IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 23, 2020, accepted July 7, 2020, date of publication July 22, 2020, date of current version August 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3011187

OEBR-GAN: Object Extraction and Background Recovery Generative Adversarial Networks

DEBAPRIYA HAZRA AND YUNG-CHEOL BYUN
Department of Computer Engineering, Jeju National University, Jeju 63243, South Korea

Corresponding author: Yung-Cheol Byun (ycb@jejunu.ac.kr)

This work was supported by the Ministry of Small and Medium-sized Enterprises (SMEs) and Startups (MSS), South Korea, through the "Regional Specialized Industry Development Program (Research and Development)" supervised by the Korea Institute for Advancement of Technology (KIAT) under Grant S2855401.

ABSTRACT Generative adversarial networks (GAN) have been widely used in the field of image-to-image translation. In this paper, we have proposed a novel object extraction and background recovery (OEBR-GAN) model, which can extract objects from an image and then complete the image by inpainting the background of the image. This model has been developed for a solar panel installation project, where the user would like to input an original colored image of the roof, and as output, the user requires an edge detected roof image. However, the condition in user requirement is that any object that is hiding the roof edges should be removed first and the background of that part of the roof image should be recovered so that the user can obtain a complete connected edge detected image of the roof. Therefore, the model also completes the image by connecting the hidden edges of the roof. We could achieve the user objective by building a GAN model with a dual generator and dual discriminator network. The generators have been built using an encoder-decoder network with and without skip connections and the discriminators have been built using deep convolutional neural networks and encoder architecture. Quantitative comparisons in the result section shows that OEBR-GAN performs much better than other adversarial models on our collected dataset.

INDEX TERMS Generative adversarial networks, object extraction, background recovery, dual generator, dual discriminator.

I. INTRODUCTION

The most striking feature of deep learning is the ability to learn from both unstructured and unlabeled data. Currently, there is a dire need to extract relevant information from the abundance of untagged data available in every kind of industry. When processing an image, the raw input is ran through a lot of layers in deep learning to transform into the required output. Data traversal through multiple layers plays an important role in automatic feature learning which helps to extract complex and detailed information.

Generative adversarial networks (GAN) has been introduced in 2014 to gain knowledge about the latent space of a dataset and to generate realistic output [1]. GAN has shown amazing outputs for image-to-image translations and many GAN models have been built for this purpose. In the field of object detection or semantic segmentation, GAN generated better results in comparison to a traditional convolutional neural network with a small dataset. A lot of work has been done in building models for background inpainting in images. However, to the best of our knowledge, no work has been done on merging these approaches. In this paper, we have proposed a model that takes in Canny edge detected roof images as input to train the generator G_1 on inpainting the parts of the images that are to be considered as obstruction (objects hiding the roof edges). The discriminator D_1 then learns to differentiate between images that have obstructions and images that have no obstructions. The next generator and discriminator (G_2, D_2) learn to map the original colored image to the image generated from D_2, i.e. edge detected image where all the roof edges are connected after object extraction and background recovery. We have trained our model with a paired dataset.

We have tested our methodology with other image translation GANs but our architecture has the following advantages over other image translation GAN and produced the best result.

The associate editor coordinating the review of this manuscript and approving it for publication was Jon Atli Benediktsson.

- One-to-one paired image style migration model
- Takes pairs of images to be able to learn the translation from one image domain to another domain.
- Through adversarial training our model learns to remove obstructions that are hiding the roof edges.
- Connects the boundaries after object removal and then extract the complete edges of the roof images automatically.

II. RELATED WORK

The basic GAN architecture was built using a single generator and discriminator which compete against each other and represents a two-player min-max game. After the introduction, the same architecture was reused to form different models for various applications. Among other GANs, one architecture that is defined for finding irregularities in videos is the adversarial visual irregularity detection GAN (AVID-GAN) [2]. It has a generator that they named it as $\mathcal{I}$ as it inpaints the irregularity in the video and discriminator D which is a patch-level detector. The author claims that even in cases where there were no irregular videos to be used during training, AVID-GAN could correctly detect the irregularities.

Guilin Liu *et al.* recommend the utilization of partial convolutions, wherein the convolution is concealing and renormalized to be molded on solely valid pixels [3]. Because current deep learning-based image inpainting strategies in the research field utilize a trendy convolutional network over the undermined picture, making use of convolutional channel responses molded on both valid pixels as well as the substitute values in the masked gaps. This often prompts to artifacts such as shading discrepancy and haziness. Postprocessing is typically used to diminish such relics yet are costly and might fail. They similarly incorporate a mechanism to automatically produce a refreshed mask for the following layer as characteristic of the forward pass. Their proposed model beats other strategies for irregular masks. They show subjective and quantitative correlations with different approaches to validate their methodology.

Recently the research in automatic image manipulation has advanced exceptionally. But the main unrolled focus of such manipulation techniques are object centric images. Shetty *et al.*, proposed a model for object removal from general scene image [4]. They have proposed a two-phase architecture for object removal. They use an auxiliary network for predictions of mask during the training phase and at interface, clients can give manual masks. Results display that performance of the proposed model, and a fully supervised segmented based removal is the same. It illustrates the suitability of weakly supervised solutions for the general scene-stage enhancing task. However, their model cannot deal with new item generation.

Maximum utilization of GAN architecture has been done for image translation domain [5]–[7]. Conditional adversarial networks (cGAN) [18], dual discriminator or D2GAN, SingleGAN, pix2pix-GAN, cycle-GAN, and many more GANs have been developed for the transformation of images. SingleGAN uses a single generator for multi-domain image-to-image translation. The work produced domain code for one-to-many, one-to-one and many-to-many domain translation [7]. D2GAN uses dual discriminators to solve one of the disadvantages of GAN, i.e. mode collapse [6]. Alireza *et al.* used the GAN architecture to build adversarial autoencoders that implement variational inference [8], [16]. Hao Tang *et al.* used edges to tackle structural information in detail, since semantic labels lacks in detailed information generation [9]. Also to generate finer edges from colored images, Kamyar Nazeri *et al.* proposed EdgeConnect [10]. Much work has been done in anomaly detection in crowded scenes. The author claims that [11] is different from other models as there are no hand-crafted features and the activity pattern is learned from original pixels using deep learning methods. Anamoly detection is generally done using convolutional neural networks which produce poor results as compared to adversarial networks [11]–[13]. GANs have proved to be the best approach for automatic image synthesis and overpowers the traditional machine learning methods [14], [15]. Super-resolution GANs were developed to extract detailed texture from the super-resolved image. This can differentiate super-resolution images from photo-realistic images [17].

Supervised, unsupervised, self-supervised learning; all can be done with GAN. Hence, it has a huge advantage while considering techniques like inpainting which is highly in demand for document retrieval, damaged artwork fixation and many more [19]–[21]. For object detection, ImageNet has a set of a database that holds approximately 500 images per node and is very useful for the researcher. Every year ImageNet holds a challenge for object detection and image classification [22], [26]. These databases have been widely used in image inpainting based on marching method, multi-scale patch synthesis and evaluated with contextual losses [23]–[25].

