5

Floor plan interpretation in natural language using machine learning techniques

In Chapter 4, understanding of a floor plan image and generating textual description from it was discussed. The floor plans used, were annotated and efficient OCR techniques were used to extract information from them. However, while considering un-annotated floor plans, OCR techniques can not be used. In that case some advance level learning based model is required and consequently some robust feature representation of floor plans becomes key requirement. In a wide perspective, feature engineering approaches are categorised into two classes: hand-crafted features and machine learnt features (advance deep learning models). Floor plan understanding requires a holistic representation of the entire plan in the form of features. The current available hand-crafted features in literature for image representation are designed to represent natural images. Also, in case of line drawings, the hand-crafted features are more suitable for representing symbols and other graphics. They are not suitable for representation of the entire floor plans. Hence, a feature representation was required which could fulfill following requirements of floor plan understanding: (1) Should be able to capture low level information such as decor, walls, doors. (2) Should be able to capture holistic information such as different room names and difference between their representations.

Hence apart from understanding a floor plan with annotated data, we propose their feature representations for annotations learning model for general floor plan images. Since, detection and classification of decor components is also an important step in floor plan understanding, we also propose a signature based algorithm for decor identification. In this chapter we propose technique, SUGAMAN (Supervised and Unified framework using Grammar and Annotation Model for Access and Navigation), to understand and describe un-annotated general floor plan image by proposing a learning based model.

SUGAMAN is a Hindi word which translates to easy passage from one place to another. Apart from describing the general information about the floor plan images, it also generates room to room navigation information, while avoiding the obstacles. This navigation information can be very useful for visually impaired people as it becomes difficult for them to move in an indoor environment. It will be really helpful for them if there is a system that tells about the surroundings environment in natural language. SUGAMAN generates such natural language description of an indoor environment from building floor plan images, which gives a detailed idea of the indoor environment. Here the input is a building floor plan image and the output is a textual description of the same. The description includes detail about the (i) rooms, (ii) connectivity among the rooms, (iii) type of decor within every room, and (iv) their relative position, and (v) navigational information, while avoiding obstacles.

Figure 5.1 exemplifies the problem and the potential solution for a real-world floor plan images. It also depicts the navigation path to be generated and to be textually described. The key characteristics that makes this work unique are: (i) proposing a unified framework for narration synthesis from floor plan images, (ii) improvement in the previously available techniques for decor characterization, (iii) proposal of novel features to represent a room within a floor plan, (iv) learning

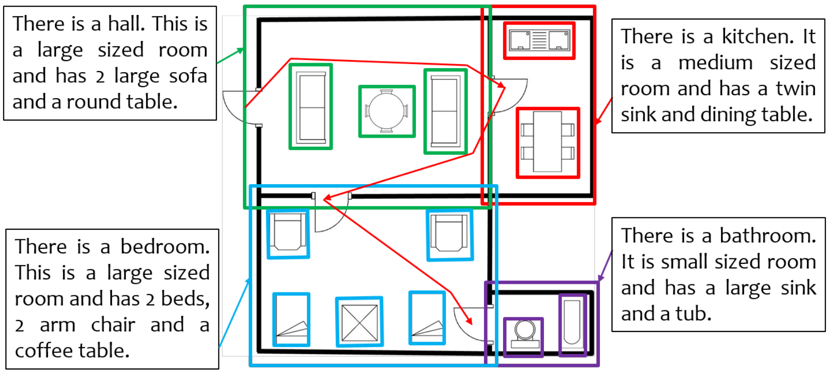


Figure 5.1 : An illustration of the problem of narration synthesis from a given input floor plan image with navigation path.

the room annotations for room classification, and (v) augmentation of an publicly available dataset by annotating floor plan images with textual descriptions. In rest of the chapter, Sec. 5.1 gives a brief overview of the methodology proposed, Sec. 5.2 describes about the dataset in floor plan images and dataset used for experiments in proposed method, Sec. 5.3 describes the detection and identification of each component in the floor plan image, including the proposed features and room classification, Sec. 5.4 gives details about the steps involved in description generation scheme and algorithm for finding door to door obstacle avoidance navigation path, Sec. 5.5 gives detailed analysis of each intermediate step along with experiments performed, Sec. 5.6 discusses the qualitative results of description generation and the limitations of proposed features, Sec. 5.7 discusses the comparative analysis of decor identification method and navigation algorithm proposed and Sec. 5.8 concludes the chapter also drawing outline for future directions.

