) are passed as input to feature extractor to obtain high-level semantic features. Then, the correlation between individual features at five different levels estimates the likelihood of finding a similar feature in a fixed neighborhood. Finally, a block of convolutional layers convert the correlation map into ten class segmentation output.

altogether. Additionally, though I_t and I_r are pre-aligned using global homography and/or optical flow, exact pixel-wise matching is not guaranteed. For example, non-rigid alignment techniques can introduce warping artifacts due to extreme seasonal scene changes and occlusion. Hence, the challenges involved in segmenting changes are seasonal differences (with and without snow, tree leaves grown during the autumn season, etc.), lightning conditions (images could have been captured during a different time of the day), alignment, warping, and occlusion.

In our approach, we identify seasonal and lightning changes by comparing semantic features from deeper convolution layers. However, pixel-wise comparison at high-level semantic features can still fail due to alignment/warping and occlusion problems. Hence, to handle such issues, we propose to compute a patch-wise correlation between feature maps. The resultant correlation map is further processed using a set of convolution layers to generate the final ten channel segmentation output.

Feature extractor: The success of change segmentation relies on the quality of extracted image features. In our method, we use VGG19 [9] model as base network architecture. We initialize the model with ImageNet pre-trained weights and finetune the weights for our task. We extract feature maps at five different locations of the network. The VGG19 model consists of five major convolutional blocks, followed by three fully connected layers. As we require only convolutional features, we discard FC layers of the VGG19 model. Also, the VGG19 model has max-pool layers at the end of each convolutional block. We extract pre-max pool features for both input images. Hence, we obtain five feature maps ($f_{t,r}^{1,\cdots,5}$), each with half-resolution as the previous feature map:

$$F_j = \{f_j^i\},\ \forall\, i = (1,2,3,4,5)\ and\ j = (t,r) \tag{1}$$

Feature Correlation: The extracted features contain semantic information about both the test and the reference frame. One simplistic way to segment out the changes is to subtract both features. However, that holds only for the case where both test and reference frames are registered accurately. To solve this problem, we make use of the correlation layer to compute pixel similarity. As shown in Figure 2, we compute correlation map between two feature maps of I_t and I_r with same dimension. Let f_t^k and f_r^k denote the k^{th} block feature map of I_t and I_r. Let the size of the feature maps be $h \times w \times c$, where (h, w, c) denote height, width and number of channels. The correlation of the two patches at location l_1 of f_t^k and l_2 of f_r^k is defined as,

$$correlation(l_1, l_2) = \sum_{o \in [-s,s] \times [-s,s]} \langle f_t^k(l_1 + o), f_r^k(l_2 + o) \rangle \tag{2}$$

The correlation is computed between two patches of size $(2s+1, 2s+1)$. Ideally, one can compute correlation between every l_1 location with all possible l_2 locations i.e. $h * w$. However, that would require huge computations. Thus, we restrict our correlation comparison to a fixed search area of $T \times T$ centered around l_2. Hence, for every l_1 location at f_t^k, we compare a patch of size $(2s+1, 2s+1)$ of f_t^k with T^2 locations centered at same l_1 location of f_r^k, resulting in a correlation map $\mathcal{C}^k$ of size $(h \times w \times T^2)$.

We compute five correlation maps, one for each of the five feature levels. These correlation maps are in different resolutions due to max-pooling operations in the feature extractor. While, f_t^1 (and f_r^1) has the same dimension as the inputs, f_t^5 (and f_r^5) dimension is downsampled by a factor of 16, because of four max-pooling operations performed due feature extraction. Hence, we upsample correlation maps to the same dimension as input shape by applying transpose convolutions. We perform four transpose convolution operations on $\mathcal{C}^5$ to bring it to the same shape as inputs. Similarly, for remaining correlation maps, we apply transpose convolution till their

dimension matches input. Three convolution layers further process the output feature maps of transpose convolution layers (see Figure 2).