There are multiple metrics or methodologies to evaluate GAN performance. Inception score, Frechet inception distance, precision, recall, and F1 score; all of these help in the evaluation of GAN performance [28]–[30].Theis *et al.* mentioned in their paper that compressing, inpainting, de-noising can be addressed by adversarial networks or GAN [30], [32]. In [30], the author explained Parzen window estimates, average log-likelihood and visual fidelity as evaluation criteria required for GAN architecture. Another method that has been adopted among many works done for mode collapsing is using multiple generators [31], [32]. The author stated that the Jensen-Shannon divergence(JSD) between the generators' and original data distribution is minimal, but among generators' distributions is maximal, and therefore it can restrict from mode collapsing [33].

III. IMPLEMENTATION OF OEBR-GAN

A. BASE MODEL

Generative adversarial network (GAN) introduced in [1] uses a training set to generate a new data set. Since it's

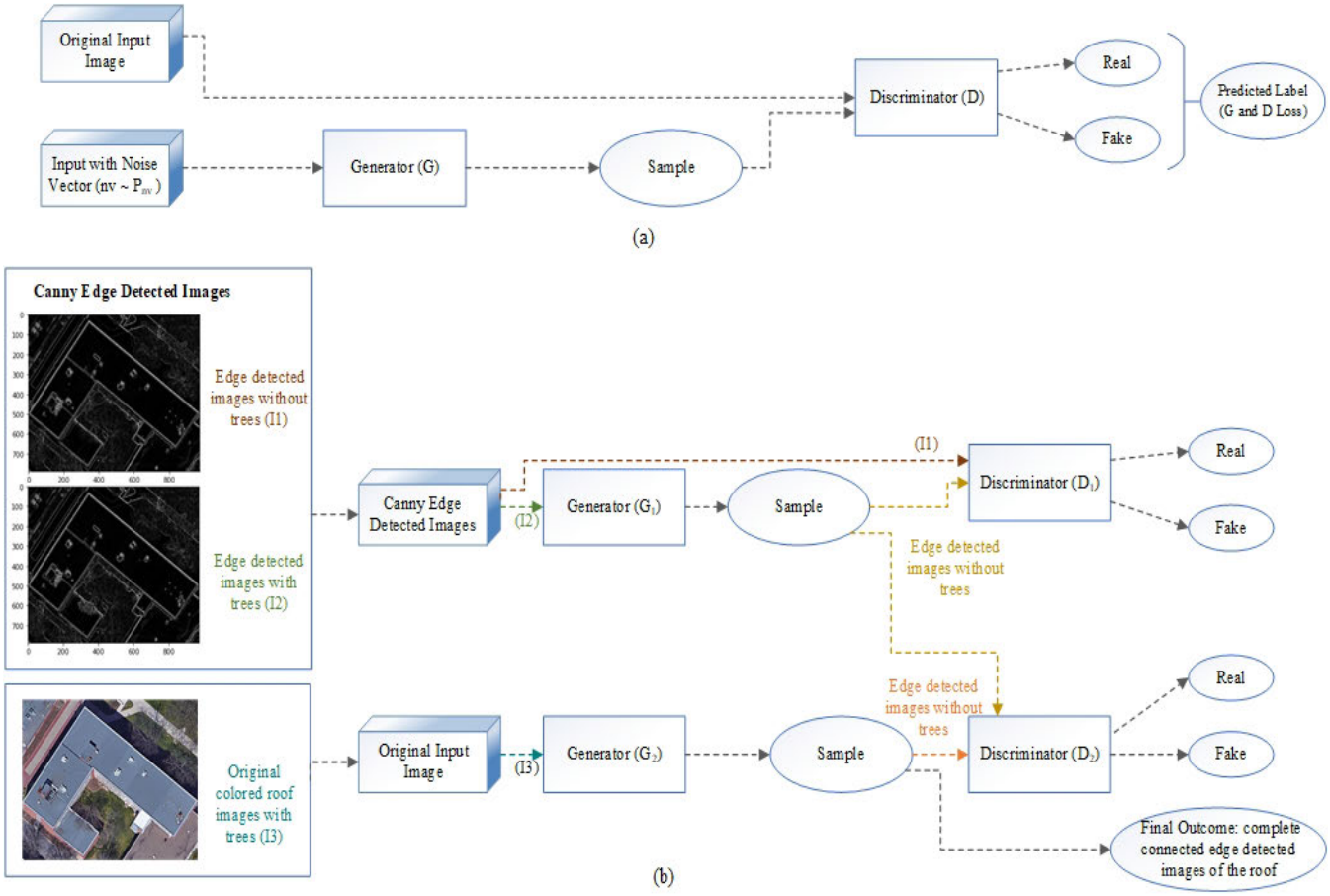

FIGURE 1. Base model illustration of (a) GAN and (b) OEBR-GAN.

invention, many people have used GAN architecture to build their model. To date, there are over five hundred GANs created to solve multiple real-world issues. It is not only useful for unsupervised learning but also for fully supervised learning, reinforcement learning and semi-supervised learning. GAN uses two neural networks namely generator G and discriminator D which compete against each other in a zero-sum game. The generator receives original input data x, adds noise vector P_{nv} and generates the sample $G(nv)$. The sample is then passed to the discriminator $D(x)$ which evaluates the probability of x being from data distribution P_{data} or from the generator. Both the neural networks are trained simultaneously. G is trained to minimize $log(1 - D(G(nv)))$ and D is trained to correctly recognize training samples and data from G. So, the two-player minimax game is defined in (1):

$$\min_G \max_D I(G, D) = \mathbb{E}_{x\sim P_{data}(x)}[logD(x)] + \mathbb{E}_{nv\sim P_{nv}}[log(1 - D(G(nv)))] \quad (1)$$

Our main goal in this paper is to extract roof edges. We have trained our adversarial network to perform following tasks:

- Connect the edges automatically ignoring any object that is obstructing the boundaries of the roof.
- After connecting the boundaries of the roof, producing the complete edges of a roof image automatically.