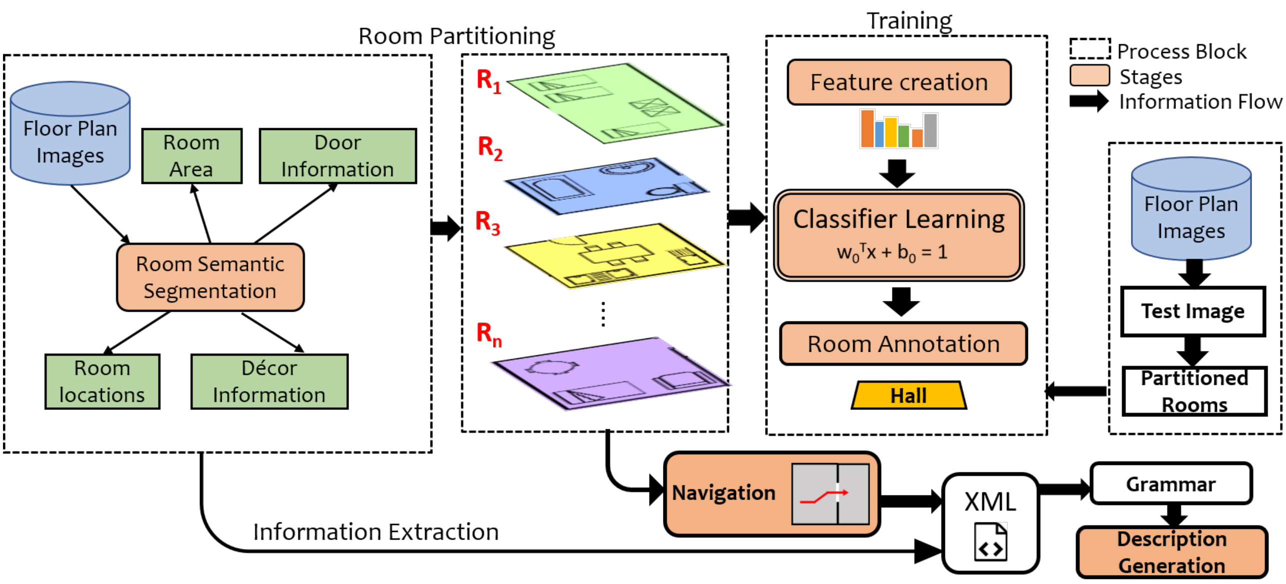


Figure 5.2 : Block diagram depicting various modules and work-flow within SUGAMAN.

5.1 BRIEF OVERVIEW

Figure 5.2 shows a block diagram depicting various modules and work-flow within SUGAMAN. The whole system is divided into two stages (i) room annotation learning and (ii) description synthesis. At first, room semantic information is extracted by room segmentation process, which gives all the required information about an input floor plan image. For example, individual room area, door information which gives room neighborhood information, room locations (room coordinates). With those room locations, the floor plan image is partitioned into individual room image and taken as sample for room annotations learning. Decor characterization is applied over these room images and decors present in a room are labeled. We have proposed Bag of Decor (BoD) and Local Orientation and Frequency Descriptor (LOFD) features, which are extracted from these room samples for automatic room annotation. A classifier is trained using these proposed feature matrix of room samples by assigning class labels to them. After this a new input image is taken as a test sample, features are extracted and room annotations are identified for it using previously trained model. An XML file is generated using the semantic information extracted by room segmentation and room classification. By parsing this XML file, textual description is generated. SUGAMAN also gives navigation path within the entire floor plan, starting from the entry door to the building. All such information about floor plan and navigation are fed to the proposed grammar model. The first stage of the proposed description synthesis method deals with *"what to say"* about the floor plan and the second stage will deal with *"how to say it"*. For ease of understanding, in rest of the chapter, we demonstrate all our analysis on the input image shown in Fig. 5.1. Later, in the experimental results, we also show the results on other floor plan images. Next we discuss about the dataset used for experimentation.