Segmention prediction: The five output feature maps from the previous step are aggregated by concatenating them in the feature space. The concatenated features are further processed by a block of four convolution layers to predict the final ten channel segmentation map. The model is trained in an end-to-end setup with cross-entropy loss between the predicted segmentation map and ground truth segmentation map.

III. Experiments

A. Network Implementation

The baseline VGG19 network consists of sixteen convolutional Layers and five Max-pool layers (with stride=2). Each convolutional layer consists of a convolution with kernel size = 3 with ReLU activation. For upscaling the $\{\mathcal{C}^5, \mathcal{C}^4, \mathcal{C}^3, \mathcal{C}^2\}$ correlation maps, we use $\{4, 3, 2, 1\}$ transpose convolution layers respectively. For each transpose convolution layer, we use kernel size = 2 and stride = 2. Further, the output of transpose convolution layers is passed to a set of three convolution layers with 64 filters and kernel size = 5. The resultant feature maps are concatenated and passed through four convolutional layers with $\{128, 64, 32, 10\}$ filters, and kernel size = 5. The network is trained in an end-to-end fashion with cross-entropy loss between predicted and ground truth segmentation map. We use Adam optimizer with learning rate of 0.0001 and β_1 = 0.9 and β_1 = 0.999 to optimize the loss function. Through ablation experiments, we found that setting s (in Equation 2) to 5 and T to 7 yields best result.

B. Datasets and protocols

We train and evaluate our method in three publicly available datasets: VL-CMU-CD ([1]), GSV, and TSUNAMI ([2]). VL-CMU-CD dataset consists of 1187 image pairs in total. We follow the dataset split used in ChangeNet ([4]) for our experiments as well. The dataset is split into a ratio of 70:15:15% for training, validation, and testing. Similarly, GSV and TSUNAMI datasets consist of 100 image pairs each, out of which 70 image pairs were used for training and 15 image pairs each for validation and testing. We perform a quantitative evaluation on all three datasets using five-fold cross-validation. For training the proposed model, we computed cross-entropy loss between ground truth and predicted output with class re-balancing weights. We implemented our model in Tensorflow ([10]) installed on a workstation with Intel Xeon at 3.50 GHz CPU with 32GB RAM and an NVIDIA Titan X GPU card.

CDNet [1] and SuperPixel [3] methods are developed to segment change or no-change regions i.e. binary segmentation. Hence, to compare these two methods, we convert our ten class segmentation map to binary map by mapping all non-background classes to 1 and background class to 0 (see Figure 1). ChangeNet [4] model is trained to predict ten class segmentation map similar to our approach. Hence, we perform both binary and multiclass segmentation comparison against ChangeNet model.

TABLE II: Quantitative analysis of CDNet++ results at class level on VL-CMU-CD data set.

Class($\rightarrow$) Metric($\downarrow$)	Barrier	Bin	Constr -uction	Other objects	Person/ Bicycle	Rubbish bin	Sign board	Traffic cone	Vehicle	Overall
Precision	0.63	0.77	0.89	0.91	0.60	0.93	0.69	0.76	0.96	0.88
Recall	0.87	0.92	0.78	0.37	0.67	0.79	0.83	0.87	0.88	0.81
f-score	0.73	0.84	0.83	0.53	0.63	0.86	0.75	0.81	0.92	0.84

TABLE III: Average results of 5-fold cross validation for binary and multi-class categories in VL-CMU-CD dataset.

