6

Domain invariant feature transform for interactive floor plan retrieval

Content-based image retrieval (CBIR) emerged as an attempt to deal not only with the absence or insufficiency of annotations for most of the images, but also to support alternative retrieval approaches, relying on visual perception, which is more appropriate in many scenarios. Within CBIR, sketch-based image retrieval (SBIR) aims at retrieving images that are similar to a sketch made by the user (typically a simple set of drawing lines). Thus, SBIR is particularly adapted in situations where a user has a perceptual image of what he is searching. In this scenario, a sketch image is useful specially when the image dataset is not annotated and the user has no similar example image to use as a query input. The challenges associated with SBIR are:

1. Finding a relevant visual content representation associated with a similarity measure that allows effective comparison with a query that is not a picture, but rather a drawing made by a user who sometimes is not very skillful.

2. Making retrieval scalable to large image datasets by building an appropriate index structure which is able to exploit the content representation and similarity measure in a better way.

3. Sketches are highly abstract and carry a huge level of variation from person-to-person.

4. Sketches don't provide much of visual cues like colour as present in images.

5. The inherent difference between two domains (sketch and image) also makes the problem more challenging.

In Chapter 3, 4 and 5, the proposed methodologies revolve around extracting semantic and structural features from floor plan images. Exploring both hand-crafted and deep learning features has been done in the previous chapters to extract and represent meaningful content in images and finally retrieve similar floor plan images from the database. The important thing to be noted here is that the mode of the query until this chapter is through an image. In this Chapter, the query mode is interactive, i.e., it can either be an image based query or a sketch based query. This Chapter is organized as follows: Sec. 6.1 gives a brief overview of the approaches proposed in this Chapter, Sec. 6.2 gives an insight into the first approach using Cyclic GAN that has been proposed in this Chapter and Sec. 6.3 details out the second approach using autoencoders and Cyclic GANs for the task of retrieval. Section 6.4 gives a qualitative and quantitative analysis of both approaches. Finally, Sec. 6.5 concludes the Chapter.

6.1 BRIEF OVERVIEW

Sketching using pen and paper is more intuitive and easy to represent ideas as compared to other modalities. With the popularity of digital hand-held devices, users are keen on using sketches to exchange ideas. Users can sketch the floor plans on a tablet in the same fashion as they would on

paper. With the increase in the usage of touchscreens and hand-held devices, sketch-based querying for finding a dream home can be a convenient proposition. Sketch-based floor plan retrieval can benefit the architects and the users while searching for floor plans in the early design phases of a building. However, sketch-based processing and retrieval come with their challenges. In this Chapter, techniques for retrieval of floor plans given an interactive query mode are proposed. Two approaches to facilitate sketch based retrieval of floor plans are presented in this Chapter. Figure 6.1 shows the flow charts of the two approaches.

1. **Approach 1:** An initial attempt aims at using a Cyclic Generative Adversarial Network (Cyclic GAN) for domain mapping between floor plan sketches and floor plan images along with Convolutional Neural Network (CNN) to extract features from the mapped query and retrieval set.

2. **Approach 2:** Although approach 1 performs well for sketch-based retrieval of floor plans, there is still a scope of improvement in terms of the precision values during retrieval. Also, there is no provision of image-based retrieval in the proposed approach 1. This serves as the motivation for an efficient unified framework to meet the requirements of both image to image as well as the sketch to image retrieval. Therefore, to improve the results of the retrieval mode and to propose a composite solution to multimodal retrieval, a unified framework for retrieval of similar floor plan images given query mode as floor plan sketches or images is proposed. This approach uses the conjunction of CNN for query classification, Cyclic GAN for sample generation and Autoencoder for domain mapping and proves to be efficient than approach 1 in terms of retrieval.

In the next section the methodology associated with Approach 1, taking into account Cyclic GANs and CNN for feature extraction and domain mapping is discussed in detail.

6.2 APPROACH 1: CYCLIC GAN FOR DOMAIN ADAPTATION AND SKETCH BASED FLOOR PLAN RETRIEVAL

In this section, a brief overview of the first approach for similar floor plan retrieval using sketch modality is discussed. Figure 6.2 depicts the complete framework of the proposed technique. If the query is a sketch, the steps carried out are as follows. The entire framework is divided into three main phases. They are: (1) domain mapping through Cycle Generative Adversarial Networks (Cyclic GANs); (2) feature representation through Convolutional Neural Networks and (3) matching and retrieval. Given an input sketch S, the goal is to retrieve similar looking floor plan images (I) from the ROBIN dataset [Sharma *et al.*, 2017] corresponding to the query sketch. The Cyclic GAN is trained using images from ROBIN and sketches from S-ROBIN to learn domain representation. As direct retrieval of floor plan images given a sketch is not straightforward (refer the low performance of CNN1 in Fig. 6.14. (b)). Therefore, first the ROBIN dataset images are converted into their sketch counterparts (called GAN generated sketch dataset (GS-ROBIN)) using a Cyclic GAN trained to map the sketch and image domain. Thereafter, the post-processing of the generated sketches (GS) is done to get their enhanced versions. Then a CNN is trained over the generated sketches to learn the deep feature representation for the floor plan retrieval task. The learned deep representation helps to extract deep features from the GS-ROBIN database. The domain mapping is necessary because of large variation between the quality of images/sketches generated using Cyclic GAN and original images/sketches from the ROBIN/S-ROBIN dataset, as shown in Fig. 6.3. During the retrieval stage, the framework extracts the deep features from the query sketch also using the same deep representation. The extracted deep features are then matched with the database features. A similarity score for a particular layout is thus calculated

for all the query samples. To know which individual layer of the deep framework performs best, similarity using all the individual deep features/layers is calculated, and at the same time, the effect of CNN trained on the ROBIN dataset, the S-ROBIN dataset, and the GS-ROBIN dataset is studied. Details of the same are discussed in Sec. 6.4.5.

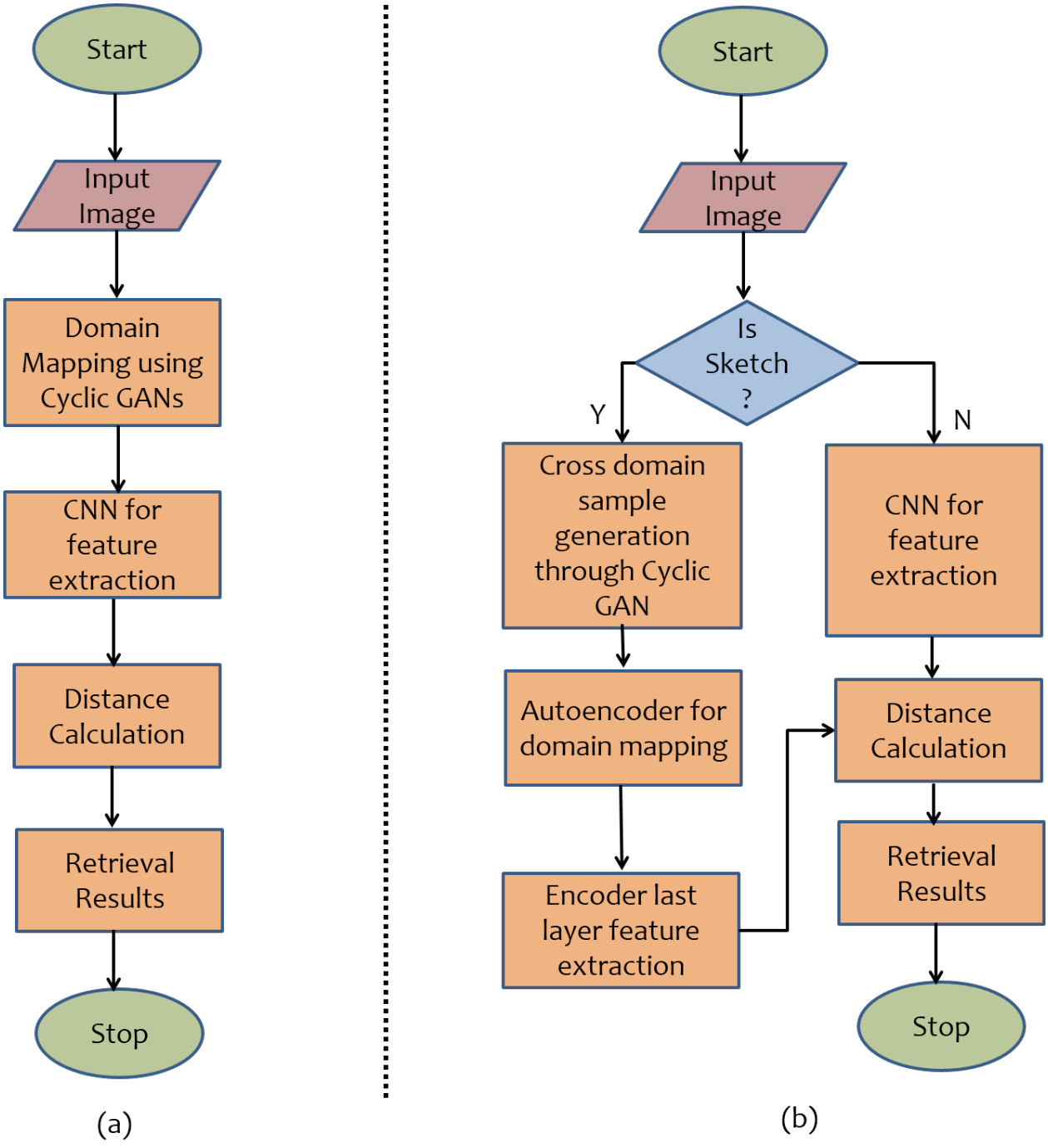

Figure 6.1. : Flowchart comparing the different sub-components in two approaches for sketch based retrieval. (a) Approach 1: Basic framework using cyclic GAN for domain mapping and sketch based retrieval (b) Approach 2: An improved unified framework using cyclic GAN and autoencoders in conjunction for floor plan retrieval.

6.2.1 Domain Mapping through Cyclic GAN

In this section, the use of Cyclic GAN for facilitating the "domain shift" between sketch and image floor plan domain is illustrated. In this case, the purpose is to choose a floor plan image domain (I) and then transform it into an image of sketched floor plan domain (S), and vice versa. Since one-to-one mapping does not exist between the sketch and the image floor plans in the training set, the problem becomes more challenging Zhu *et al.* [2017]. Having this relaxation of the non-existence of one-to-one mapping makes this formulation quite powerful. The need for a paired image in the target-domain is eliminated by the two-step transformation that involves mapping of source domain image to the target and back to the original image. Mapping the image to the target domain is improved by competing for the generator with the discriminator in the Cyclic GAN model. For training the Cyclic GAN model, a training set consisting of 70% samples from the ROBIN and S-ROBIN datasets is used.