5.2 FLOOR PLAN DATASET

In the literature, three floor plan datasets were proposed, namely (i) Systems Evaluation SYnthetic Documents (SESYD), proposed by Delalandre *et al.* [2010] (ii) Computer Vision Center Floor Plan (CVC-FP), proposed by de las Heras *et al.* [2015] and (iii) Repository Of BuildIng plaNs (ROBIN), proposed by Sharma *et al.* [2017]. SESYD has ten classes of floor plans, with 100 samples/class. On the other hand, CVC-FP has 122 scanned floor plan documents divided into four categories based on the origin and style. In ROBIN there are three broad categories, which are different from each other in terms of the number and type of rooms present in a floor plan. The three categories are (i) 3 room, (iv) 4 room, and 5 room floor plans. Each category is further classified into 10 sub-categories depending upon the global layout of the floor plan. ROBIN helps in better visualization of the floor plans and aids in efficient capturing of various high-level features while fine-grained retrieval. Since ROBIN has significant number of floor plans, as well as intra-class similarity and inter-class dissimilarity, it is suitable in our case. However, in ROBIN there is no textual description available for a given floor plan. For our purpose we further augmented ROBIN dataset by introducing textual description for each floor plan image and termed it A-ROBIN as described in Chapter 3.

5.3 SEMANTIC SEGMENTATION AND ROOM CLASSIFICATION

In all the previous approaches available in the literature, rooms have been classified by recognizing the textual label present in the floor plan image using Optical Character Recognition (OCR) techniques. Room classification in floor plans is not done by extracting salient feature from