The proposed adversarial network can also be used for a general-purpose problem in which extraction of an object and background recovery in an image is required. Hence, we have named this model as object extraction and background recovery generative adversarial network or OEBR-GAN. OEBR-GAN uses two generators and two discriminators to remove the objects obstructing the roof and to generate edge detected images of the roof. A simple base model of how our architecture differs from GAN is given in Fig. 1

Generator G_1 is provided with Canny edge detected roof images B. G_1 adds noise samples to the input images B and inpaints the obstructions which hides the visibility of the roof edges. By segmenting the obstructions G_1 tries to fool the discriminator D_1. D_1 is also provided with the Canny edge detected roof images. Since D_1 is aware of the original distribution of data, it learns to identify the obstructions in the images. G_1 and D_1 trains themselves through self-supervised learning. This way, D_1 will be able to produce edge detected images of the roof where the objects hiding the roof edges

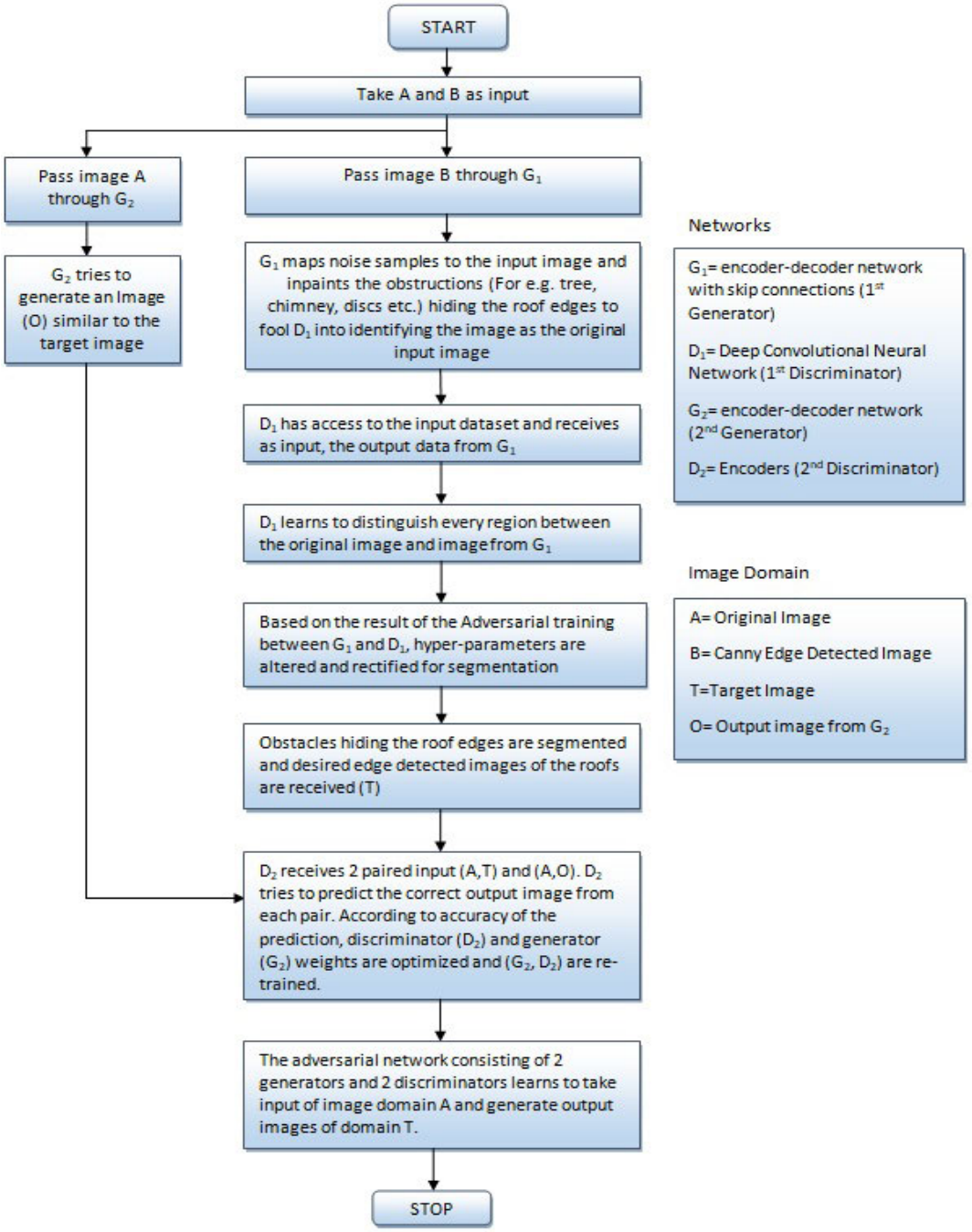

FIGURE 2. Workflow of OEBR-GAN.

have been removed. This adversarial model of G_1 and D_1 is influenced by Adversarial Visual Irregularity Detection or AVID-GAN [2].

Generator G_2 gets the corresponding original image of the Canny edge detected roof images. As GAN concept works, G_2 tries to fool the discriminator D_2 which receives samples from G_2 as well as from D_1. Therefore D_2 receives paired input and generates the prediction. The final output is the images where the object obstructing the roof edges has been extracted and the background recovered to get complete connected edge detected images of the roof. The workflow of the proposed approach is given in Fig. 2. The sample of input and targeted output images is shown in Fig. 3.

B. CANNY EDGE DETECTION

We have used Canny edge detected roof images as input images in our model. Canny edge detection uses a multi-stage

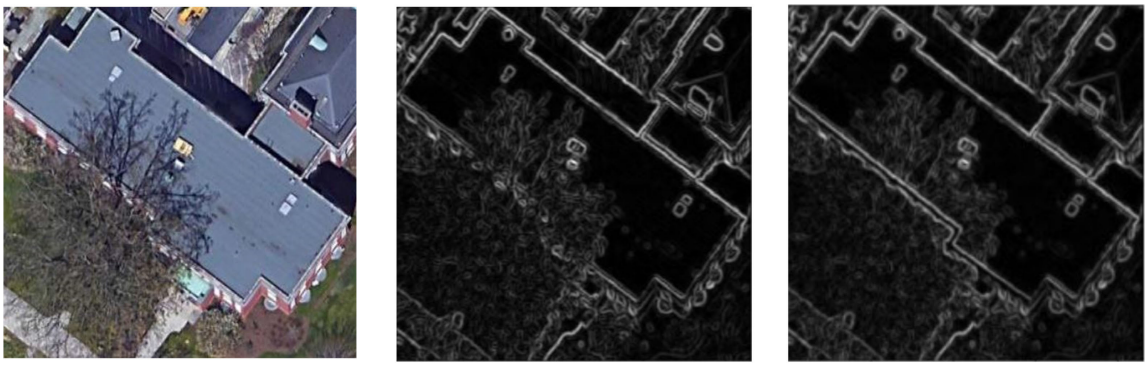

FIGURE 3. Input and target image sample (a) Original Image, (b) Canny edge detected image and (c)Target image without obstruction.

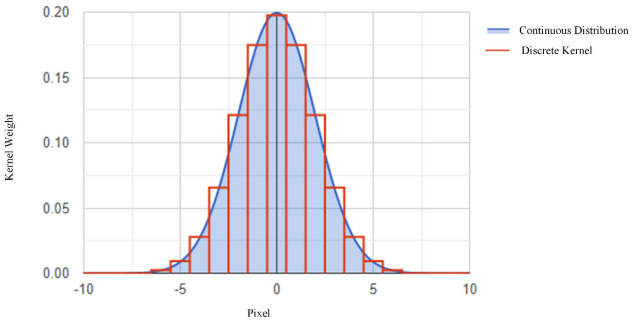

FIGURE 4. Plot for continuous distribution and discrete kernel approximation.

algorithm to detect edges in images [34]. The steps involved are:

- Grayscale conversion of colored input image.
- Applying Gaussian filter to extract the important edges by filtering out noise. We have used a Gaussian filter kernel size of 13 with value of sigma = 2. Fig. 4 shows the plot for Gaussian kernel calculation and the details of continuous distribution function and discrete kernel approximation.
- We have used Sobel edge detection operator in horizontal and vertical direction which helped in determining the intensity gradient of the images.
- Non-maximum suppression has been done with interpolation
- To eliminate edges with poor gradient values and keep edges with high gradient values we selected double threshold values i.e. low and high threshold values.
- Edge tracking by hysteresis and then we receive the edge detected output images from Canny edge detection algorithm.