	Accuracy	Precision	Recall	f-score
Binary	0.991	0.94	0.94	0.94
Multi-class	0.953	0.88	0.81	0.84

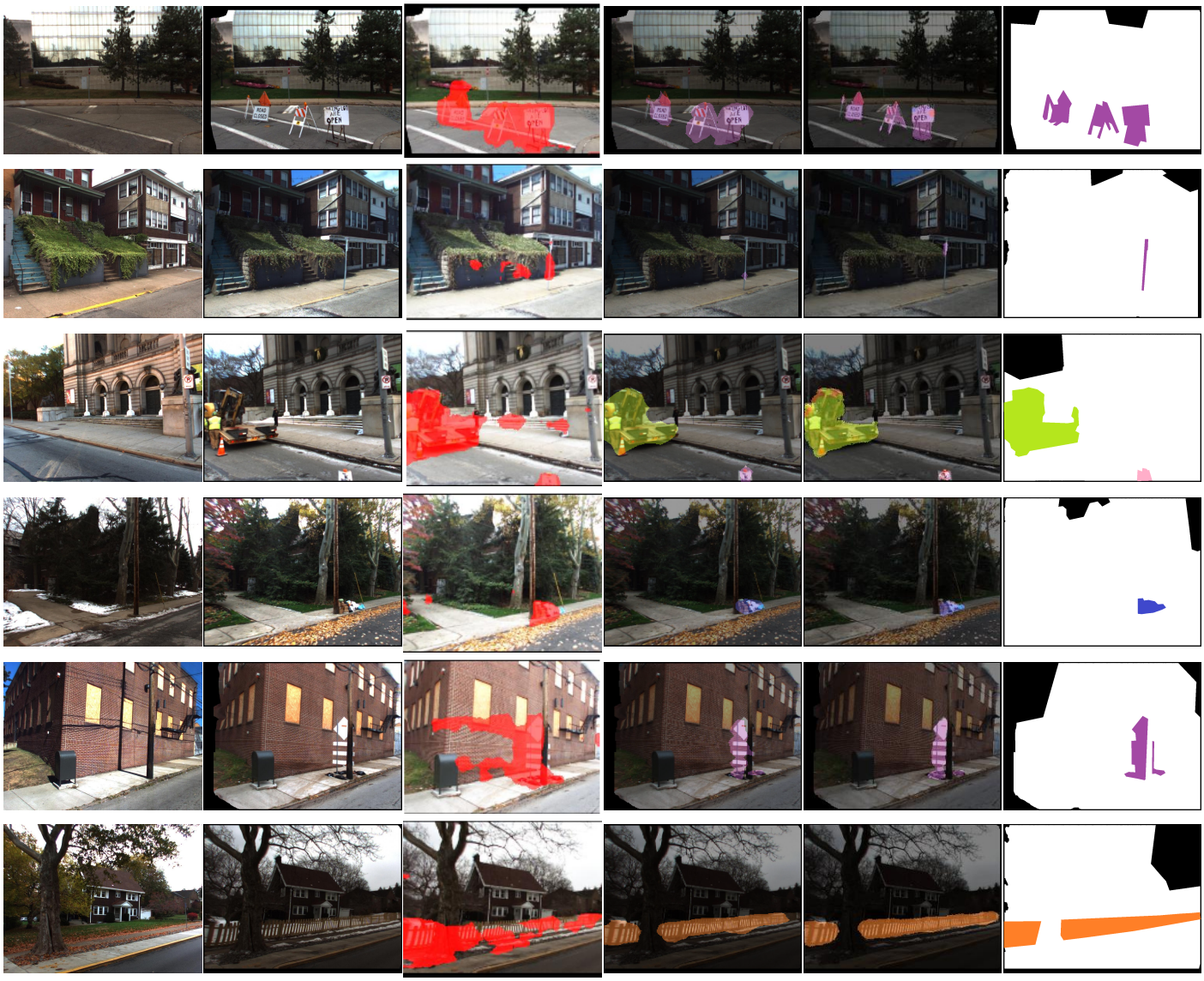
C. Results

1) Evaluation metrics: We evaluate the performance of the proposed method with state-of-the-art methods using standard F1 evaluation metrics. Also, we evaluated proposed method on three datasets in Table VIII using following standard metrics: Accuracy, Precision, Recall, F1 score, mean Intersection over Union (mIoU), Matthew's correlation coefficient (MCC), Sensitivity, Percentage of Wrong Classifications (PWC), Specificity, False Positive Rate (FPR) and False Negative Rate (FNR).