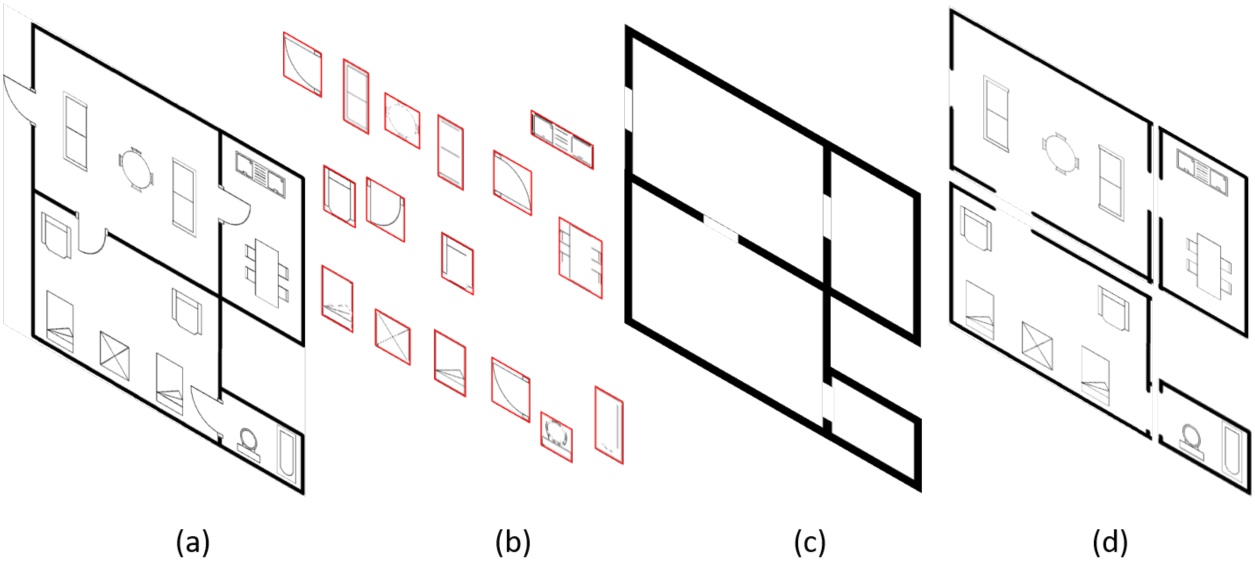


Figure 5.3 : An illustration of the room segmentation and room partitioning process.

it. Room classification on the basis of their functionality is very useful in building information modelling (BIM). When a person enters a room in a house, he or she tells the functionality (class) of the room by looking at the decor items present inside the room. This inspired us to propose a unique feature for room classification. We have proposed new features called Bag of Decors(BoD), which represents the frequency of decors present in a room and Local Orientation and Frequency Descriptor (LOFD), which represents the frequency of decors present in a room along with their normalized distance from the center of the room. We proposed room classification approach as a 5 class classification problem, which annotates each room in a floor plan into one of the 5 classes namely, BEDROOM (label-1), BATHROOM (label-2), ENTRY (label-3), KITCHEN (label-4), HALL (label-5). The following subsections describes the details of room label learning and classification.

5.3.1 Room Segmentation

We have adopted the technique proposed in D. Sharma, C. Chattopadhyay and G. Harit [2016] for the identification of rooms. Walls are detected by performing morphological closing on the input floor plan image I (see Fig. 5.3(c)). To delineate room boundaries, we detect doors

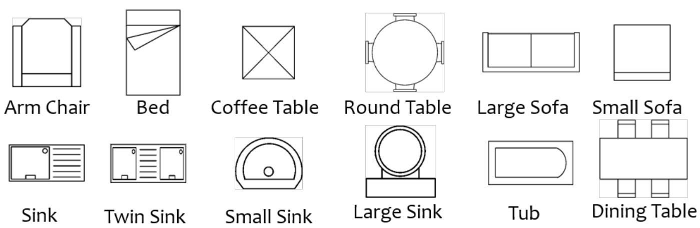


Figure 5.4 : Pictorial depiction of the twelve classes of decor models used in the experiments.

using scale invariant features and close the gaps in wall image corresponding to the door locations. To obtain the rooms, we identified the connected components in the wall image by applying flood fill technique. The obtained connected components are the required rooms and their locations are obtained. Also, we calculate the areas of the respective rooms (polygon area), converted them into square feet (taking 100 pixels= 1 feet) and store all the information obtained, that is neighborhood, room area, room location coordinates, in a separate data structure.

5.3.2 Floor Plan Partitioning

A floor plan image is partitioned into rooms using the room coordinates extracted from the previous steps as shown in Fig. 5.3(d). These individual room images are the samples taken for training the room annotations. We have applied decor characterization in further stages on each of these individual room images to extract the features.

5.3.3 Decor Classification

In this section we describe the procedure employed for decor characterization and their classification. Figure 5.4 shows the 12 decor symbols used in the dataset proposed in, Sharma *et al.* [2017]. We have improved the technique of decor characterization proposed in D. Sharma, C. Chattopadhyay and G. Harit [2016] by applying sequence of morphological operations. The technique in D. Sharma, C. Chattopadhyay and G. Harit [2016] uses a normalized area ratio of largest three components of a decor symbol for classification and characterization of decors. We have improved the technique by first collecting 10 different signatures for each symbol, taking a mean over them (symbols with different orientations) and stored them in a signature library. During classification, we first pre-process the symbol by applying a sequence of morphological operations (erosion and dilation), so that the symbol do not have broken lines. Then we applied blob detection over the image an cropped each decor symbol for signature comparison. Now we compare the test image's signature with the signature stored in library and closest one is classified in its respective category. This modification in the technique greatly improved the classification accuracy for some symbols. Figure. 5.3 (b) depicts the detection of symbols in the floor plan input image Fig. 5.3 (a) with bounding boxes. These decors are classified in their respective categories shown in Fig. 5.4. The signature of the decor symbols if represented by a proposed feature Unique Decor Identifier (UDI), calculated by Alg. 2. This UDI feature is a set of area ratios of three largest connected components in a decor symbol. The decor library is created using the signature function by computing UDI of 10 different symbols and taking an average over it. The formulation of the UDI feature is governed by the following equation:

$$F_i = \sum \frac{f_j}{10}, \text{where j=1...10, i=1...12} \tag{5.1}$$
$$F = \{F_1, F_2, ..., F_{12}\}$$

Our decor characterization method calculates UDI feature of the decor items in sample room images and compares it with the UDI feature present in decor template library. The decor having closest UDI with decor item present in the library, corresponding decor is assigned to it. The cropped room image is given as input and walls are removed from it. After that morphological filling is applied to join the broken lines of the symbols. Each decor symbol is cropped from the image and given for UDI identification, which is further compared with UDI present in decor library and closest decor is assigned to it. Before evaluating UDI of decor items, blob detection is applied

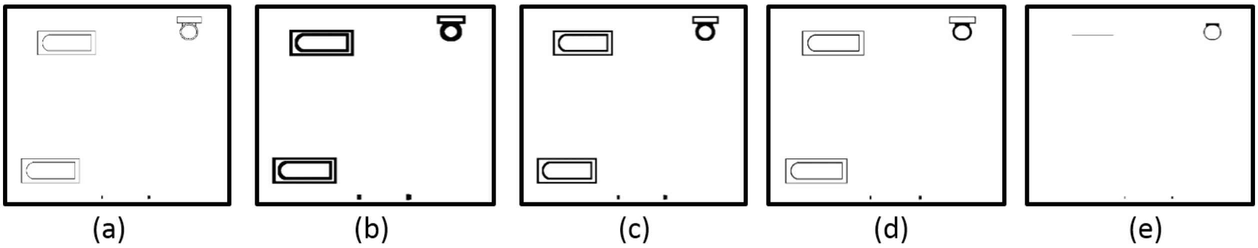


Figure 5.5 : Qualitative comparison between ours and D. Sharma, C. Chattopadhyay and G. Harit [2016] for the task of Decor identification.

over image and number of decor items present is calculated.

Figure 5.5(b)-(d) shows the intermediate processing stages of our proposed model. We have applied erosion with structuring element of size 8, followed by dilation. However, the number of times should these morphological operations be applied has an effect on the overall result. Figure 5.5(e) shows the over application of filling operations and results in fading of symbols.

5.3.4 Bag of Decor (BoD)

Once the decors inside a room are recognized, we compute the representative features to classify the room. For room classification, we proposed a new feature named Bag of Decor (BoD). It is a 1×12 vector containing the decor information of a room sample. Since there are 12 decor models, every element of this vector represent the count of one decor item. However it is not necessary for any room to have all types of decor present, therefore BoD is sparse in nature. Figure 5.6 shows the feature vector for the room shown as inset. In this example, the room has three types of decors, 3 sofas, 2 tables, and 2 chairs. As shown schematically in Fig. 5.6, the BoD feature vector has the count of the specific type of furniture, while the other bins are assigned a value of zero. After the room classification, this room is classified as ENTRY having 2 small sofas, 3 large sofas and 2 arm chairs.