C. GENERATOR NETWORK

For generator G_1, we have used an encoder-decoder convolutional network with skip connections. It's a 16 layer architecture. The encoder is utilized here to extract features from the image. It preserves the required components from the image and eliminates the obstructions that are hiding the roof edges in the image. While the decoder is used to retrieve the important details of the image after the obstruction removal. The classification is done per-pixel wise and skip connections are included from the encoder to the corresponding decoder. The extracted feature map is passed to the next layer of the encoder and to the mirrored decoder. Our framework consists of two stride-1 convolutional and transposed convolutional layers for downsampling and upsampling. For a specific decoder, it's input or output size has been kept equal to it's mirrored encoder output or input for achieving pixel-wise representation for a skip connection.

Generator G_2, on the other hand, has been built using the encoder-decoder network without skip connections. We have tested the generator G_2 network with and without skip connections on the training set but in our case G_2 without skip connections produced better results in terms of loss over time. Fig. 5 shows the difference in loss over time for generator G_2 with and without skip connections.The encoder and decoder are part of a convolutional and deconvolutional layer. During processing, each layer in the network receives input from the preceding layer and the current layer has to continuously adjust to the new distribution [35], this leads to a covariate shift. This issue is solved in our architecture using batch normalization. Batch normalization speeds up the training process and improves the performance of the model. The generator network G_1 and G_2 is described in Fig. 6.

D. DISCRIMINATOR NETWORK

Normally discriminator in normal GAN architecture provides judgment about whether the entire sample is from the original data distribution or the generator. But, in our proposed methodology, the discriminator D_1 can differentiate or judge every region of the input image. Therefore, the discriminator D_1 can identify which part of the image is referred to as obstruction hiding the roof edges, by the generator.

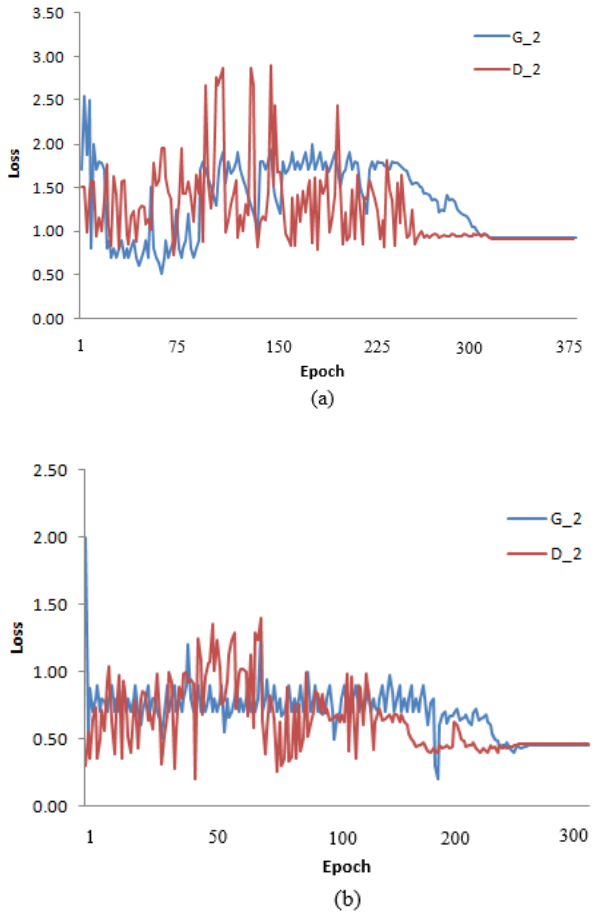

FIGURE 5. Generator G_2 loss over time (a) with skip connections and (b) without skip connections.

Discriminator D_1 comprises of various 1×1 convolutional layers. The output for D_1 is represented by matrix $OUT \in \mathbb{R}^{n_i \times n_j}$, where OUT(i,j) refers to i^{th} and j^{th} region of the image. We have used 12 feature maps with a 2×2 pooling operation, which helped in decreasing overfitting.

Discriminator D_2 receives paired input defined as below:

- D_2(A,T), where A is original image and T is Target image as shown in Fig. 3.
- D_2(A,O), where A is original image as mentioned in the above point and O is output image from G_2.

D_2 using the encoder architecture as shown in Fig. 7 predicts the correct output image from each pair. Depending on the accuracy of the prediction, parameters of D_2 and G_2 are optimized and are re-trained. The encoder architecture of D_2 helps in eliminating the disconnected edges of the roof. After D_1 outputs an edge detected roof image of the corresponding original image eliminating the obstruction; through the encoders, discriminator D_2 learns to connect the disconnected edges of the roof. Hence obtains a complete edge detected and connected roof image without any obstruction.

IV. EXPERIMENTS

A. DATASET

We have prepared a separate real-world dataset to test our algorithm since our main goal was to extract edges of a roof after eliminating any obstruction. Satellite images of roofs have been collected from Kakao Maps and Google Maps. The dataset has been splitted into training set containing 12750 images and test set of 6250 images prior to training. Mainly we have divided the training dataset into two parts:

- The collection of original images of rooftops
- The collection of the edge detected rooftop images of the original image. We have used Canny edge detection algorithm to obtain the edge detected images of the roof.

The test set consists of original images of roof that differs from the original set of the training data. The input roof images were 256×256 colored images. We use them as original data distribution which is fed to generator G_2. Generator G_1 receives Canny edge detected images of the corresponding original image as the input which is again of the size 256×256.

To verify the robustness of our proposed model, we have also used two other open-access datasets. The Inria Aerial Image Labeling dataset (IAILD) consists of pixelwise labeling of aerial images and mostly used for remote sensing [37]. There are two ground truths, building and not building for semantic classes. There are various images from urban, rural and also highly dense populated cities. Total no buildings covered in this dataset is 0.15 million. The other dataset is with the title Aerial Imagery for Roof Segmentation (AIRS) [38].This dataset contains roof images of more than 0.2 million buildings and it focuses on the largest city of New Zealand. It has also two semantic classes, roof and non-roof images. We have pre-processed both the dataset for experimental purpose. We have reconstructed the input roof images to 256×256 and divided the training and test data into 70% and 30% for both the dataset.