Figure 6.4 depicts the overall objective of the proposed model for domain adaptation. Let, the image domain be denoted as I, and the sketch domain be denoted as S. An i^{th} floor plan image sample is denoted as f_I^i, where $i = 1,2,\ldots,m$. Here m is the number of floor plan image samples taken from ROBIN. On the other hand, the j^{th} floor plan sketch sample from S-ROBIN is denoted as f_S^j, where $j = 1,2,\ldots,n$. In this case, n is the number of sketches in S-ROBIN. To convert a floor plan sketch to its corresponding image version, a mapping function $G(.)$ is introduced. The task of $G(.)$ is to take a floor plan sketch from S-ROBIN and convert it to an image, which closely resembles a floor plan image ($\tilde{f}_I^j$) from ROBIN. This process is denoted in Fig. 6.4. Similarly, to convert a floor plan image from ROBIN to that of a sketch (belonging to S-ROBIN), another mapping function $F(.)$ is introduced. This function performs the same task as that of $G(.)$, but in the reverse order. The hypothesis is that the generated images, i.e., outputs of $G(.)$ and $F(.)$ will eventually match the target distributions. To compare between the actual and the desired distributions, two adversarial discriminators D_S and D_I have been introduced. The task of D_I is to check how good the mapping is from a given sketch to an image, whereas D_S aims to achieve the reverse. During the training phase, examples from both S-ROBIN and ROBIN dataset are given to the network to learn the mapping functions $G(.)$ and $F(.)$.

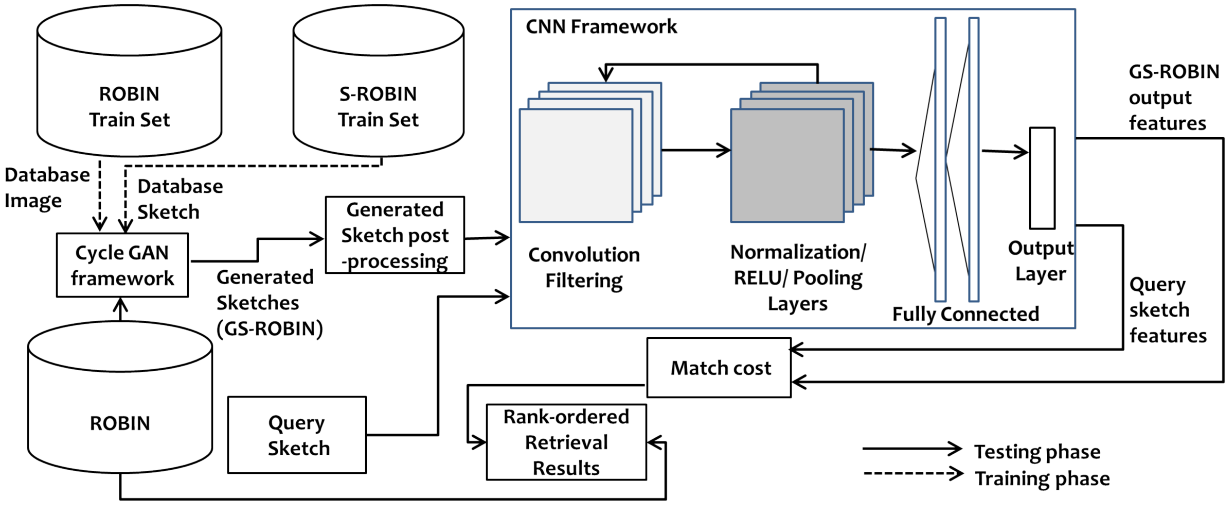

Figure 6.2. : Framework for sketch based retrieval of floor plans. Domain adaptation is facilitated by a Cyclic GAN.

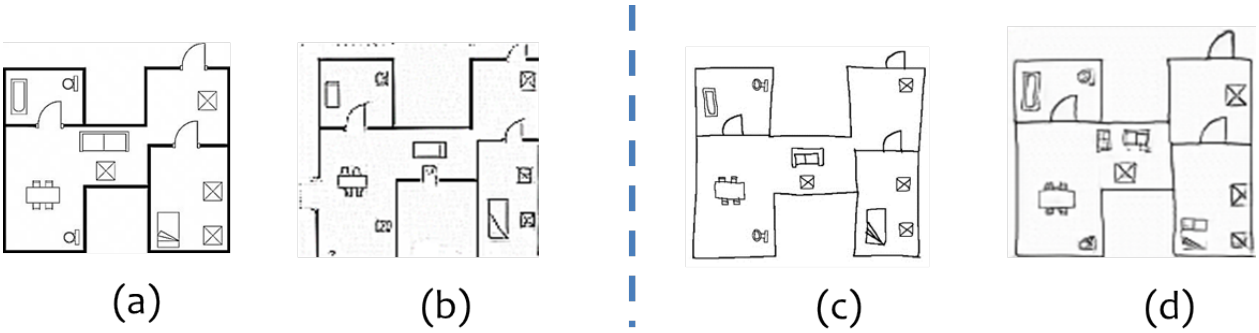

Figure 6.3. : Comparing the original sketches/images with outputs generated from Cyclic GAN. (a) Original floor image from ROBIN dataset. (b) Generated floor plan image using Cyclic GAN. (c) Original floor plan sketch from S-ROBIN dataset. (d) Generated floor plan sketch using Cyclic GAN. Kindly note the quality of generated sketches is better than that of generated images.

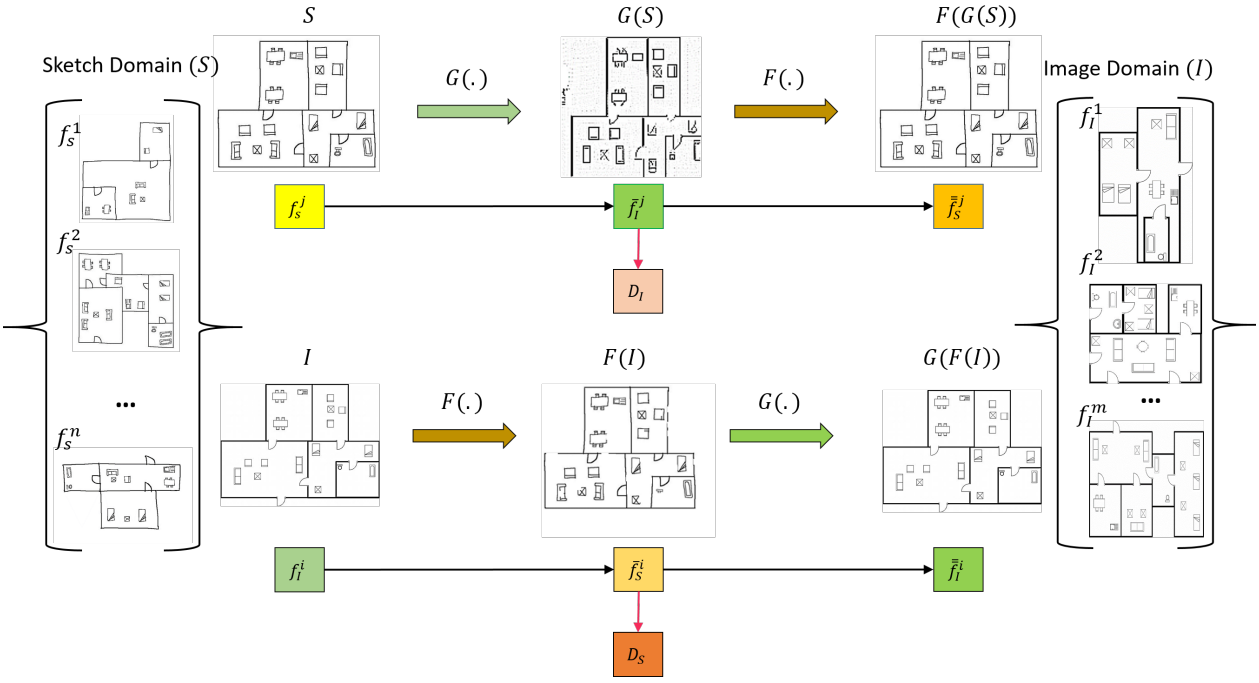

Figure 6.4. : Cyclic GAN framework, corresponding to the proposed approach with converting the sketches from the S-ROBIN dataset into GAN generated images (constituting GI-ROBIN dataset) closely similar in features to the to-be retrieved images and images from the ROBIN dataset to sketches (constituting GS-ROBIN dataset).

Let the probability distribution of a sample $f_S \in S$ be denoted as $f_S \rightarrow p(f_S)$, and for a sample $f_I \in I$ be denoted as $f_I \rightarrow p(f_I)$. Then the adversarial loss function for mapping a floor plan sketch to an image ($S2I_A(.)$) can be written as:

$$S2I_A(G, D_I, S, I) = E_{f_I \rightarrow p(f_I)}[logD_I(f_I)] + E_{f_S \rightarrow p(f_S)}[log(1 - D_I(F(f_S)))] \tag{6.1}$$

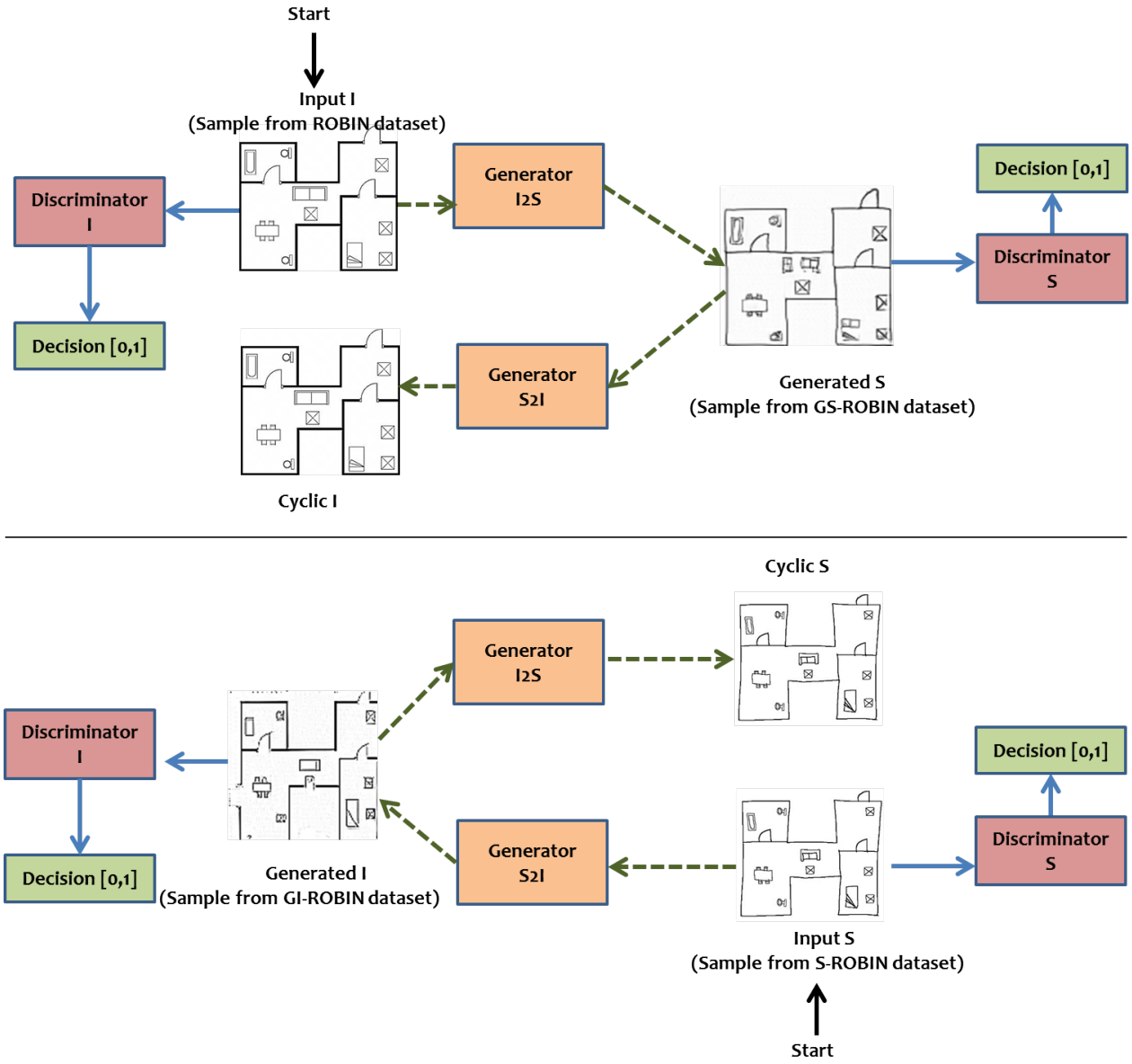

Figure 6.5. : Network architecture for the Cyclic GAN framework, representing the inputs from ROBIN and S-ROBIN dataset. The outputs being generated constitute the GS-ROBIN dataset.