5.3.5 Local Orientation and Frequency descriptor (LOFD)

Another feature which is proposed for room classification, named Local Orientation and Frequency descriptor (LOFD) along with BoD. BoD only captures the frequency count of decor components, while LOFD also captures their respective orientations inside a room for a more

Algorithm 2 UDI Computation

procedure SIGNATURE(J)
    $C = CC(J)$ ▷ CC:Connected components
    $Count = |C|$ ▷ | |: Cardinality of connected components
    for k=1 to Count **do**
        $\mathscr{A}_k = Area(c_k)$, where $c_k \in C$
        $A = Sort_{desc} \mathscr{A}_k$
        $F(J) = \{(A_1/A_3), (A_2/A_3), 1\}$ ▷F : *Signature*

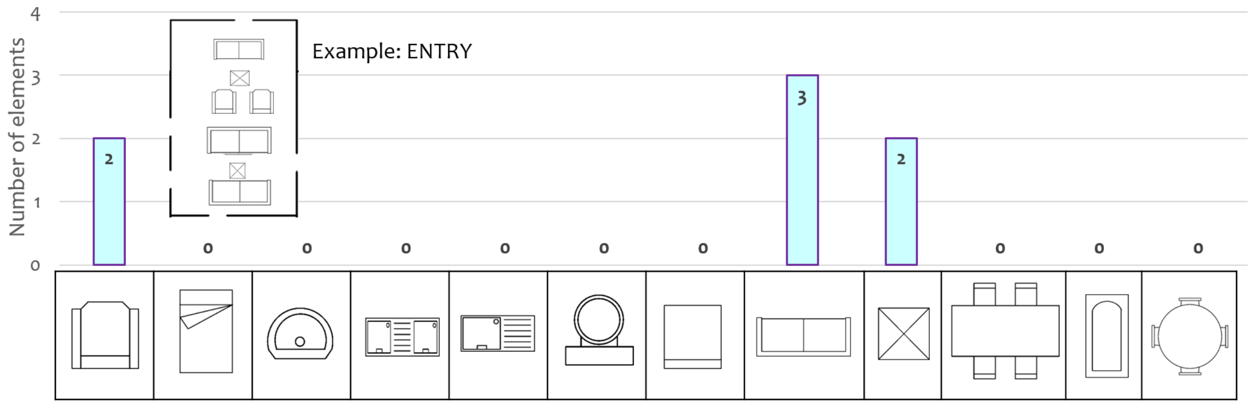


Figure 5.6 : An example of a Bag of Decor (BoD) feature descriptor computed for a sample room image.

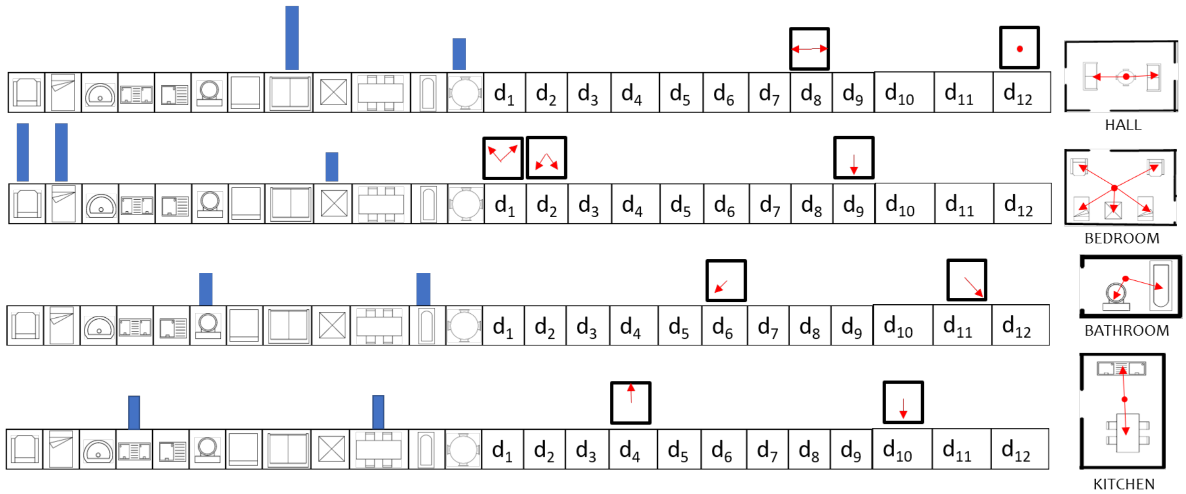


Figure 5.7 : AN schematic illustration of computing the LOFD descriptor for four sample room images.