2) Quantitative comparison: We compare our proposed CDNet++ method with three state-of-the-art methods: SuperPixel ([3]), CDNet ([1]) and ChangeNet ([4]). The results are shown in Table I. Compared with ChangeNet in VL-CMU-CD dataset, our proposed method offers over 14% improvement in f-score. Our method outperforms all three comparison methods in all of the three datasets. In the GSV dataset, our method performs better than CDNet by 7% and better than SuperPixel method by 42%. In TSUNAMI dataset, our method similarly performs better than CDNet by 9% and better than SuperPixel method by 48%. The improvement in accuracy is attributed to the fact that the proposed model is robust enough for image misalignment.

In Table II, we present the class-specific metrics for all ten classes in VL-CMU-CD dataset. From the table, we observe that our method performs better for vehicles, rubbish bin, bin, construction, and traffic cone classes. While it underperforms for other objects and person/bicycle category. We perform five-fold cross-validation and report the results for both binary and multiclass segmentation in Table II. In overall, our method achieves 0.84 f-score, which is 0.11 more than ChangeNet (see Table IV).

In Table V, we present the precision, recall and f-score values for two FPR values 0.1 and 0.01 in all three datasets. Similarly in Table VII, we present the comparison between proposed approach and existing methods for same two FPR values in VL-CMU-CD dataset. In Figure 6, we compare ROC curves of proposed method against SuperPixel [3], CDNet [1] and ChangeNet [4]. CDNet++ achieves steep curve as compared to CDNet and SuperPixel methods. Also, CDNet++



(a) Reference image (b) Test image (c) CDNet [1] (d) ChangeNet [4] (e) CDNet++ (f) Ground truth

Fig. 3: Qualitative comparison for multi-class segmentation with CDnet, ChangeNet and proposed method for images from VL-CMU-CD dataset. As CDNet is trained to trained for binary segmentation, we color code change region to red for visualization purpose.

TABLE IV: Average results of 5-fold cross validation for multi-class categories in VL-CMU-CD dataset.

Methods (→) Metrics (↓)	ChangeNet [4]	CDNet++ (VGG16)	CDNet++ (VGG19)
Precision	0.77	0.87	**0.88**
Recall	0.71	**0.84**	0.81
f-score	0.73	**0.86**	0.84

TABLE V: Quantitative analysis of our approach at FPR=0.1 and 0.01 for three different datasets.

	FPR = 0.1			FPR=0.01		
	Precision	Recall	f-score	Precision	Recall	f-score
VL-CMU-CD	0.910	0.900	0.910	0.880	0.975	0.925
TSUNAMI	0.782	0.939	0.853	0.953	0.588	0.727
GSV	0.669	0.687	0.678	0.864	0.209	0.336

achieves AUC (area under the ROC curve) of 99.4%.

In Table VIII, we evaluate our proposed CDNet++ model on three different datasets using 11 standard metrics. Overall, CDNet++ performs well for the VL-CMU-CD dataset in all metrics compared to TSUNAMI and GSV datasets. The drop in performance in GSV and TSUNAMI datasets is because challenges faced in them are different and also due to less training data.