In Eq. 6.1, two operations are taking place. Here the discriminator term D_I, tries to maximize the objective function by distinguishing between the generated floor plan images and the given sketches. Whereas, the generator function $G(.)$ tries to minimize the objective and synthesize image samples as close as possible to the sketch. Similarly, the adversarial loss function for the reverse mapping process, i.e. image to sketch $I2S_A(.)$ is formulated as:

$$I2S_A(F, D_S, I, S) = E_{f_S \to p_{(f_S)}}[logD_S(f_S)] + E_{f_I \to p_{(f_I)}}[log(1 - D_S(F(f_I)))] \tag{6.2}$$

As shown in Fig. 6.4, the generated target domain image using $G(.)$, and $F(.)$, is close to that of the source domain. The learned mapping functions are able to map to any random permutation

of images. However, it doesn't ensure that for a given floor plan image f_I^i, a corresponding f_S^i is obtained. In other words, given an image sample $f_s \in S$, if the generator function $G(.)$ is able to generate f_I^G (here the sub-script I stands for the image domain, and the super-script G denotes that it is a generated image), then the mapping function $F(.)$ must be able to take f_I^G and bring back f_S. So the sketch to image generation cycle completes when the proposed network is able to re-generate the input samples, from the synthesized (or GAN generated) samples, and thus termed as "cycle-consistent". Similar consistency is also desired for image to sketch domain translation. The cycle consistency loss ($CCL(.)$) function between the two mapping functions is formulated as:

$$CCL(G,F) = E_{f_S \to p(f_S)}[||F(G(f_S)) - f_S||_1] + E_{f_I \to p(f_I)}[||G(F(f_I)) - f_I||_1] \tag{6.3}$$

The final objective function ($X(.)$), that combines the two adversarial loss functions and the cycle consistency loss function is given by :

$$X(G,F,D_I,D_S) = S2I_A(G,D_I,S,I) + I2S_A(F,D_S,I,S) + \alpha CCL(G,F) \tag{6.4}$$

Here the hyper-parameter α controls the relative significance of the pair of loss functions, i.e. the adversarial loss and cyclic loss. Using this objective function $X(.)$, the optimum value for the two mapping functions $G(.)$ and $F(.)$ (by minimizing over G and F, while maximizing D_I and D_S) is obtained. The value of α was empirically determined to be 10, which yields the best retrieval accuracy for all the experiments performed for this problem. Another diagram to depict the Cyclic GAN framework is shown in Fig. 6.5.

Figure 6.6 depicts the schematic structure of the generator and discriminator network. There are two parts to the network structure, the generator network and the discriminator network. There are three components to the generator network, namely, an encoder, a transformer, and a decoder. The encoder is a convolutional neural network that extracts the features from a given floor plan dataset. The second building block is a transformer network. This layer has been constructed using 6 ResNet blocks to transform from I domain to S domain (refer Fig. 6.4). In Fig. 6.6, the internal structure of an individual ResNet block is shown. Each block is a neural network layer

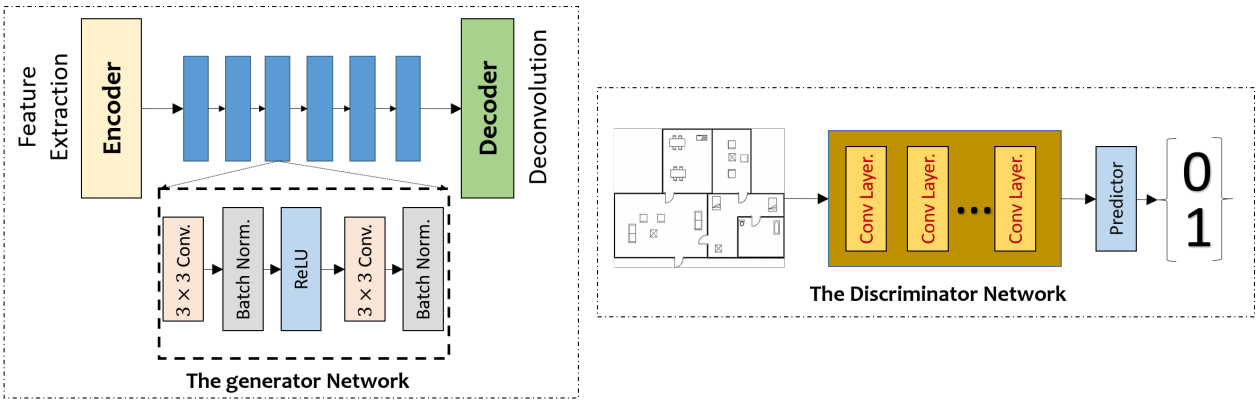

Figure 6.6. : Schematic structure of the Cyclic GAN network used in the proposed framework for image floor plan to synthesized sketched floor plan mapping, and vice-versa.

which consists of two convolution layers where a residue of input is added to the output. This is done to ensure properties of previous layer inputs are available for later layers as well, so that their output does not deviate much from original input resulting into abrupt results. The third component is a decoder network. The decoder block recreates the low level features from the feature vector by applying a deconvolution layer. The next part of the network is a discriminator network. The input to the discriminator network is an image. The output of the discriminator network is a prediction value 0 or 1, that denotes whether the generated image is original or the output from the generator. The discriminator is implemented as a series of convolutional neural networks. After this training and testing phase, a set of generated images corresponding to the sketches, as well as, a set of sketches corresponding to the ROBIN dataset are obtained. Thus, GI-ROBIN dataset generated from the S-ROBIN dataset, and GS-ROBIN dataset generated from the ROBIN dataset, are obtained which are further used for retrieval purposes. To improve the quality of the generated images and sketches, they are sharpened as a post-processing step.

After this training and testing phase, a set of generated images corresponding to the sketches, as well as, a set of sketches corresponding to the ROBIN dataset are obtained. It was observed that if the obvious path, of converting the query sketches into corresponding image counterparts and then retrieving from the ROBIN dataset was taken, the retrieval performance was quite weak giving an average precision value as low as 0.1631 and 0.1219, upon being trained on ROBIN as well as S-ROBIN dataset respectively (refer Fig. 6.14. (a)). Therefore, the less intuitive path of converting the ROBIN dataset into sketches and then performing retrieval was taken to yield better results.

6.2.2 Feature Representation through CNN

CNNs have proved to be very effective for feature extraction and classification purposes. In the recent past, research gaining insight into CNN models includes exploring new non-linear activation functions, new training techniques, optimal network configurations and making them system time efficient and reducing overfitting.

In this work as proposed in Chapter 4, a combination of convolutional, pooling, normalization and fully connected layers, similar to described in the work AlexNet [Krizhevsky *et al.*, 2012a], is taken to obtain an effective feature representation from floor plans. Changes are made in the model as compared to AlexNet by not using relighting data augmentation during training; and by changing the order of pooling and normalization layers [Jia *et al.*, 2014]. For learning a representation of CNN (deep) layers, training of the deep model was done on ROBIN, S-ROBIN and GS-ROBIN dataset. It was observed that the deep learning framework trained on GS-ROBIN dataset performs the best during retrieval (see Fig. 6.14. (b)). The effectiveness of each layer was also investigated by taking the activations of all the layers (convolution, pooling, norm, and fully connected). The major motivation behind using lower convolution neural network layers is to understand the effectiveness of low-level features as compared to higher-level feature representation for floor plan retrieval tasks. Since one only needs to compute the feed-forward network based on the matrix multiplication for one time, the whole scheme is highly time efficient. Therefore, this is how the features from both the sketched query samples as well as the ROBIN generated sketches (GS-ROBIN dataset) are generated. In matching and retrieval stage, query samples are matched with database samples to effectively retrieve the rank-ordered samples from the database.

6.2.3 Matching and Retrieval

As the retrieval is performed on the generated sketches from the ROBIN dataset, therefore, let $(x_\phi^{GS}, y_\phi^{GS})$, $\phi = 1,2...,m$, represent the generated sketches, where, m is the total number of samples in the GS-ROBIN dataset, x_ϕ^{GS} is the observed variable and y_ϕ^{GS} is the corresponding class label in the retrieval database. Let $f(x_\phi^{GS}) : \{f_1(x_\phi^{GS}), f_2(x_\phi^{GS}), .., f_H(x_\phi^{GS})\}$ be the set of hidden layer features extracted from x_ϕ^{GS} GS-ROBIN dataset sample, where $f_p(x_\phi^{GS})$ represents p^{th} hidden layer feature extracted from x_ϕ^{GS} sample in database and H represents the total number of hidden layers in deep feature hierarchy. For experimentation purpose, the features have been extracted from 12 hidden layers of CNN model. For each sample in the database, features from each hidden layer $(f_p(x_\phi^{GS}))$ are extracted and stored in the feature database as discussed in Sec. 6.2.2. In a similar manner, features are extracted from the query sketches as well. Firstly, the performance of individual hidden layer for proposed floor plan retrieval task is observed. Let $f_s(q) : \{f_{s1}(q), f_{s2}(q), .., f_{sH}(q)\}$ be the the set of hidden layer features extracted from a query sketch (q). Then, the Matching Score (M) between query sketch (q) and a sample from the database (x_ϕ^{GS}) using p^{th} hidden layer feature is calculated (and used for ranking), as:

$$M_{approach_1} = \sum_{i=1}^{|f_p(x_\phi^{GS})|} |f_p^i(x_\phi^{GS}) - f_{sp}^i(q)|_2 \tag{6.5}$$

Secondly, the best hidden layer representation for each retrieval sample is obtained by combining the matching score outputs of each hidden layer using a *min* operation. The Matching Score (M) between a query sketch (q) and a generated sketch from the ROBIN dataset (x_ϕ^{GS}) using all the hidden layer features is calculated (and used for ranking), as:

$$M_{approach_1} = argmin_p \sum_{i=1}^{|f_p(x_\phi^{GS})|} |f_p^i(x_\phi^{GS}) - f_{sp}^i(q)|_2, \quad p \in (1,...,H) \tag{6.6}$$

Using Eq. 6.5 and 6.6, given a query sketch the similar generated sketches from the GS-ROBIN dataset are obtained. It is to be noted, that as these sketches have been generated from ROBIN dataset, therefore, transitively they correspond to the images they are generated from. Thus, through this indirect relation, it is shown that the images belonging to the ROBIN dataset [Sharma *et al.*, 2017] are obtained as the retrieved result given the query sketch.