descriptive building information modelling. The LOFD feature is a 1×24 vector containing the decor information of a room sample and their locally aggregated spatial information. Figure. 5.7 shows the LOFD feature matrix for the sample floor plan image. In LOFD, we have aggregated the local information of the room image in a vector form. LOFD is compact representation of the frequency of the decor items and normalized distance of their centers from center of the room. The first 12 cells of the vectors are occupied by the 12 decor items from $D = \{D_1, D_2, ...D_{12}\}$ and next 12 cells are occupied by their normalized distances as $D = \{d_1, d_2, ...d_{12}\}$ where d_n is the distance of each decor item from the center of the room. Here, d_n is calculated as:

$$d_n = \frac{\sum_{i=1}^{k} dist(\mathscr{R}_c, \mathscr{D}_c)}{max(D_n)} \tag{5.2}$$

Here, d_n is the normalized distance for each decor item, i is the count of each decor which may go up to k, which is the maximum number of a that decor item in the room, $dist$ is the Manhattan distance between room center $\mathscr{R}_c$ and decor center $\mathscr{D}_c$, $max(D_n)$ is the maximum of all the distances

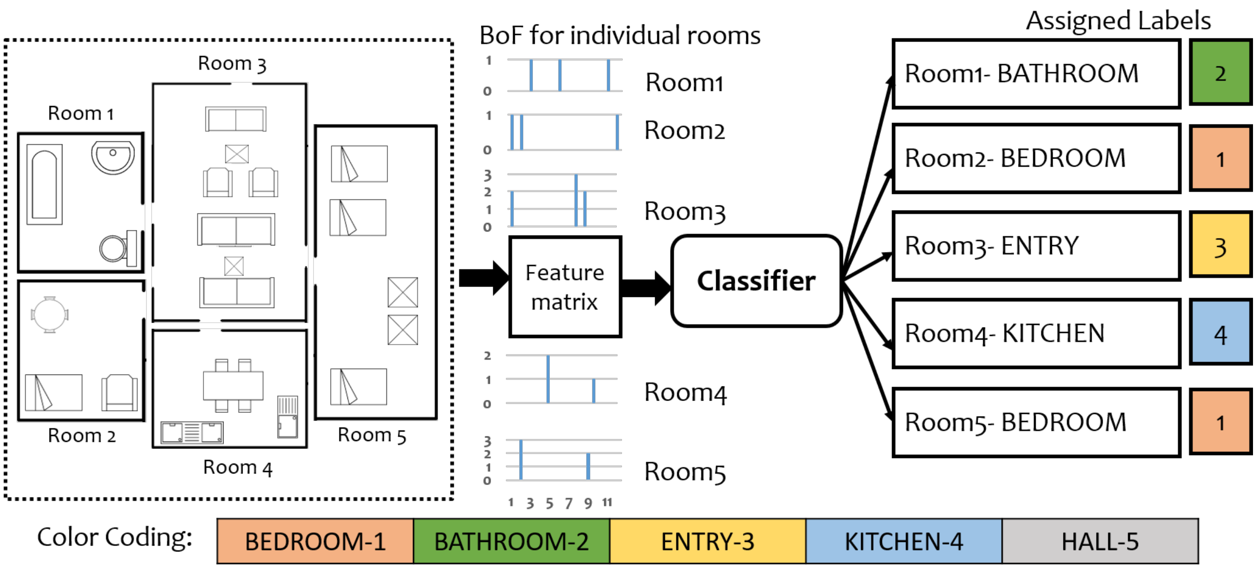


Figure 5.8 : Schematic representation of automatic room label classification.

obtained for all decors to normalize the distance value. Hence, LOFD feature distinguishes each room uniquely by the frequency of each decor item and their spatial location in the room.