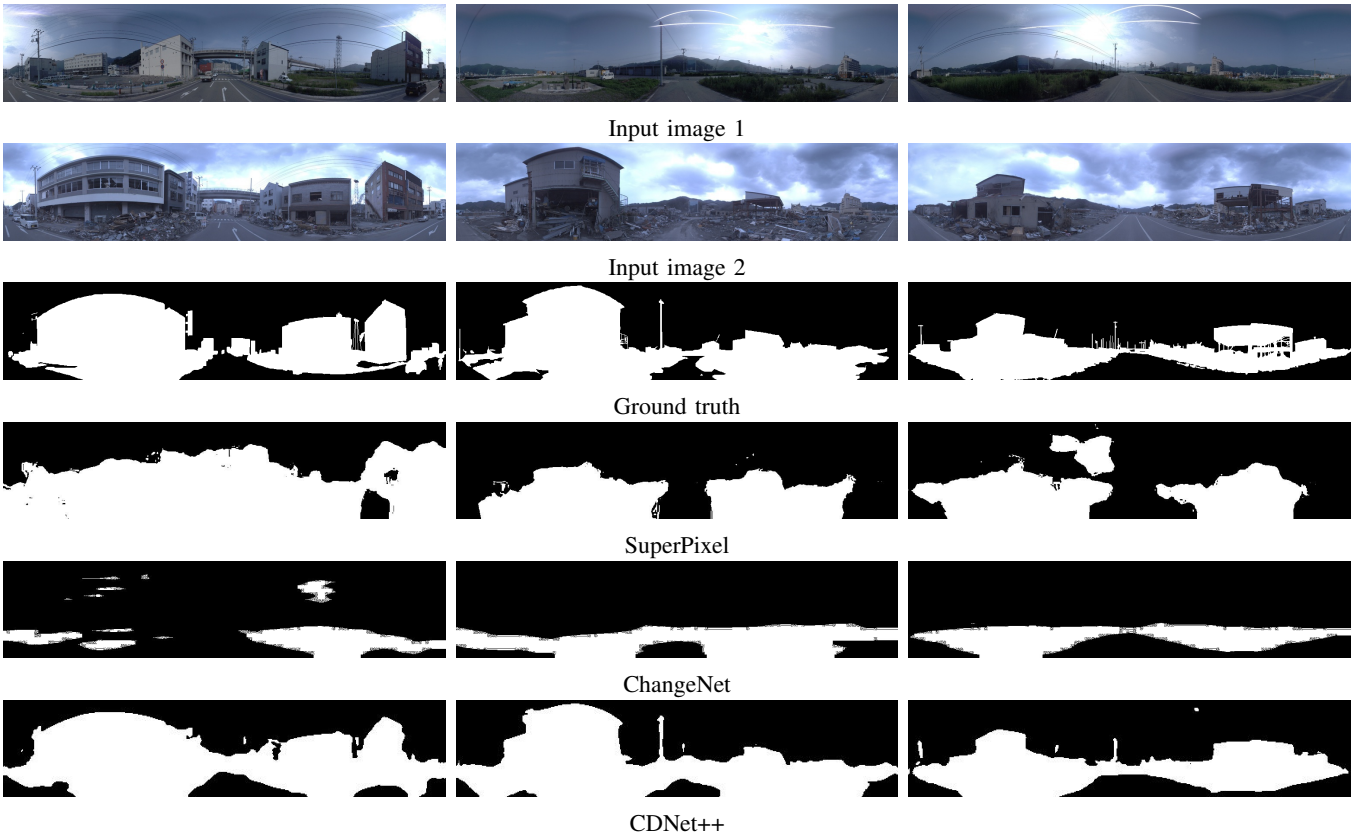


Fig. 4: Qualitative comparison with SuperPixel ([3]), ChangeNet ([4]) and proposed method for images from TSUNAMI dataset.

TABLE VI: Quantitative comparison between different baseline network architectures used in our model for feature extraction. The scores are reported for binary segmentation in VL-CMU-CD dataset.