In the next section, Approach 2, catering to sketch based retrieval is discussed in detail. This approach additionally proposes a unified framework for dealing with both image and sketch based queries and finally performaing efficient matching and retrieval of similar floor plans given the dual query mode.

6.3 APPROACH 2: UNIFIED FRAMEWORK FOR RETRIEVAL USING AUTOENCODER AND CYCLIC GAN

Figure 6.7 depicts the complete framework of the proposed technique. Firstly, a query is classified as a floor plan sketch or image using a Convolutional Neural Network (CNN). If

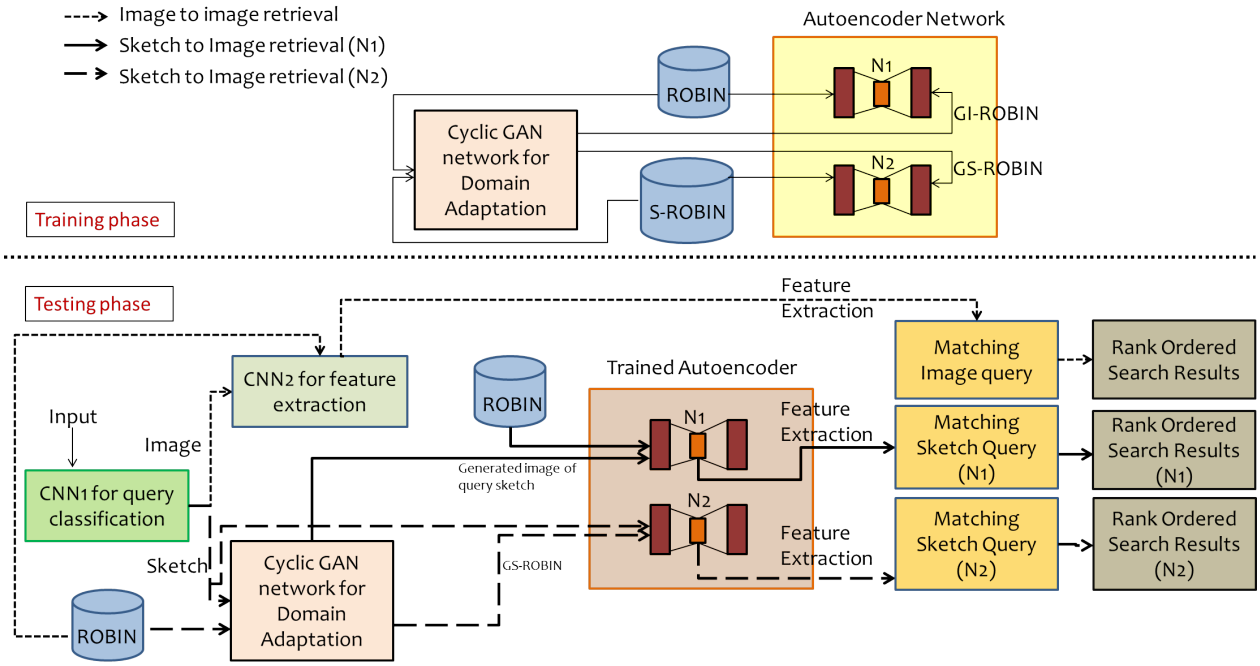

Figure 6.7. : Unified framework for multimodal retrieval.

the query is a floor plan image, the query and the database images are passed through a CNN for feature extraction, and retrieval is carried out using a distance metric to define similarity as proposed in Chapter 4. However, if the query is a sketch, the steps are taken as follows. The entire framework is divided into three main phases. They are: (1) cross-domain sample generation through Cyclic Generative Adversarial Networks (Cyclic GANs); (2) domain mapping through autoencoder and Cyclic GANs, and (3) matching and retrieval. Given an input sketch S, the goal is to retrieve similar looking floor plan images (I) from the ROBIN dataset [Sharma *et al.*, 2017] corresponding to the query sketch. The Cyclic GAN is trained using images from ROBIN and sketches from S-ROBIN to learn domain representation. Direct retrieval of floor plan images, given a sketch, is not straightforward (refer to the low performance of DANIEL in Fig. 6.18) proposition. Therefore, firstly the ROBIN dataset images are converted into their sketch counterparts (called GAN generated sketch dataset (GS-ROBIN)) and the S-ROBIN dataset into their image counterparts (called GAN generated image dataset (GI-ROBIN)) using a Cyclic GAN trained to map the sketch and image domain (as shown in Fig. 6.4).

Further, to map both the floor plan sketches/ retrieval set floor plan images and the cyclic GAN generated outputs, to a common space, experiments are performed by training two autoencoders, taking input as images from ROBIN (network N1) and sketches from S-ROBIN (network N2). The basic autoencoder network is tweaked a little, by keeping the target as GAN generated images (GI-ROBIN) in case of N1 and GAN generated sketches (GS-ROBIN) in the case of N2. This autoencoder learning helps in domain mapping of (a) Images from ROBIN and Generated Images from S-ROBIN (i.e., GI-ROBIN) using autoencoder N1, (b) Sketches from S-ROBIN and Generated Sketches from ROBIN (i.e., GS-ROBIN) using autoencoder N2. Further, the compressed features of both query sketch and floor plan images in the database are extracted from the last layer of the encoder of the trained autoencoder network and used for matching and retrieval purposes. In the next section, details about the various modules of the unified framework are discussed.

6.3.1 CNN for query classification

In this module of the framework a Convolutional Neural Network (CNN), trained on both the floor plan sketch and image datasets, ROBIN and S-ROBIN, is used to classify whether a query is a sketch or an image. A combination of convolutional, pooling, normalization, ReLU and fully connected layers (see Fig. 6.8) along with dropout regularization technique, similar to described in the work AlexNet [Krizhevsky *et al.*, 2012a], is used to obtain an effective feature representation from both floor plan images and sketches.

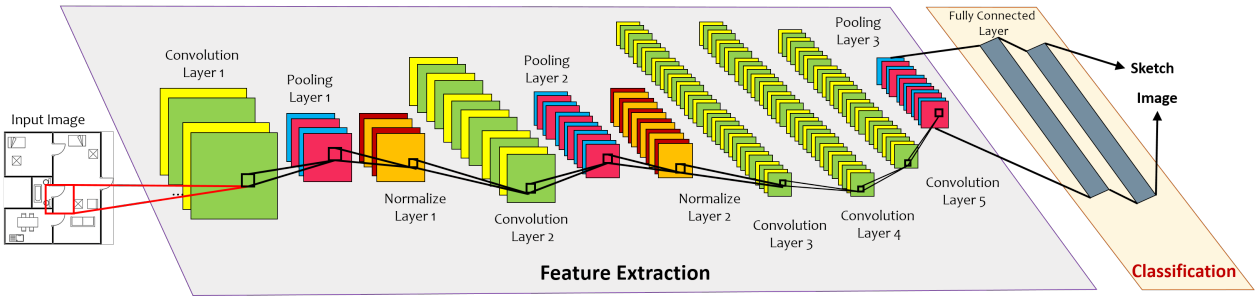

Figure 6.8. : Layerwise depiction of the Convolutional Neural Network (CNN) for query classification

As proposed in Chapter 4, to create a distinction from the original AlexNet framework, the model is not trained with the relighting data-augmentation; and pooling is done before normalization [Jia *et al.*, 2014]. To obtain the feature representation from both the floor plan sketches and images, they are fed into the input layer of the learned CNN model. The sketched floor plans are labeled as 1 and the floor plan images are labeled as 0. The model takes the activations of all the layers viz. a viz. convolution, pooling, normalization, and fully connected to extract the low-level features from the sketch and image floor plan datasets. Since the feed-forward network performs matrix multiplication only once, it makes the whole scheme highly time efficient. The datasets are divided into training and testing sets, which are comprised of 90% and 10% samples respectively and the network is appropriately trained. Finally, on giving the test set to the trained network, the probability of each sample from the mixed bag of S-ROBIN and ROBIN is evaluated to determine which class (0 or 1) they lie into and finally, each test sample is classified into a floor plan image or a sketch. If it is a floor plan image, retrieval proceeds as in [Sharma *et al.*, 2017]. However, if it is a floor plan sketch then domain adaptation as proposed in the subsequent subsections is done.

The next task is cross-domain sample generation using Cyclic GANs, which is done in a similar manner as discussed in Sec. 6.2.1. After the training and testing phase of the Cyclic GAN, a set of generated images corresponding to the sketches, as well as, a set of sketches corresponding to the ROBIN dataset is obtained. Thus, after this step, two sets, GI-ROBIN dataset generated from the S-ROBIN-dataset, and GS-ROBIN dataset generated from the ROBIN dataset, are obtained which are used for retrieval purposes.

6.3.2 Autoencoder for domain mapping

If the query is classified as a sketch, it is first converted into its image representation using Cyclic GAN as discussed in Sec. 6.2.1. As observed in Fig. 6.3, function $G(S)$ generates an image corresponding to the floor plan sketch S that looks approximately similar to how a corresponding image would be in the ROBIN dataset. However, the approximation is still a little far from the exact actual image representation (Fig. 6.3 (b)). Therefore, if direct matching between a generated image from the query sketch and samples of the ROBIN dataset (retrieval set) is performed, the average

precision values obtained were as low as 0.2, which although, is still better than matching sketches and images directly (average precision in this case was obtained to be 0.04) but requires significant improvement. Therefore, to overcome these shortcomings, the introduction of autoencoders is proposed. This helps in mapping the sketch query and sketches generated from the floor plan images, or in another case the image generated from query sketch and the retrieval set floor plan images onto a common space for matching purposes. An autoencoder neural network is an unsupervised learning algorithm that applies backpropagation and sets the target values to be equal to the inputs, i.e., it uses $y^i = x^i$. The autoencoders here are not being used to generate output image similar to the input image at the output layer, rather the use of autoencoders is proposed to learn a common subspace representation of two similar looking domains. Therefore, in the proposed case, additional output image (generated using Cyclic GAN) corresponding to the input image is used for training the autoencoders (Fig. 6.9). This additional output image helps to learn intermediate representation of both input and output domains. This intermediate representation acts as a common subspace representation in our case. The novelty in this case lies in the fact that, unlike traditional autoencoders, which try to generate an output similar to the input fed to them by computing the loss between the input and the target output and updating the weights, the training process of the proposed autoencoder is slightly modified. In the proposed case, a new way of learning representation is performed, where, predefined outputs are used along with the decoder generated output for autoencoder training. The aim is to minimize the difference between the decoder generated output and provided output (generated through Cyclic GAN). The autoencoder is trained in two ways, firstly, the autoencoder labeled as N1 is trained keeping the input as images from ROBIN dataset (i $\in$ ROBIN), whereas, the additional target output as images generated from sketches through the Cyclic GAN network (i' $\in$ GI-ROBIN). This helps in learning the intermediate representation till the loss between the decoder generated output and the sketch generated image domain converges to a small value. For this task, an autoencoder is used as shown in Fig. 6.9.