B. TRAINING DETAILS AND LOSS FUNCTION

Both the generator and discriminator are adversarially trained. Generator G_1 maps Gaussian noise sample to the original distribution during training which helps G_1 to differentiate between presence (X) and absence of obstruction ($\bar{X}$). Since D_1 has access to original distribution P_{data}, it can classify which region of $G_1(\bar{X})$ follows from the original distribution P_{data}. As we have mentioned earlier that the output for D_1 is represented by matrix $OUT \in \mathbb{R}^{n_1 \times n_2}$, here $n_1 \times n_2 = n$ represents the total number of area or region that is determined by D_1 to be considered as obstruction. So, the training of (G_1, D_1) can be optimized with the objective function in (2) [2], where $\mathcal{N}_\sigma$ is normal distribution with standard deviation.

$$\min_{G_1} \max_{D_1} (\mathbb{E}_{X \sim P_{data}}[log(\| D_1(X) \|^2)] + \mathbb{E}_{\bar{X} \sim P_{data} + \mathcal{N}_\sigma}[log(\| OUT - D_1(G_1(\bar{X})) \|^2)] \quad (2)$$

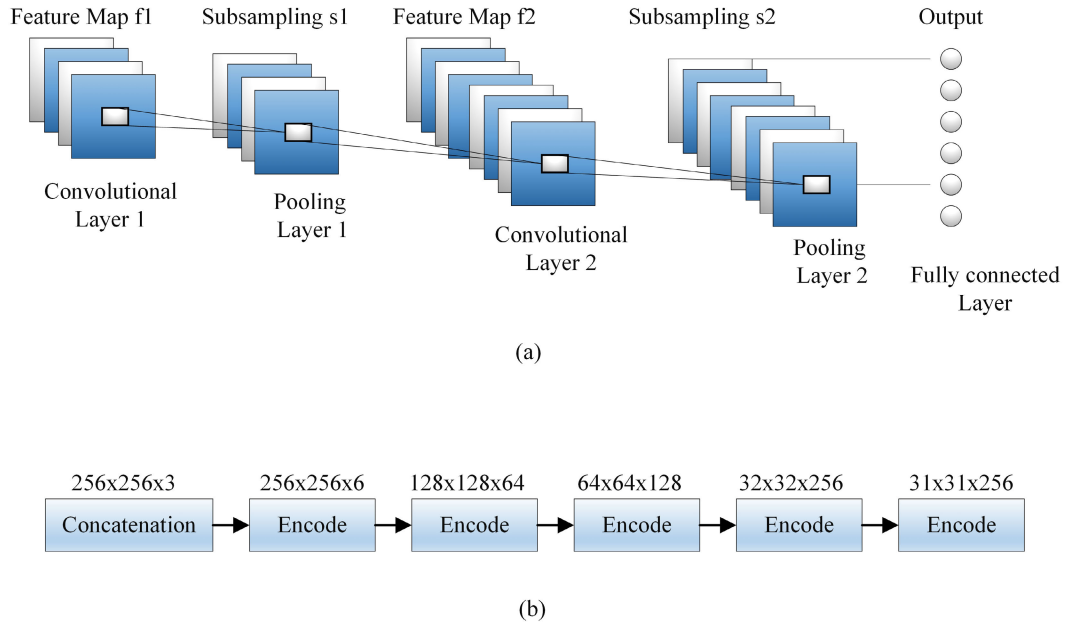

FIGURE 6. Architecture of (a) Generator G_1 and (b) Generator G_2.

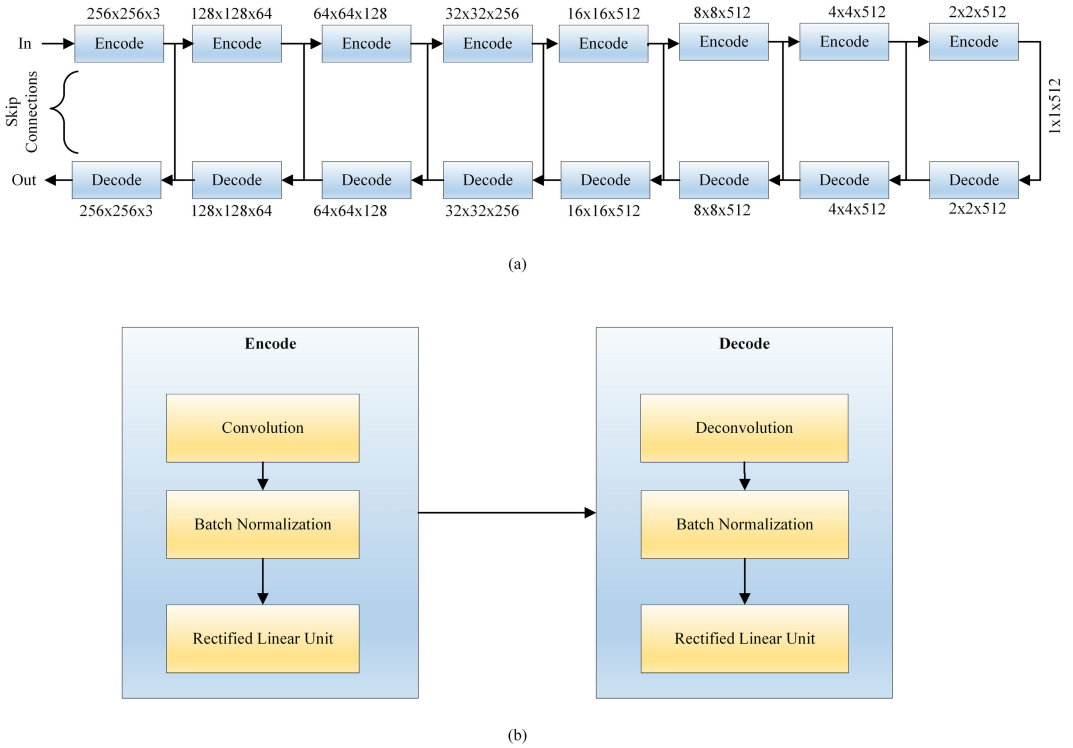

FIGURE 7. Architecture of (a) Discriminator D_1 and (b) Discriminator D_2.

We have used 300 epochs and there were 15900 iterations for the training. Both the generator and discriminator loss for training data is shown in Fig.5 and Fig. 8.

Since we have used two generators and discriminators in our architecture we will define the loss function as in (3) and (4). To measure the distance between the data generated by GAN architecture and the distribution of original data the below loss function is defined.

$$\mathcal{L}(G_1, D_1) = \mathbb{E}_{X \sim P_{data}}[log(D_1(X))] + \mathbb{E}_{\bar{X} \sim P_{data}}[log(1 - D_1(G_1(\bar{X})))] \quad (3)$$

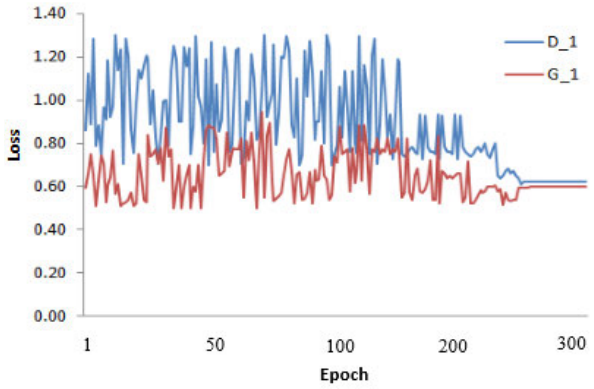

FIGURE 8. Generator G_1 and Discriminator D_1 loss over time.

$$\mathcal{L}(G_2, D_2) = \mathbb{E}_{x\sim P_{data}}[log(D_2(x))] + \mathbb{E}_{(G_1(\bar{X}))\sim P_{data}}[log(1 - D_2(G_2(\bar{X})))] \quad (4)$$

Our model has been implemented on TensorFlow 1.14.0 version, CUDA Toolkit version 10.1.243 and cuDNN version 7.6.5. The hardware configuration we used for developing is GeForce GTX 1060 3GB, driver version 436.15, Intel(R) Core(TM) i5-8500 CPU with 3.00GHz and 15.92 GB RAM.