Feature extractor	Precision	Recall	f-score
Vanilla	0.910	0.904	0.908
DenseNet-121 [11]	0.915	0.916	0.916
DenseNet-161 [11]	0.923	0.924	0.924
DenseNet-201 [11]	0.922	0.924	0.923
ResNet-50 [12]	0.884	0.876	0.88
ResNet-101 [12]	0.881	0.847	0.864
ResNet-152 [12]	0.803	0.800	0.802
GoogleNet [13]	0.887	0.806	0.845
VGG-16 [9]	0.919	0.918	0.919
VGG-19 [9]	**0.941**	**0.939**	**0.940**

3) Qualitative comparison: We show the results generated by SuperPixel, ChangeNet and our method for images from GSV, TSUNAMI and VL-CMU-CD datasets in Fig. 1, 3, 4, and 5. In Figure 3, we show few examples from VL-CMU-CD dataset and the corresponding multi-class predictions from CDNet, ChangeNet and our proposed model. From the results, we observe that CDNet method wrongly predicts non-changing regions also as change regions in second, third and fifth row examples in Figure 3. The output of ChangeNet is better than CDNet, however the boundaries are not accurate.

TABLE VII: The quantitative comparison of our method with other approaches for FPR = 0.1 and FPR = 0.01 in VL-CMU-CD dataset. The best scores are highlighed in **bold** and the second best in blue color.

	FPR = 0.1			FPR = 0.01		
Metrics (→) Methods (↓)	Precision	Recall	f-score	Precision	Recall	f-score
Super-pixel [3]	0.17	0.35	0.23	0.23	0.12	0.15
CDnet [1]	0.40	0.85	0.55	0.79	0.46	0.58
ChangeNet [4]	0.79	0.80	0.79	0.80	0.79	0.79
CDNet++	**0.91**	**0.90**	**0.91**	**0.88**	**0.97**	**0.92**

Comparitively, our proposed method localizes changes well and segments them with reasonably accurate boundaries. In Figure 1, we show few examples for binary segmentation in VL-CMU-CD dataset.

4) Ablation experiments: In Table VI, we present the ablation experiments for different baseline network architecture used for feature extraction. We evaluate them for binary segmentation in the VL-CMU-CD dataset and present three metrics in Table VI. By vanilla architecture, we refer to simple seven convolution layers without any max-pooling and transpose convolution layers. We experimented with some of the famous CNN architectures like DenseNet [11], ResNet