An autoencoder consists of two parts, the encoder and the decoder, which can be represented as $\mathscr{E}$ and $\mathscr{D}$ respectively. Where, $\mathscr{E} : i \rightarrow z$ and $\mathscr{D} : z \rightarrow \hat{i}$, and z is the compressed representation of the input. Note that during training the target has been set as the images generated form sketches of the S-ROBIN dataset, so that the autoencoder learns to map both domains.

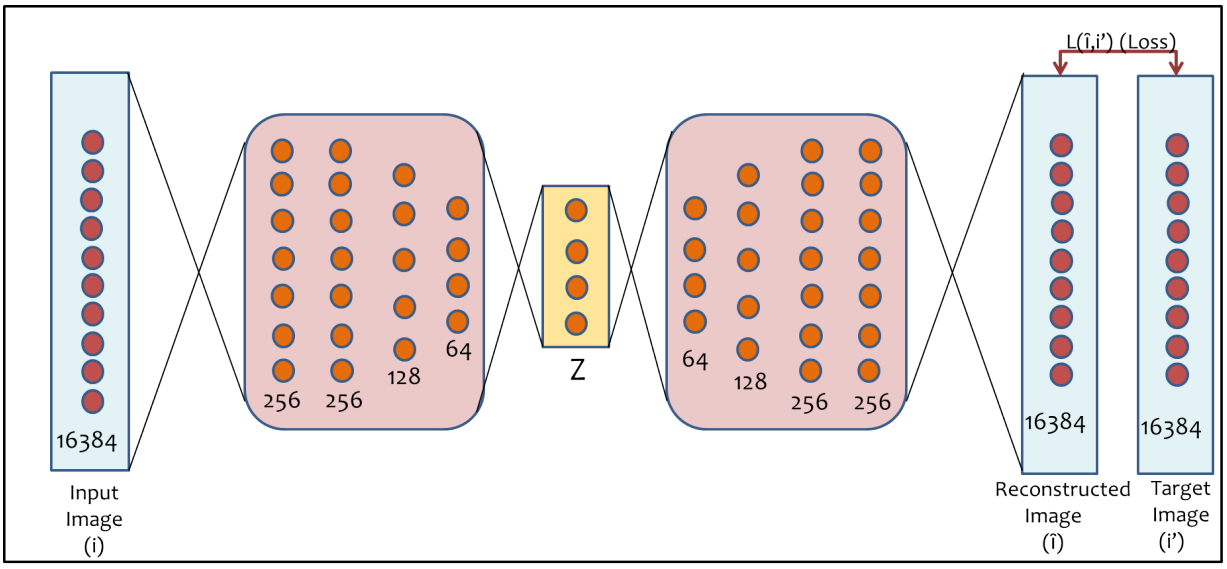

Figure 6.9. : Layerwise depiction of the autoencoder network.

Further, the squared loss function to minimize the reconstruction error to train the network is defined as follows:

$$\mathscr{L}(\hat{i}, i') = ||\hat{i} - i'||^2 \tag{6.7}$$

Note that, $\hat{i}$ is the reconstructed image from the trained autoencoder corresponding to the input. The distinction of our autoencoder network from a simple autoencoder is that the loss is calculated between the GAN generated floor plan image and the reconstructed image. This helps in efficient mapping of both sketch and image domains by taking aid from the Cyclic GAN network, for conversion of sketch query to image or retrieval set images to sketch. The parameters for the autoencoder network are listed in Tab. 6.3.

After the network is fully trained, features from the last layer of the encoder are extracted for both the query sketch generated image that has been obtained using a Cyclic GAN and images from ROBIN dataset that serve as the retrieval set. This helps to obtain a compressed feature vector for both, which is further used for matching and retrieval. This feature extraction helps in mapping both domain of images (I) and domain of sketches (S) to a common domain called (C) where, now the matching and retrieval process takes place.

On similar lines, the autoencoder network, labeled N2, was also trained keeping the input as sketches from the S-ROBIN dataset and the target as generated sketches (GS-ROBIN dataset) obtained by the Cyclic GAN network. It was interesting to note, that due to the quality of the generated sketches from the Cyclic GAN being better, the autoencoder learned a better representation and mapping of both the domains using the N2 network. As per our knowledge, this is a novel way to learn a common embedding space between the sketch and image domain, using a modification in the autoencoder framework by not keeping the input and the expected output to be the same.

6.3.3 Matching and Retrieval

During testing, if the query is a floor plan image then matching and retrieval proceeds as proposed in Sec. 4.3. However, if the query encountered in the framework is classified as a sketch, then two paths can be taken for retrieval purposes (solid-line and dashed line) as shown in Fig. 6.7. The query sketch can be converted into its image representation using the Cyclic GAN. Compressed vector can be obtained from the final layer of the encoder, by passing query sketch generated image and the ROBIN dataset through the autoencoder network N1. The features thus obtained can be used for matching and retrieval purposes. On similar lines, the other path that can be adopted is if the query is a sketch, then the query sketch, along with the sketch counterparts of the ROBIN dataset (GS-ROBIN dataset) can be passed through N2 autoencoder network, where again, compressed vector obtained from the last layer of the trained encoder can be used as a feature vector for both query sketch and the retrieval set. To illustrate the matching process for one such path, let $f(x_i) : \{f_1(x_i), f_2(x_i), .., f_n(x_i)\}$ be the feature vector extracted for x_i image where $i \in$ ROBIN dataset, and n represents the size of this feature vector, which in our case is 64. For each sample in the database, features are extracted and stored in the feature database. In a similar manner, features are extracted from the query sketch generated image. Then, the Matching Score ($M_{Approach_2}$) between query sketch generated image(gi) and a sample from the database (x_i) is calculated (and used for ranking), using three different distance metrics:

Euclidean Distance Metric

$$M_{Approach_2} = \{ \sum_{a=1}^{|f(x_i)|} \{f_a(gi) - f_a(x_i)\}^2 \}^{\frac{1}{2}} \tag{6.8}$$

Manhattan Distance

$$M_{Approach_2} = \sum_{a=1}^{|f(x_i)|} |f_a(gi) - f_a(x_i)|_2 \tag{6.9}$$

Cosine Distance

$$M_{Approach_2} = \sum_{a=1}^{|f(x_i)|} \frac{\sum_{a=1}^{|f(x_i)|} \{f_a(gi).f_a(x_i)\}}{\{\sum_{a=1}^{|f(x_i)|} \{f_a(gi)\}^2\}^{\frac{1}{2}}.\{\sum_{a=1}^{|f(x_i)|} \{f_a(x_i)\}^2\}^{\frac{1}{2}}} \tag{6.10}$$

The distance is calculated using 10-fold cross-validation where, the query sketch samples consist of 70% of the sketched floor plans from S-ROBIN dataset and the retrieval set comprises of all the 510 floor plans in the ROBIN dataset. Based on this distance the database floor plan images are rank ordered. Shorter the distance, more similar a pair of floor plan is. The rank ordered samples are shown to the user. Next, the details of the experiments performed, implementation details, and the results obtained from them are discussed.

6.4 EXPERIMENTS AND RESULTS

6.4.1 Dataset Creation

The experimentation was done on 2 datasets : (1) ROBIN dataset[Sharma *et al.*, 2017] and (2) S-ROBIN dataset. For the framework all the floor plan images of ROBIN to sketches were replicated. Since, there is no publicly available sketch floor plan dataset for the DAR community, a new dataset was created and named as S-ROBIN as discussed in Annexure A. S-ROBIN dataset has 510 real world floor plan sketches divided into 3 broad categories and 17 sub-categories differing in the overall global outer shape of the layouts. It is to be noted that class-belongingness of each sketched layout is determined in the dataset by its outer shape, for example all the hand-drawn floor plans in one sub-category will have the same L-shaped outer layout. To capture these hand drawn sketched floor plans, the volunteers were asked to draw the floor plans on a digital platform using a Wacom Tablet. The resolution ranges from 591×517 pixels to 1483×884. Since working in the off-line mode is considered, the hand drawn floor plans are stored as an image and used for experimentation. All the images were saved in a gray scale format.

6.4.2 Implementation Details

For training the Cyclic GAN, the datasets ROBIN and S-ROBIN were divided into training (70%) and testing (30%) sets. Further, the Cyclic GAN with the help of the losses discussed in Sec. 6.2.1, learns the mapping between both domains. During testing, due to the cyclic nature of the network, both the generated sketches from [Sharma *et al.*, 2017] and the corresponding images generated from the S-ROBIN dataset are obtained. It was observed that visually the sketches generated from ROBIN were of better quality. To make it clear to the reader, let query sketch

be S, the ROBIN dataset images to be retrieved be $\{I\}$ and the generated sketches from ROBIN ($\{I\}$) be $\{GS\}$. Then as S and $\{GS\}$ are in the same domain, similar sketches in $\{GS\}$ are retrieved smoothly through the framework. Further, it is to be noted, these generated sketches $\{GS\}$ exactly correspond to their image counterparts from $\{I\}$. Hence, it is safe to say that through the framework relationship between S and $\{GS\}$ has been determined. Also, relationship between $\{GS\}$ and $\{I\}$ pre-exists. Therefore, transitively, relationship between S and $\{I\}$ has thus been established. It is this indirect relationship that helps to justify that ROBIN dataset images similar to the query sketch can be shown as the retrieval result obtained through the framework.

Approach 1:

For feature representation in Approach 1, three CNNs were trained using the ROBIN dataset, S-ROBIN dataset and the GS-ROBIN dataset, respectively. All the three datasets are again divided into training set with 70% samples and test set with 30% samples. As a retrieval database, the entire GS-ROBIN dataset is used which is generated from the ROBIN dataset and thus, transitively represents it. Each floor plan image has been resized to a size of 128×128 for both matching and feature representation task. For deep model configuration, the architecture described in [Krizhevsky *et al.*, 2012a] is followed.

Approach 2:

After the classification task, if the query is a floor plan image, feature extraction and retrieval proceeds as in [Sharma *et al.*, 2017]. If the query is a sketched floor plan then the Cyclic GAN network comes into picture to convert the sketched floor plans into their image counterparts, which are then used for training the autoencoder, to bridge the difference between the sketched and the image domain.

6.4.3 Parameter Setting

Convolutional Neural Network:

The parameters for the CNN network for both feature extraction in Approach 1 and query classification in Approach 2, are similar as mentioned in Chapter 4. Table 6.1 lists out these parameters again.