To detect overfitting we have used measurement techniques like Median Recovery Error (MRE) and MRE-gap defined by Ryan Webster *et al.* in [36]. MRE defines the distribution of error into a single value. For a generator G and dataset X, MRE is defined as in (5)

$$MRE_G(X) = median[\min_z \parallel x_i - G(z) \parallel^2]_{x_i \in X} \quad (5)$$

To measure the extent to which the generator overfits, MRE-gap is computed on the training set. For training dataset D_t and validation dataset D_v MRE-gap is defined as in (6).

$$MRE-gap_G = [MRE_G(D_v) - MRE_G(D_t)]/MRE_G(D_v) \quad (6)$$

Ryan Webster *et al.* in [36] also evaluated p-value of the Kolmogorov-Smirnov test (KS) to estimate the amount of overfitting. Threshold of below 1% for p-value and above 10% for MRE-gap has been used to detect overfitting. For our model, value for Kolmogorov-Smirnov (KS) p-values, normalized median error difference (MRE-gap) and Median recovery errors (MRE) is given in Table 1. which indicates that our model prevents overfitting.

TABLE 1. Kolmogorov-Smirnov (KS) p-values, normalized median error difference (MRE-gap) and Median recovery errors (MRE).

KS p-value (Train vs Val)	MRE-gap (Train vs Val)	MRE (Train)	MRE (Val)	MRE (Generated)
3.65e-01	5.72e-02	3.01e-02	1.7e-01	1.2e-01

C. RESULTS AND MODEL COMPARISON

In this section, we show our results and compare the results of different models with our model OEBR-GAN. To evaluate the quality of our output images, or to measure how realistic the images are, we have used the inception score. Inception score is calculated by the following equation as in (7) [39],

$$e(\mathbb{E}_x[D_{KL}(p(y|x) \parallel p(y))]) \quad (7)$$

where e is an exponential function, D_{KL} is the Kullback–Leibler divergence, which measures how two probability distributions are different. $p(y)$ is the marginal distribution and $p(y|x)$ is probability of distribution of image x computed using an inception model. Along with inception score we have computed Learned Perceptual Image Patch Similarity (LPIPS) to measure the variance in the generated samples, Frechet Inception Distance (FID) has been used to quantify the feature vector distance between real and generated images. Lower value of FID indicates high quality images. Precision, Recall and F1 values has also been compared in Table 2 to evaluate our model's performance with other adversarial networks. Table 2. shows the quantitative score comparison of OEBR-GAN with five different models on roof images. Fig. 9 shows the comparison of output images from traditional edge detection method and OEBR-GAN model on our dataset. Fig. 10 displays the visualization results of our model compared with the existing GAN models mentioned in Table 2 and Table 3.

Also, in Table 3, computation of other evaluation metrics like mean l1 error, mean l2 error, peak signal-to-noise ratio (PSNR) and structural similarity index metric (SSIM) on our validation datasets shows that our model OEBR-GAN performs better compared to other models.

TABLE 2. Quantitative comparison on our dataset- Inception Score (IS), Frechet Inception Distance (FID), Learned Perceptual Image Patch Similarity (LPIPS), Precision, Recall and F1.

GAN Model Names	IS	FID	LPIPS	Precision	Recall	F1
Cycle-GAN [40]	7.98 ± 0.34	75.3	0.21	95.04	96.02	95.07
pix2pix-GAN [5]	9.04 ± 0.04	75.6	0.17	94.08	95.34	95.36
Contextual Attention [41]	8.56 ± 0.26	72.8	0.19	96.22	96.12	96.17
GatedConv [42]	9.62 ± 0.04	73.2	0.18	93.31	94.76	93.28
PartialConv [3]	9.90 ± 0.32	71.4	0.22	94.05	95.96	96.54
Pix2PixHD [43]	9.73 ± 0.65	70.5	0.20	95.45	96.78	95.84
SelectionGAN [44]	8.94 ± 0.44	71.7	0.21	93.92	94.71	95.87
OEBR-GAN	11.28 ± 0.02	69.5	0.14	97.88	97.02	97.83

TABLE 3. Mean l1 error, mean l2 error, PSNR and SSIM.

GAN Model Names	l1 error	l2 error	PSNR	SSIM
Cycle-Gan [40]	16.7%	5.7%	29.66	0.8943
pix2pix-GAN [5]	11.4%	4.6%	29.25	0.9237
Contextual Attention [41]	11.9%	5.1%	29.93	0.9164
GatedConv [42]	10.5%	5.4%	28.99	0.9067
PartialConv [3]	9.7%	4.9%	27.54	0.8972
Pix2PixHD [43]	10.4%	4.8%	28.61	0.7593
SelectionGAN [44]	12.4%	5.0%	26.92	0.6132
OEBR-GAN	8.5%	3.0%	34.51	0.9553

V. DISCUSSION

We propose the use of dual generator and discriminator in GAN architecture to extract objects, recover background, connect edges of the roof and to generate completely connected edge detected images of the roof from the original colored input roof images. Our architecture can be further