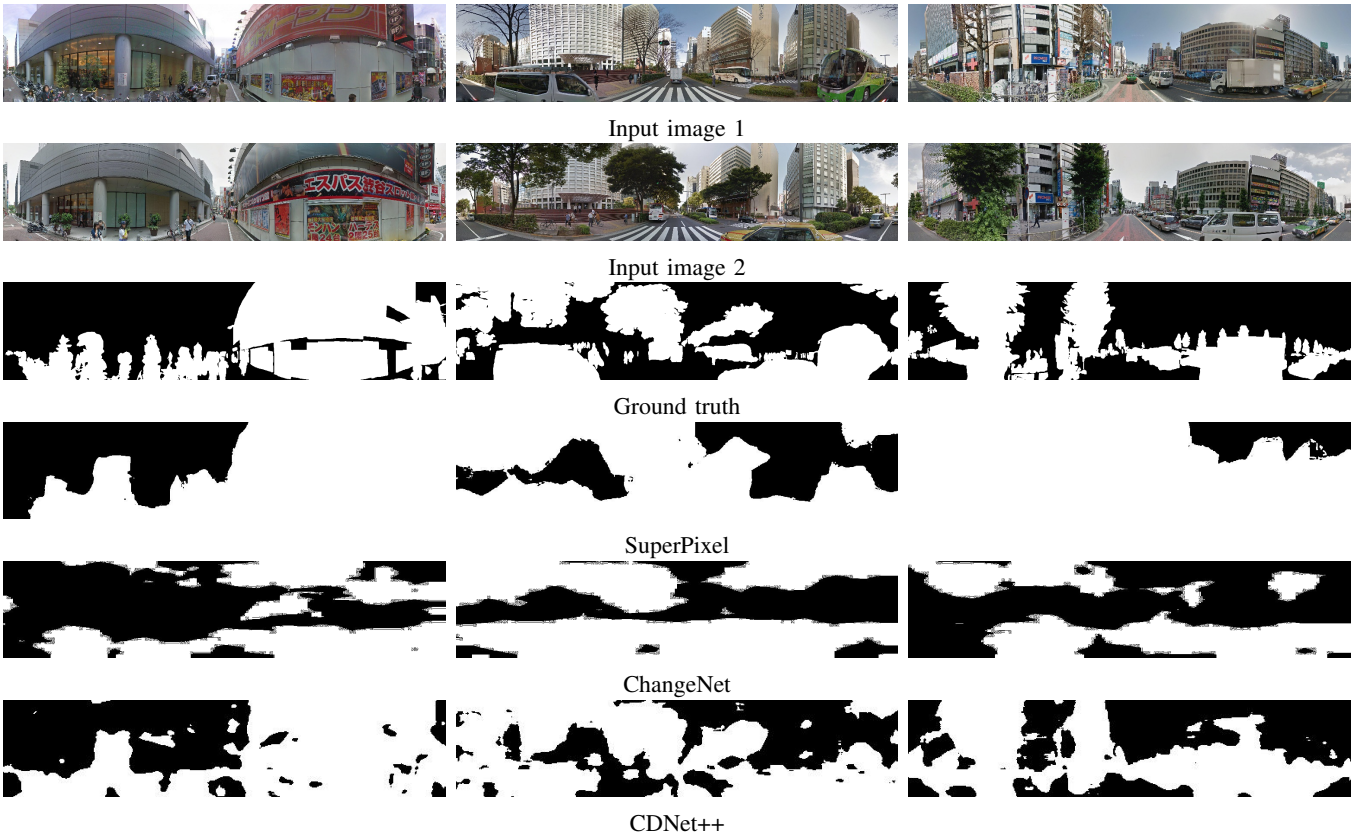


Fig. 5: Qualitative comparison with SuperPixel ([3]), ChangeNet ([4]) and proposed method for images from GSV dataset.

TABLE VIII: Performance metrics of CDNet++ for binary classification (change or no-change) on three different datasets.

Metric (→) Dataset (↓)	Accuracy	Precision	Recall	F1	mIoU	MCC	Sensitivity (TPR)	PWC	Specificity (TNR)	FPR	FNR
TSUNAMI	0.921	0.845	0.881	0.863	0.827	0.808	0.881	0.078	0.938	0.062	0.119
GSV	0.840	0.629	0.759	0.688	0.665	0.587	0.759	0.159	0.865	0.135	0.241
VL-CMU-CD	0.992	0.94	0.94	0.94	0.938	0.935	0.94	0.008	0.865	0.005	0.060

TABLE IX: Ablation study on number of convolution layers and filters used after transpose convolution and after feature concatenation. The best performing architecture is highlighted in bold and second best in blue.

# of filters before concatenation	# of filters after concatenation	Precision	Recall	f-score
128, 64, 32, 16, 8	128, 64, 32	0.923	0.935	0.927
256, 128, 64, 32, 16	128, 64, 32	0.941	0.950	**0.942**
64, 64, 64, 64, 64	128, 64, 32	0.929	0.947	0.938
256, 256, 256, 256, 256, 256	128, 64, 32	0.915	0.947	0.929
64, 64, 64, 64, 64, 64	128, 64, 32	0.943	0.937	0.939
32, 32, 32, 32, 32, 32	128, 64, 32	0.928	0.918	0.923
32, 32	128, 64, 32	0.940	0.939	0.938
128, 128	128, 64, 32	0.886	0.899	0.896
64, 64, 64	128, 64, 32	0.944	0.942	0.941
256, 256, 256, 256, 256, 256	256,128,64,32,16,8	0.921	0.916	0.921

[12], GoogleNet [13] and VGG [9]. We see from the table that VGG-19 performs better than other architectures.