Cyclic GAN:

Setting appropriate values for the hyper parameter plays a pivotal role for the success of the Cyclic GAN and the deep learning implementation. Table 6.2 lists out the parameters and their corresponding values used while implementing the framework. The effect of higher number of iterations to generate good quality image is depicted in Fig. 6.10. It is to be noted that the input is given in the form of an image and number of filters in the 1^{st} layer of the generator determine the output features of the convolution layer. This is then passed to the next convolution layer, which subsequently leads to extraction of progressively higher level features aiding in domain mapping. It can be observed that given an original floor plan image our proposed network is able to generate a sketched version of it with very high accuracy. As the number of iterations increases the quality improves. It was empirically found that after the optimum value of number of iterations is achieved, the quality of the image doesn't improve significantly, and hence to have a trade-off between time and quality, an appropriate value for the number of iterations is chosen, which in our case was chosen to be 60000 iterations.

Autoencoder framework:

Layers	1	2	3	4	5	6
Type	conv1+ maxpool1+ norm1	conv2+ maxpool2+ norm2	conv3	conv4	conv5+ maxpool3	full (fc6,fc7)
Channels	96	256	384	384	256	4096
Filter Size	11*11	5*5	3*3	3*3	3*3	-
Convolution Stride	4*4	1*1	1*1	1*1	1*1	-
Pooling Size	3*3	3*3	-	-	3*3	-
Pooling Stride	2*2	2*2	-	-	2*2	-
Padding Size	-	1*1	1*1	1*1	1*1	-

Table 6.1. : Network Parameters for the proposed framework

Table 6.2. : Parameter Values used while implementing Cyclic GAN

Parameter	**Value**
# filters in the 1^{st} layer of the generator	64
# filters in the 1^{st} layer of the discriminator	64
batch size	1
pool size	50
img width and img height	128

The autoencoder used in Approach 2, plays a pivotal part in mapping both the sketched and image floor plans onto a common space for efficient retrieval. The images/sketches are reshaped to a size 128×128 and vectorized into a 16384×1 vector, before being fed to the autoencoder network.

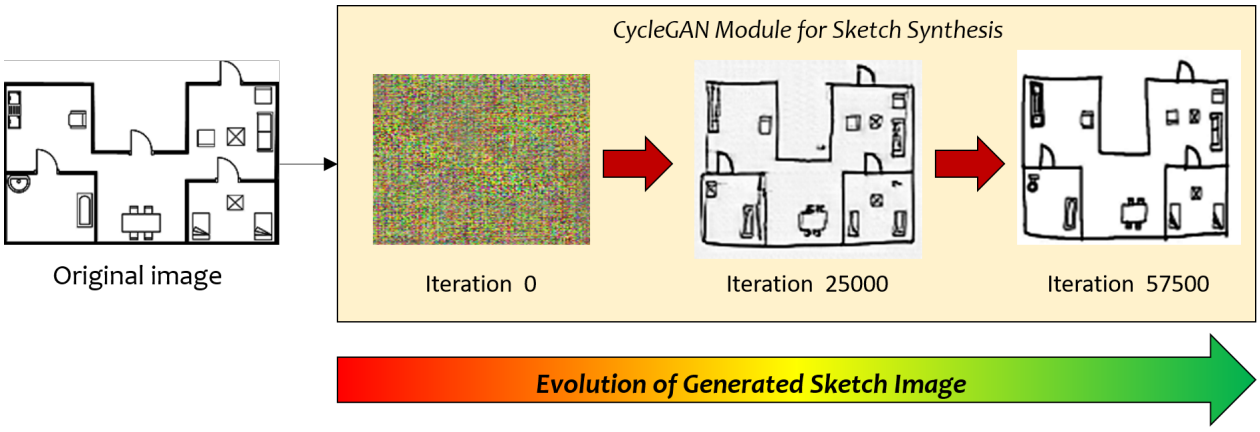

Figure 6.10. : A diagram depicting stages of sketch synthesis from an image through our Cyclic GAN framework. Note how, the discriminator and generator improve with increasing the number of training iterations.

100,000 iterations were undertaken for the autoencoder to train properly and 70% of the samples from the ROBIN and S-ROBIN datasets were used as training set for training both CNN, Cyclic GAN and the autoencoder. Further, the details of the parameters involved in the autoencoder network are listed in Tab. 6.3.

Table 6.3. : Parameter Values for Encoder Network

Component	Parameter	Value
Encoder	No. filters in the 1^{st} layer	256
	No. filters in the 2^{nd} layer	256
	No. filters in the 3^{rd} layer	128
	No. filters in the 4^{th} layer	64
Decoder	No. filters in the 1^{st} layer	64
	No. filters in the 2^{nd} layer	128
	No. filters in the 3^{rd} layer	256
	No. filters in the 4^{th} layer	256

6.4.4 Qualitative Results

Approach 1:

Approach 1 is able to successfully retrieve a rank order set of floor plan images from the ROBIN dataset given a query in the form of sketch, using the proposed domain mapping framework consisting of Cyclic GAN and CNN. The top 5 rank ordered retrieval results for three different hand-drawn queries, are shown in Fig. 6.11 (a),(b) and (c). The samples which are erroneously retrieved, i.e. not from the same class as that of the query class, are highlighted with a "red" bounding box. For all the results shown in Fig. 6.11, the features extracted from the first normalization layer of the CNN stack have been used, owing to the fact that it yields the best performance. The first rank ordered result retrieved from the ROBIN dataset, is the image exactly matching the query sketch. This is due to maximum matching score between the retrieved layout and the query layout. The subsequent results differ in the matching scores and are thus, ranked lower. In case of Fig. 6.11 (a) and (b), all the top 5 retrieved samples belong to the query class. Global layouts of all the floor plans are identical and so is the number of rooms in the floor plans. It is to be noted that there is a difference in the shape and position of the individual rooms, as well as, the furniture components for the rank ordered samples. There are 4 rooms in each floor plan (refer Fig. 6.11 (a)), however this framework is able to rank order the samples correctly by ordering the most similar ones in terms of the features present inside the floor plans.

Figure 6.11(c) depicts one of the cases where this algorithm does not perform perfectly. As it can be noted, for a given query sketch, the 5^{th} ranked result is erroneously retrieved (highlighted by "red") by this technique. Justification of the Rank 5 erroneously retrieved result can be understood through qualitative analysis. Observe that the rooms in both the layouts share very similar adjacency. The bottom part of the floor plan is almost identical. Moreover, the size and type of furniture present in the individual rooms is quite similar in case of the query and the Rank 5 sample. Hence, similarity score between the query floor plan and the retrieved Rank 5 layout is high.

Approach 2:

The framework proposed in this approach can successfully retrieve a rank order set of floor plan images from the ROBIN dataset given a query in the form of floor plan sketch/image. It is

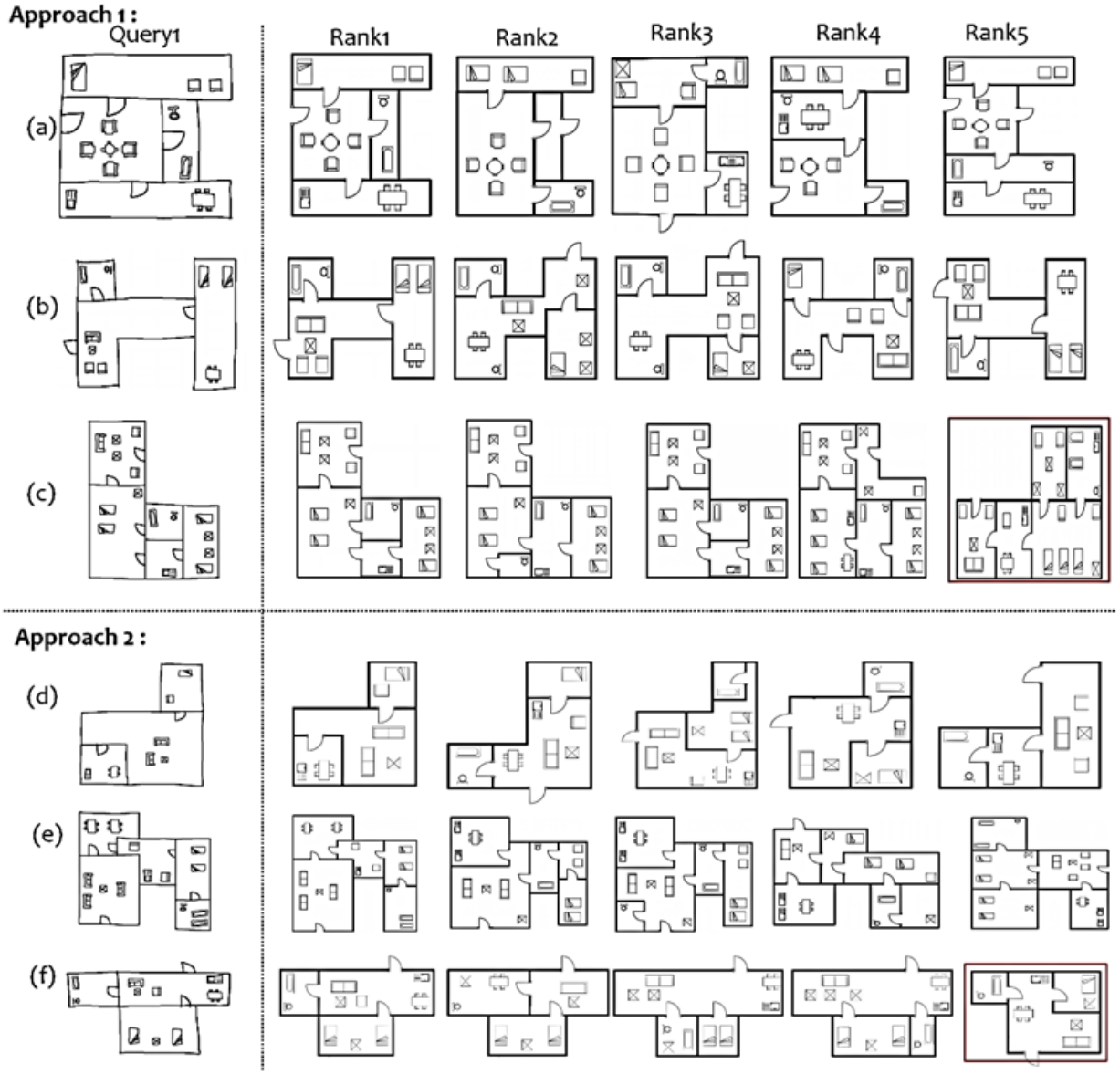

Figure 6.11. : Top five rank ordered retrieval results for the two proposed approaches for three different hand-drawn query floor plans.

to be noted that the retrieval set is always the ROBIN dataset. Experimentation was done taking random queries from both S-ROBIN and ROBIN dataset. The top 5 rank ordered retrieval results for three different hand-drawn queries are shown in Fig. 6.11 (d), (e) and (f). The samples which are erroneously retrieved, i.e., not from the same class as that of the query class, are highlighted by a bounding box. As shown in Fig. 6.11, it can be observed that the conjunction of Cyclic GAN and autoencoder for mapping as proposed in the framework, is efficiently able to capture the similarity between the query sketch and the floor plan images. In the case of Fig. 6.11 (d) and (e), all the top 5 retrieved samples belong to the query class. Global layouts of the all the floor plans are identical

and so is the number of rooms in the floor plans. It is to be noted that there is a difference in the shape and position of the individual rooms, as well as, the furniture components for the rank ordered samples. There are three rooms in each floor plan (refer Fig. 6.11 (d)). However, this framework can rank order the samples correctly by ordering the most similar ones in terms of the features present inside the floor plans.