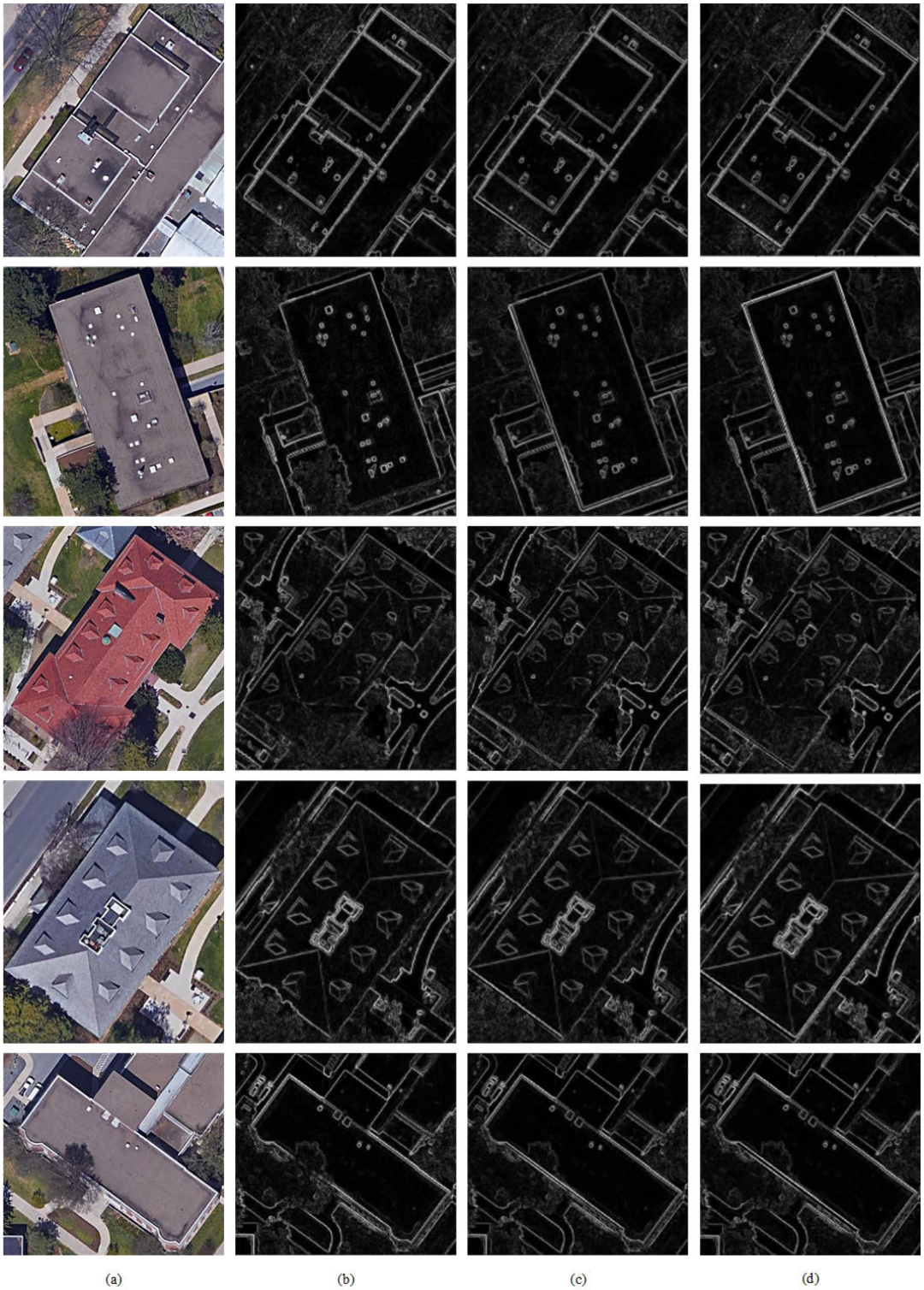

FIGURE 9. Images from Kakao and Google Maps (a) Original image (b) Canny edge detected image (c) Output image using OEBR-GAN and (d) Ground truth.

modified to be used for driverless car where the vehicle should be capable of sensing its environment. In this scenario, our model can be reformed to build applications that could automatically discover upcoming roads hidden with trees or any other objects by extracting the objects and recovering behind the object scenes. Further, it can also be adapted for videos. For instance, tracking or identifying masked object in CCTV surveillance videos.We have built the architecture

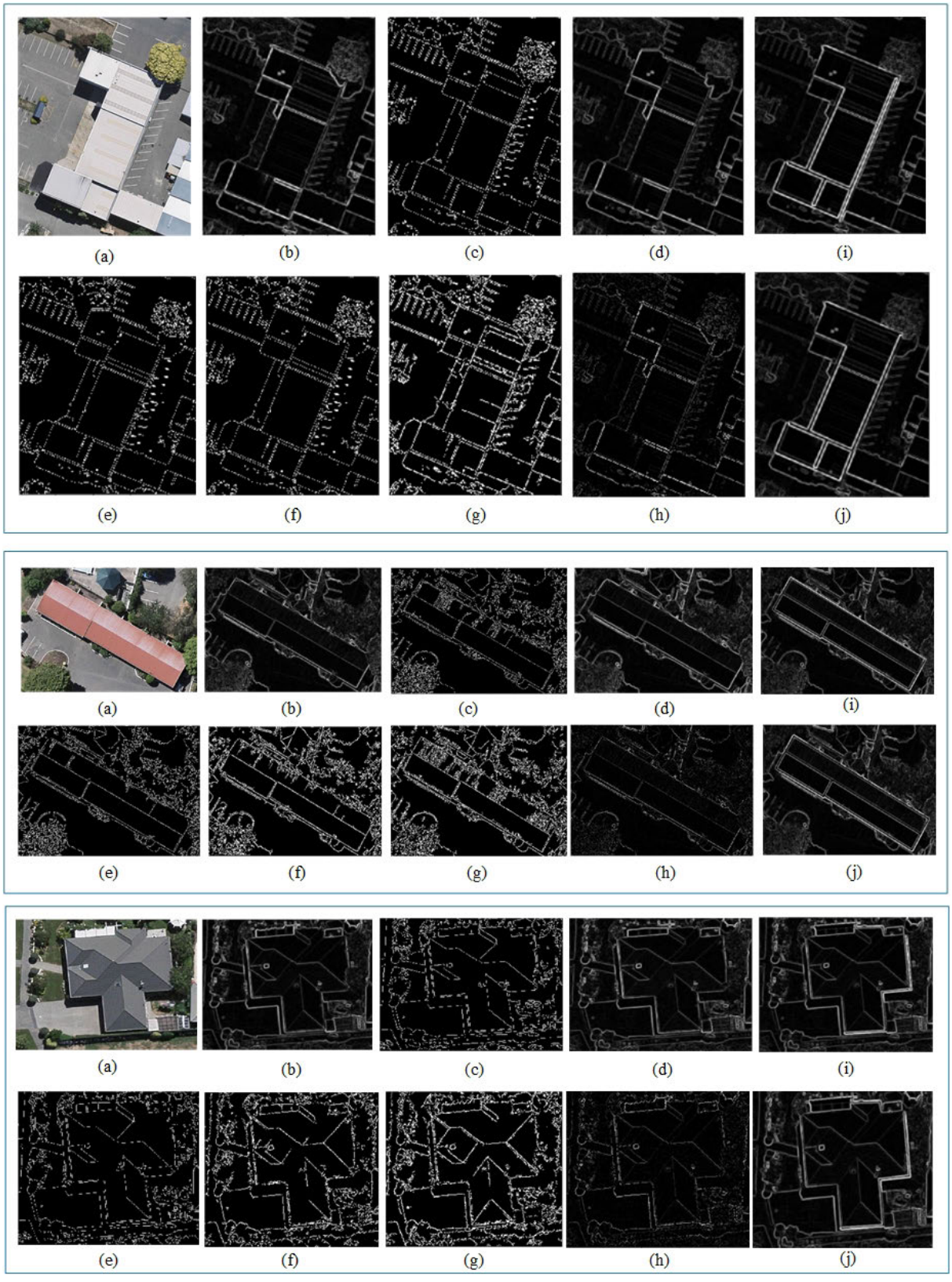

FIGURE 10. Output images from IAILD and AIRS dataset: (a) Original image (b) Cycle-GAN (c) pix2pix-GAN (d) Contextual Attention (e) GatedConv (f) PartialConv (g) Pix2PixHD (h) SelectionGAN (i) Ground truth and (j) OEBR-GAN.

for extract roof edges to ease the measurement process for solar panel installation. In the future, to make our model more applicable, we would like to train our model to identify and label each object individually, including shadows and make it more user-friendly by building a user interface and also include the calculation of area covered.