In Table IX, we show the ablation for a different choice of layers and filters to use after transpose convolution. The highest f-score is achieved by using five convolution layers with the number of filters $\{256, 128, 64, 32, 16\}$ after transpose convolutions, followed by three convolution layers after concatenation with number of filters $\{128, 64, 32\}$. The second highest score is achieved by using three convolution layers with 64 filters each. For our experiments, we choose the second-best architecture as the difference between the second best and the best architecture is minimal (only in the third decimal point), while it has fewer parameters.

IV. Conclusion

In this paper, we have presented CDNet++, a novel CNN-based method for detecting changes from a pair of images. We handle the changes at various semantic levels, from simple structural change to illumination or seasonal changes, by using low-level to high-level convolutional features extracted at different depths in our encoder. Additionally, we make use of the correlation layer to handle the misregistration between

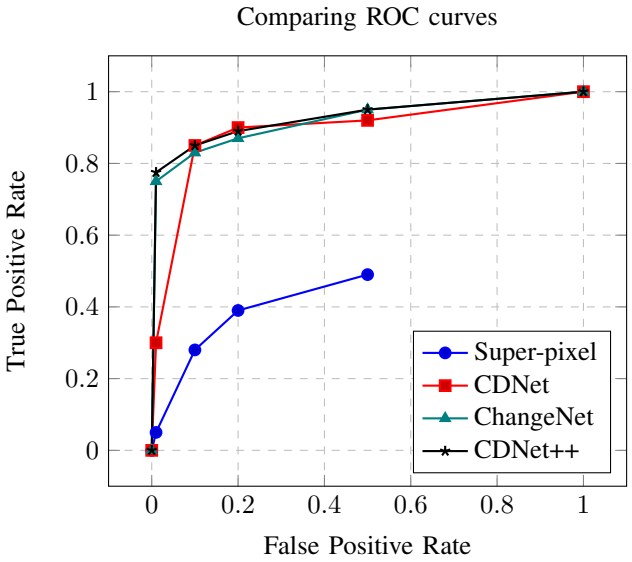


Fig. 6: ROC and TPR-FPR curve for binary class segmentation.

input pairs. Hence, the burden of having a perfect pixel-to-pixel alignment is alleviated. Through extensive evaluation on three different datasets, we have shown that our proposed CDNet++ method offers better accuracy over existing state-of-the-art methods. In particular, CDNet++ offers 14% boost in f-score at VL-CMU-CD dataset.

REFERENCES

[1] P. F. Alcantarilla, S. Stent, G. Ros, R. Arroyo, and R. Gherardi, "Street-view change detection with deconvolutional networks," *Autonomous Robots*, vol. 42, no. 7, pp. 1301–1322, 2018.

[2] K. Sakurada and T. Okatani, "Change detection from a street image pair using cnn features and superpixel segmentation." in *BMVC*, 2015, pp. 61–1.

[3] J. Gubbi, A. Ramaswamy, N. Sandeep, A. Varghese, and P. Balamuralidhar, "Visual change detection using multiscale super pixel," in *Digital Image Computing: Techniques and Applications (DICTA), 2017 International Conference on*. IEEE, 2017, pp. 1–6.

[4] A. Varghese, G. Jayavardhana, R. Akshaya, and P. Balamuralidhar, "Changenet: A deep learning architecture for visual change detection," in *European Conference on Computer Vision Workshops (ECCVW)*. IEEE, 2018.

[5] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 780–785, 1997.

[6] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition*. IEEE, 1999, p. 2246.

[7] M. S. Allili, N. Bouguila, and D. Ziou, "A robust video foreground segmentation by using generalized gaussian mixture modeling," in *Fourth Canadian Conference on Computer and Robot Vision*. IEEE, 2007, pp. 503–509.

[8] L. A. Lim and H. Y. Keles, "Learning multi-scale features for foreground segmentation," *arXiv preprint arXiv:1808.01477*, 2018.

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

[10] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

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