Figure 6.11(f) depicts one of the cases where the algorithm does not perform perfectly. As it can be noted, for a given query sketch, the 5^{th} ranked result is erroneously retrieved (highlighted by "red") by this technique. Justification of the Rank 5 erroneously retrieved result can be understood through qualitative analysis. Observe that the rooms in both the layouts share very similar adjacency. The similarity lies in specific parts such as the 1^{st}, 2^{nd}, and 3^{rd} rooms have the same furniture components as the query floor plan, although the overall layout differs.

Similarly, qualitative analysis of the framework for the query being an image was also done as shown in Fig. 6.12. For all the results shown in Fig. 6.12, the features extracted from the first normalization layer of the CNN stack are used, as it yields the best performance. The first rank ordered result retrieved from the ROBIN dataset, is the image exactly matching the query image. The reason is the maximum matching score between the retrieved layout and the query layout. The subsequent results differ in the matching scores and are thus, ranked lower.

Both the proposed frameworks are also able to perform well for queries belonging to unseen classes. Figure 6.13 (1) and (2) depict two such scenarios where the sketch query does not match any of the samples in the ROBIN dataset. Still, this framework is capable of retrieving the closest

Figure 6.12. : Top five rank ordered retrieval result of proposed framework for three different query floor plan images using Approach 2 proposed in the Chapter.

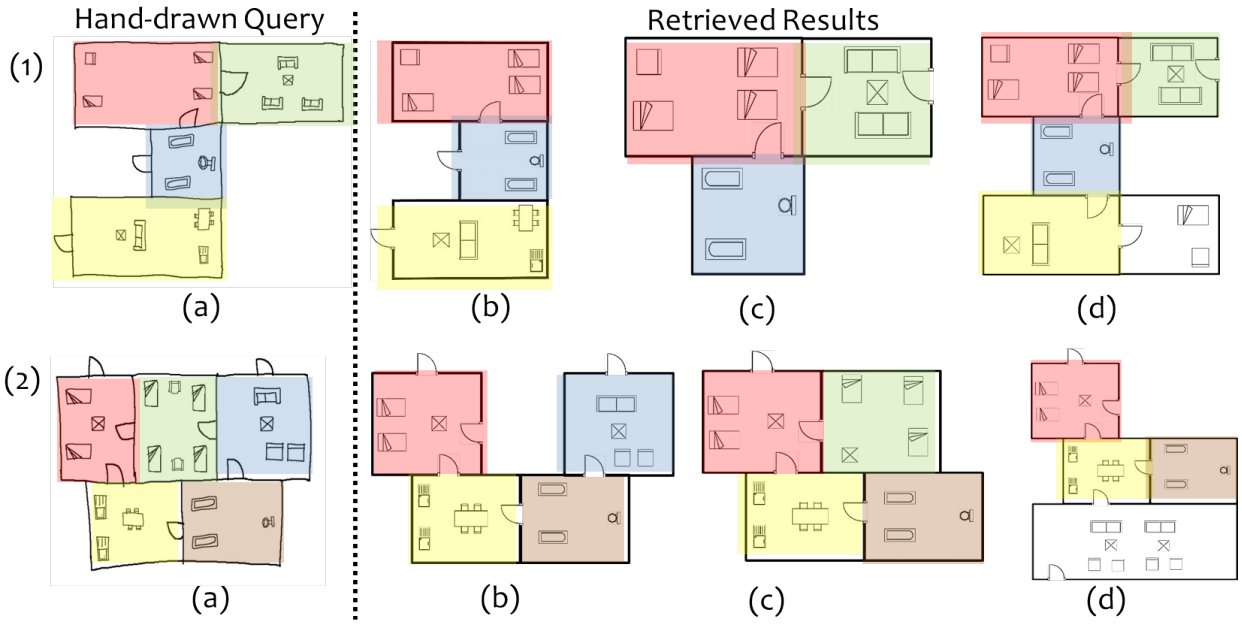

Figure 6.13. : Two examples showing results for querying the system for partial matches. (a) Query Sketch, (b),(c),(d) Retrieved results. Colour Coding establishes room correspondence between the query and the results.

similar results, in terms of room furniture and room placement. E.g., note the similarity between the rooms highlighted in "red", "blue" and "yellow" of the query sketch Fig. 6.13 (1) (a) and the retrieved result of Fig. 6.13 (1) (b). The decor components and room placement are quite similar. The "green" part of Fig. 6.13 (1) (a) is missing in Fig. 6.13 (1) (b). Similar observations can be made for the other results also. Although the shape of the layout is a subset of the shape of the query sketch and does not match completely, this framework is able to correctly retrieve and rank-order the results. Thus, this can help to suggest varied finished designs to architects/property buyers if they have some abstract ideas in mind.

6.4.5 Quantitative Results

The performance of both the approaches is quantified using the Precision (P) and Recall (R) metric. The precision values have been averaged over all the queries for the particular recall values. Given a query sketch, retrieved layouts should belong to the same sub-category of layouts as the query, keeping in mind the preference set by the property seeker during querying. Here, the global outer shape of the layout is the criteria for class belongingness of a given floor plan.

Approach 1:

As discussed, direct retrieval of floor plan images given a sketch is not straightforward (refer to the low performance of the CNN framework in Fig. 6.18 using DANIEL proposed in Chapter 4). Moreover, converting query sketches into corresponding image counterparts and then retrieving from the ROBIN dataset yielded weak performance (refer CNN A, B (trained on ROBIN and S-ROBIN respectively) in Fig.6.14 (a)). Therefore, sketch counterparts of ROBIN dataset generated through CyclicGAN are used for retrieval. The experimentation was carried out with CNNs trained on ROBIN (CNN_1), S-ROBIN (CNN_2) and the GS-ROBIN dataset (CNN_3), for feature

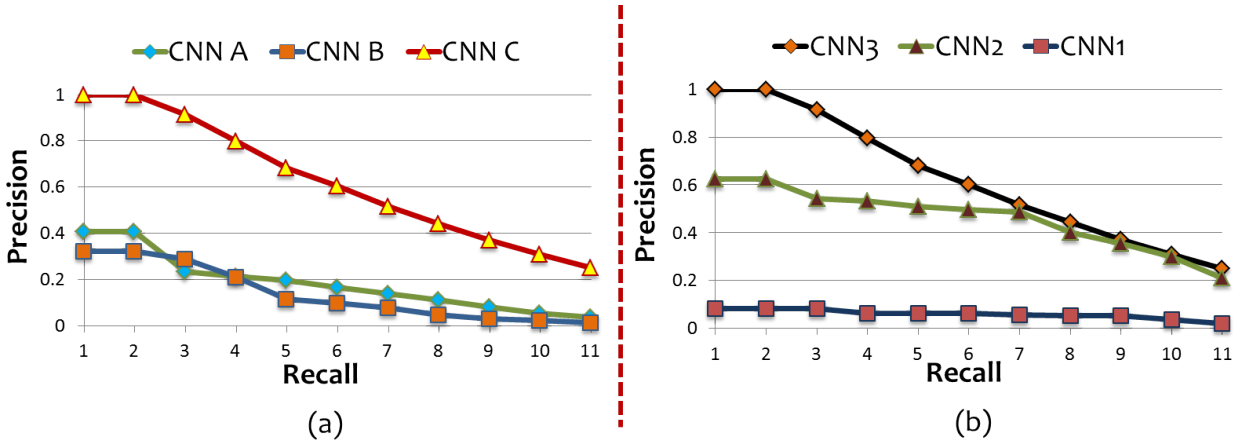

Figure 6.14. : (a) Effect of converting query sketch into image through Cyclic GAN and performing retrieval (CNN A (trained on ROBIN) and CNN B (trained on S-ROBIN)) as opposed to converting the retrieval database into sketches (CNN C). (b) Effect of training CNNs on ROBIN (CNN_1), S-ROBIN (CNN_2) and GS-ROBIN dataset (CNN_3)

extraction purpose, and interesting quantitative results were obtained showing that CNN trained on the GS-ROBIN dataset performs the best in terms of retrieval of similar floor plans (refer Fig.6.14 (b)). To understand the effectiveness of the proposed CNN layers, the PR and Mean Average Precision (MAP) of floor plan retrieval task was evaluated using each layer separately (as shown in Fig. 6.15). It was analyzed that normalized and pooling layers are quite powerful as compared to convolutional layer for floor plan retrieval task. It was observed that the normalized layer 1 yielded the best average precision value of 0.63 while retrieval. An analysis was also performed for the proposed min based approach (refer Eq. 6.6) during retrieval task and it was observed that the precision values through the min-based approach were obtained to be 0.525.

Approach 2:

Using the unified framework proposed in Approach 2, the mean average precision (MAP) value obtained if the query is an image is 0.56 as reported in Chapter 4.

However, if the query is a sketched floor plan, as discussed before, direct retrieval of floor plan images given a sketch is not straightforward (refer the low performance of CNN framework, DANIEL in Fig. 6.18). The experimentation was done with training the autoencoder network using sketches as well as images as described in Sec. 6.3.2. The PR plot obtained for retrieval using network N1 and N2 is shown in Fig. 6.16.

As shown in Fig. 6.11, given a query sketch the rank 1 result is the corresponding query floor plan image itself. This leads to the highest precision value of 1 during the initial recall. With further retrieval the average precision value decreases due to some incorrectly retrieved results belonging to other categories of layouts as compared to the query layout. It is to be noted that the area under the PR curve (refer Fig. 6.16) is greater for network N2, trained on sketch datasets due to the fact that the generated sketches from Cyclic GAN network are qualitatively better. The average precision value is obtained to be 0.642 using N2 network for domain mapping and 0.621 using the N1 network. Also, Tab. 6.4 lists out how each distance metric shown in Sec. 6.3.3 performs while matching and retrieval.