VI. CONCLUSION

Many image translation GANs achieve the transformation of an image from one domain to another. We have proposed a novel approach to eliminate objects from images, recover and complete the background of the images from where the object has been removed using GAN. Using two generators and two discriminators, we have named our architecture as OEBR-GAN. Values of evaluation metrics proves that OEBR-GAN produces better results as compared to other models. Our model can be trained on a large-scale dataset and is scalable.

In our defined methodology, OEBR-GAN is not only producing an image from one domain to other but also removing irregularities from the image and inpainting the image background as well as completing the image by connecting the edges of the roof. So, OEBR-GAN can be considered as a multipurpose GAN. The architecture for OEBR-GAN differs from other image translation GAN and it achieves better performance compared to other models according to the results.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[2] M. Sabokrou, M. Pourreza, M. Fayyaz, R. Entezari, M. Fathy, J. Gall, and E. Adeli, "Avid: Adversarial visual irregularity detection," in *Proc. Asian Conf. Comput. Vis.*, Cham, Switzerland: Springer, 2018, pp. 488–505.

[3] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 85–100.

[4] R. R. Shetty and M. F. B. Schiele, "Adversarial scene editing: Automatic object removal from weak supervision," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7706–7716.

[5] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

[6] T. Nguyen, T. Le, H. Vu, and D. Phung, "Dual discriminator generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2670–2680.

[7] X. Yu, X. Cai, Z. Ying, T. Li, and G. Li, "SingleGAN: Image-to-image translation by a single-generator network using multiple generative adversarial learning," in *Proc. Asian Conf. Comput. Vis.*, Cham, Switzerland: Springer, 2018, pp. 341–356.

[8] A. Makhzani, J. Shlens, N. Jaitly, and I. F. B. Goodfellow, "Adversarial autoencoders," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–16.

[9] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, "EdgeConnect: Generative image inpainting with adversarial edge learning," 2019, *arXiv:1901.00212*. [Online]. Available: http://arxiv.org/abs/1901.00212

[10] H. Tang, X. Qi, D. Xu, P. H. S. Torr, and N. Sebe, "Edge guided GANs with semantic preserving for semantic image synthesis," 2020, *arXiv:2003.13898*. [Online]. Available: http://arxiv.org/abs/2003.13898

[11] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 1577–1581.

[12] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. Sebe, "Training adversarial discriminators for cross-channel abnormal event detection in crowds," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1896–1904.

[13] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Comput. Vis. Image Understand.*, vol. 172, pp. 88–97, Jul. 2018.

[14] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423.

[15] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Learning to generate images of outdoor scenes from attributes and semantic layouts," 2016, *arXiv:1612.00215*. [Online]. Available: http://arxiv.org/abs/1612.00215

[16] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," 2015, *arXiv:1512.09300*. [Online]. Available: http://arxiv.org/abs/1512.09300

[17] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.

[18] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

[19] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2536–2544.

[20] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[21] S. Reed, A. van den Oord, N. Kalchbrenner, V. Bapst, M. Botvinick, and N. De Freitas, "Generating interpretable images with controllable structure," in *Proc. ICLR*, Toulon, France, 2017.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[23] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6721–6729.

[24] A. Telea, "An image inpainting technique based on the fast marching method," *J. Graph. Tools*, vol. 9, no. 1, pp. 23–34, Jan. 2004.

[25] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," 2016, *arXiv:1607.07539*. [Online]. Available: http://arxiv.org/abs/1607.07539

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[27] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," 2016, *arXiv:1611.02163*. [Online]. Available: http://arxiv.org/abs/1611.02163

[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[29] T. Le, H. Vu, T. Dinh Nguyen, and D. Phung, "Geometric enclosing networks," 2017, *arXiv:1708.04733*. [Online]. Available: http://arxiv.org/abs/1708.04733

[30] L. Theis, A. van den Oord, and M. Bethge, "A note on the evaluation of generative models," 2015, *arXiv:1511.01844*. [Online]. Available: http://arxiv.org/abs/1511.01844

[31] R. Wang, A. Cully, H. Jin Chang, and Y. Demiris, "MAGAN: Margin adaptation for generative adversarial networks," 2017, *arXiv:1704.03817*. [Online]. Available: http://arxiv.org/abs/1704.03817

[32] D. Warde-Farley and Y. Bengio, "Improving generative adversarial networks with denoising feature matching," in *Proc. ICLR*, Toulon, France, 2017.

[33] Q. Hoang, T. Dinh Nguyen, T. Le, and D. Phung, "Multi-generator generative adversarial nets," 2017, *arXiv:1708.02556*. [Online]. Available: http://arxiv.org/abs/1708.02556

[34] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167
[36] R. Webster, J. Rabin, L. Simon, and F. Jurie, "Detecting overfitting of deep generative networks via latent recovery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11273–11282.
[37] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? The inria aerial image labeling benchmark," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 3226–3229.
[38] Q. Chen, L. Wang, Y. Wu, G. Wu, Z. Guo, and S. L. Waslander, "Aerial imagery for roof segmentation: A large-scale dataset towards automatic mapping of buildings," 2018, *arXiv:1807.09532*. [Online]. Available: http://arxiv.org/abs/1807.09532
[39] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.
[40] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.
[41] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5505–5514.
[42] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, "Free-form image inpainting with gated convolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4471–4480.
[43] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8798–8807.
[44] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan, "Multi-channel attention selection GAN with cascaded semantic guidance for cross-view image translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2417–2426.

DEBAPRIYA HAZRA received the B.Sc. degree (Hons.) in computer science in 2012, and the M.C.A. degree in 2015. She is currently pursuing the Ph.D. degree in machine learning with Jeju National University, South Korea. She worked at Atos Global IT Solutions and Services Pvt., Ltd., for three and a half years, as a Software Engineer. Her current work involves generative adversarial networks, image inpainting, and medical image analysis in the field of machine learning.

YUNG-CHEOL BYUN received the B.S. degree from Jeju National University, South Korea, in 1993, and the M.S. and Ph.D. degrees from Yonsei University, in 1995 and 2001, respectively. He worked as a Special Lecturer at Samsung Electronics, from 2000 to 2001. From 2001 to 2003, he was a Senior Researcher with the Electronics and Telecommunications Research Institute, and he was promoted to join Jeju National University, as an Assistant Professor, in 2003, where he is currently a Professor with the Department of Computer Engineering. From 2012 to 2014, he had research activities at the University of Florida, as a Visiting Professor. His research interests include the areas of pattern recognition, image processing, artificial intelligence, machine learning, security based on pattern recognition, home network and ubiquitous computing, u-healthcare, RFID, and IoT middleware systems.