Figure 6.17, compares both the approaches and their analyzed sub-components. In Approach

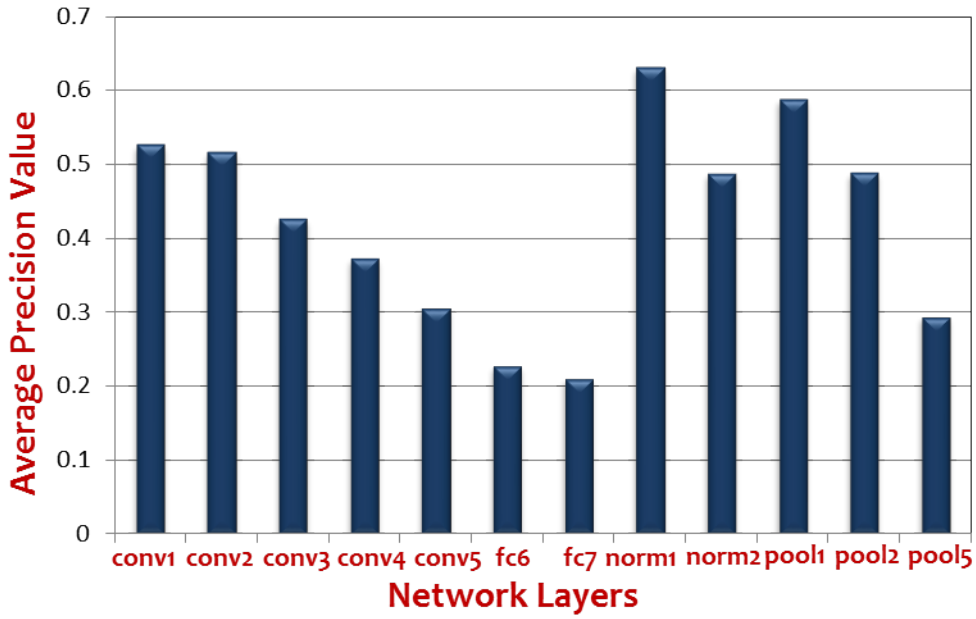

Figure 6.15. : Precision and Recall (PR) plot, comparing result of the framework, using all the hidden layers CNN features.

Table 6.4. : Mean Average Precision values for the proposed framework considering different distance metrics.

Distance Metric	Average Precision Value
Euclidean Distance	0.642
Cosine Distance	0.636
Manhattan Distance	0.625

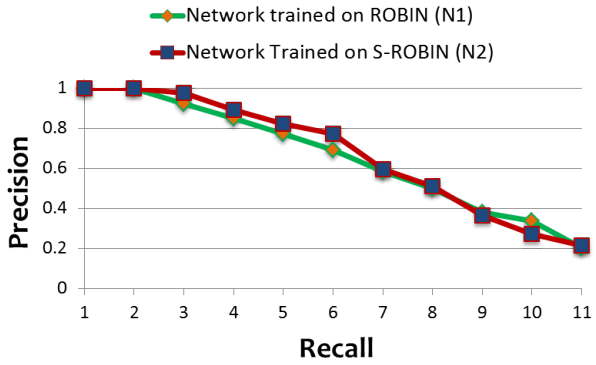

Figure 6.16. : Precision and Recall (PR) plot, comparing result of this framework on autoencoder trained with ROBIN dataset (N1) and S-ROBIN dataset (N2).

1, retrieval results are demonstrated using (1) the min-based approach taking the minimum of distances obtained by retrieval from all layers of the CNN framework and then calculating the Precision and Recall. (2) The best precision obtained from each layer of the CNN framework. This layer was obtained to be normalization one layer, which out-performed all other layers of the CNN framework in terms of retrieval. In Approach 2, N1 autoencoder is trained using images from ROBIN dataset as input and the target images as sketch generated images, (GI-ROBIN dataset) obtained through Cyclic GAN network. N2 autoencoder network, on the other hand, is trained using sketches from S-ROBIN dataset as input and the target images as the image generated sketches, GS-ROBIN dataset obtained through Cyclic GAN network.

It can be observed looking at the area under the curve in Fig. 6.17, that Approach 2, using N2 autoencoder network outperformed all the other approaches while retrieval and gave a MAP value of 0.642 during retrieval.

Comparative studies with state-of-the-art (SOA) techniques were carried out to show the effectiveness of the proposed approaches. Figure 6.18 depicts the PR plot comparing the result of best of the two approaches (Approach 1 using norm1 layer and Approach 2 using N2 network), with the other SOA techniques. There is a significant increase in the performance while using the two proposed approaches as compared to other features or techniques keeping sketch as query mode during retrieval.

Kindly note, the under-performance of the basic deep learning framework [Sharma *et al.*, 2017] (average precision value : 0.04393) using straight-forward feature extraction from CNNs.

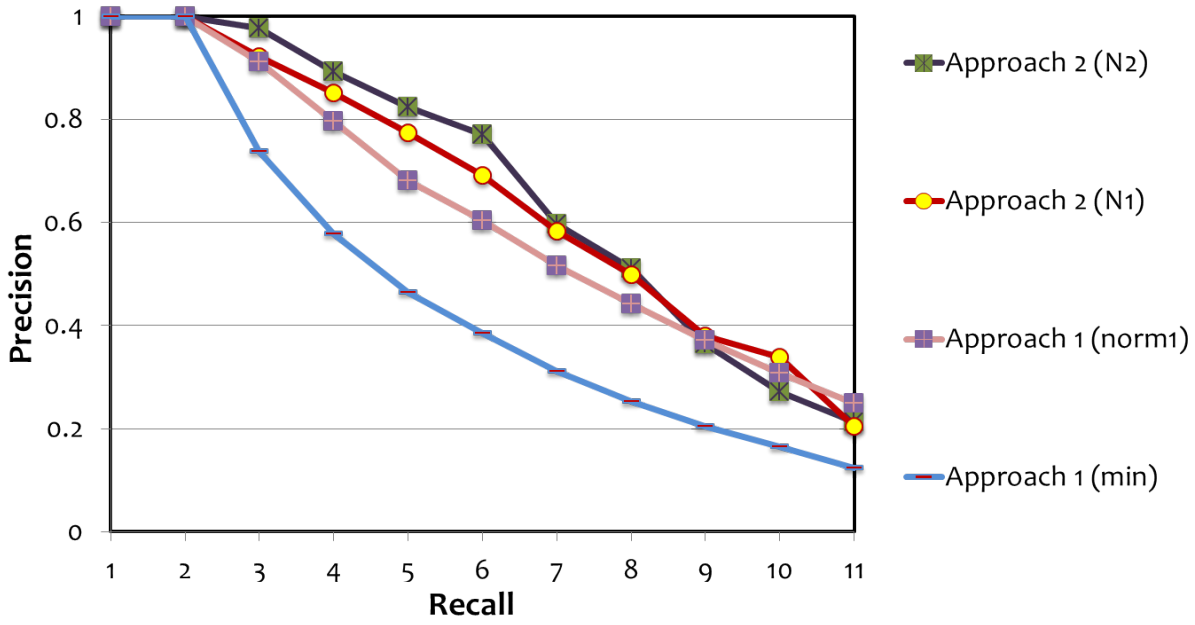

Figure 6.17. : PR plot, comparing the two approaches. Approach 1 extracts features and performs retrieval taking minimum of the precision values obtained by each layer (Approach 1 (min)) and taking the best layer (normalization 1 layer) giving the highest MAP value (Approach 1 (norm1)). Whereas, Approach 2 has two autoencoder networks, N1 trained on ROBIN and GI-ROBIN and N2 trained on S-ROBIN and GS-ROBIN.

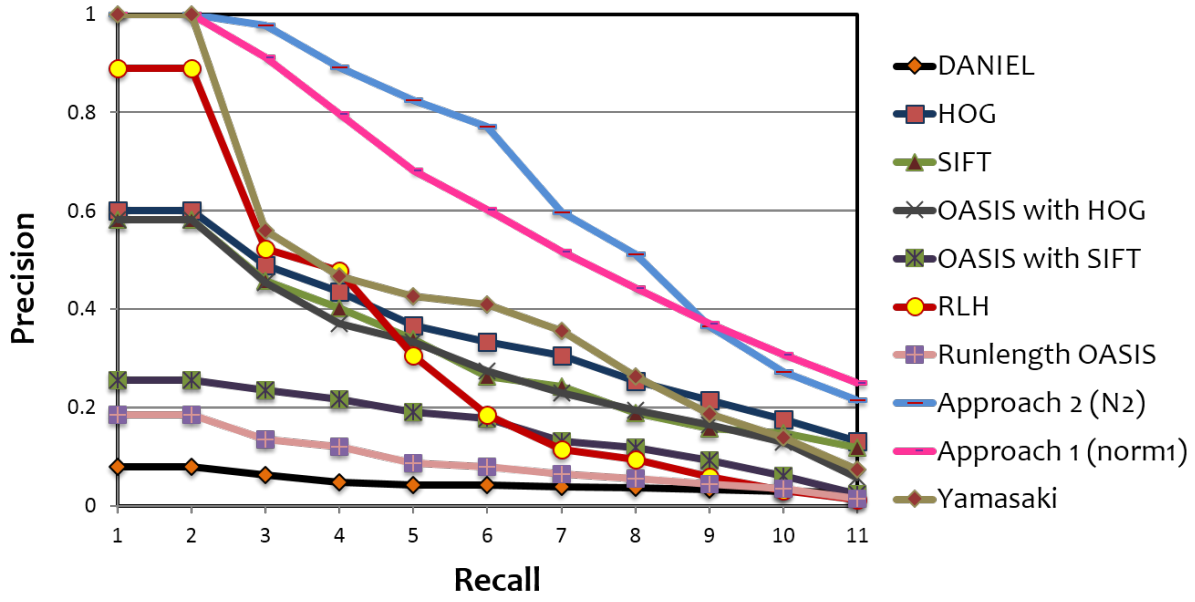

Figure 6.18. : PR plot, comparing the best of the proposed two approaches (Approach 1 (norm1)) and (Approach 2 (N2)) with SOA techniques.

The proposed approach was also compared with the very recent technique mentioned in [Yamasaki *et al.*, 2018] that segments the floor plans using CNN and matches them using Maximum Common Subgraph. It was observed that the proposed technique performs considerably well as compared to [Yamasaki *et al.*, 2018] (improvement in average precision value by 0.254), because the matching criteria in Yamasaki *et al.* [2018] only pertains to matching adjacencies and is also not well scaled for domain adaptation from sketch to image. Also, SIFT, HOG, Runlength Histogram and OASIS approaches perform weakly as they are not able to capture the abstract and sparse nature of sketches and are unable to map between the features of images and sketches. Thus, the comparative results justify the importance of the introduction of both the domain-mapping approaches given a sketch based query and image-based retrieval samples.

6.5 SUMMARY

In this chapter, two frameworks are proposed to handle multimodal retrieval in floor plans. Firstly, a deep learning framework using GAN model is proposed for sketch-based retrieval of building floor plan images. Sketch-based queries make the framework more convenient for the end user to access. Moreover, sketch based queries help in capturing the user's intent in the best possible manner. Due to the limitation on the available images in the database, it is not always necessary to find a perfect mapping of what the user desires to query in the database. Hence, drawing one's ideas on a device which can capture the basic structure of a floor plan and return similar images can prove to be beneficial as an application. The framework is capable of achieving an average precision value of 0.63 upon experimenting with 510 real-world floor plan images and their sketch replicas. Also, the framework is capable of partial matching of sketched floor plans. These abstract floor plans match in certain features, if not all, to the floor plan images in the database. Even if the user has some basic idea of what he desires his floor plans to be then the user can use this

framework for retrieval of similar floor plans.

The second framework applies domain adaptation techniques that help to map both sketches and images to a single framework and are capable of performing better retrieval. There was a need to develop a unified framework capable of both image to image and sketch to image retrieval to serve as a composite tool for multimodal retrieval in floor plans. It was observed that autoencoders in conjunction with Cyclic GANs performed better during retrieval of architectural floor plans giving an average precision value of 0.642 which is improvement by a value 0.1 over the network containing only Cyclic GANs for domain adaptation.