Since there are 12 decor models as shown in Fig. 5.4, first 12 element of LOFD represents the count of one decor item. However, it is not necessary for any room to have all types of decor present, therefore LOFD is sparse in nature. Figure 5.7 depicts the room image followed by the corresponding LOFD feature vector. The colored bar over each cell represents the frequency count for each decor item, while arrows in red represent their relative spatial location in the room. In the next section, training of the classifier using the features proposed for room classification is explained.

5.3.6 Room annotation Learning and Room classification

Room annotations for training samples (divided 1355 room images into 70% and 30% for training and testing respectively) are learned by the two proposed features and classified into predefined categories. For training purpose, we have manually annotated the room samples and used those annotations during training. Extensive experiments were performed using various classifiers and the best classifier in term of highest training accuracy is taken for testing the model. For testing purpose, an image from the test set is taken and class labels are evaluated accordingly for the room samples of that floor plan image. For each new test floor plan image, the feature vector is evaluated for every room. Therefore, dimension of the feature matrix for a test floor plan image will be $N_r \times Dim_F$ where N_r is the number of rooms in the floor plan and Dim_F is the dimension of the feature vector. Trained classifier is used for this feature matrix and the output class labels are evaluated. Figure 5.8 represents a schematic diagram of room annotation learning and classification in different classes with the proposed features.

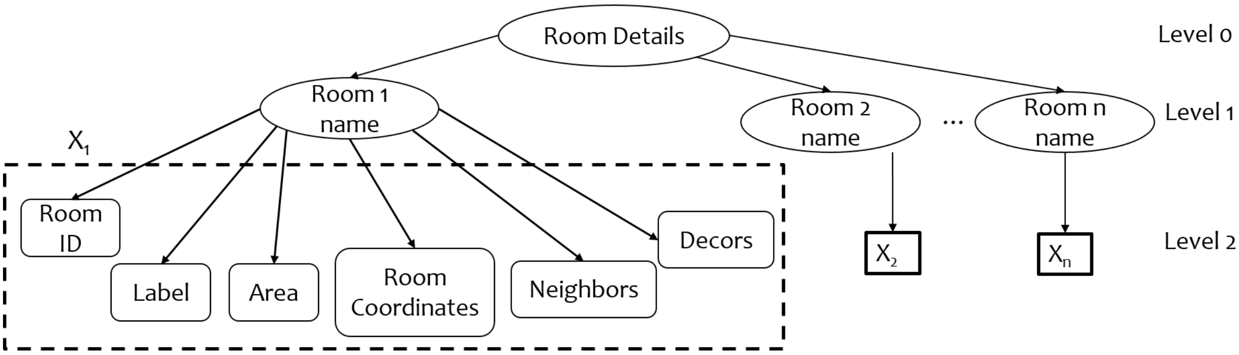


Figure 5.9 : Structure of the XML file generated for description synthesis.

5.4 DESCRIPTION SYNTHESIS

Rooms are classified and their annotations are learned in Sec.5.3.6. Information extracted from room segmentation are combined and used for generating the description of the floor plan image. Information related to individual rooms are combined and stored in an XML file, which is parsed to generate description of the floor plan.

5.4.1 XML File generation

An XML file has many benefits in terms of cross platform portability, ease of understanding by novices, and extendability. We have created an XML file by combining the semantic information extracted from room segmentation and room annotations learned in previous steps. As shown in Fig. 5.9, the tree-like structure of XML file contains "Room details" as the root node at level 0, "Room names" as nodes at level 1, and information of rooms as nodes at level 2 (leaf nodes), which are Room ID, Room annotations, Room area, Room Coordinates, Room neighbors and Room Decors. Apart from room annotations, a room ID is given to each room since room annotations can be same for two rooms. to generate a description.

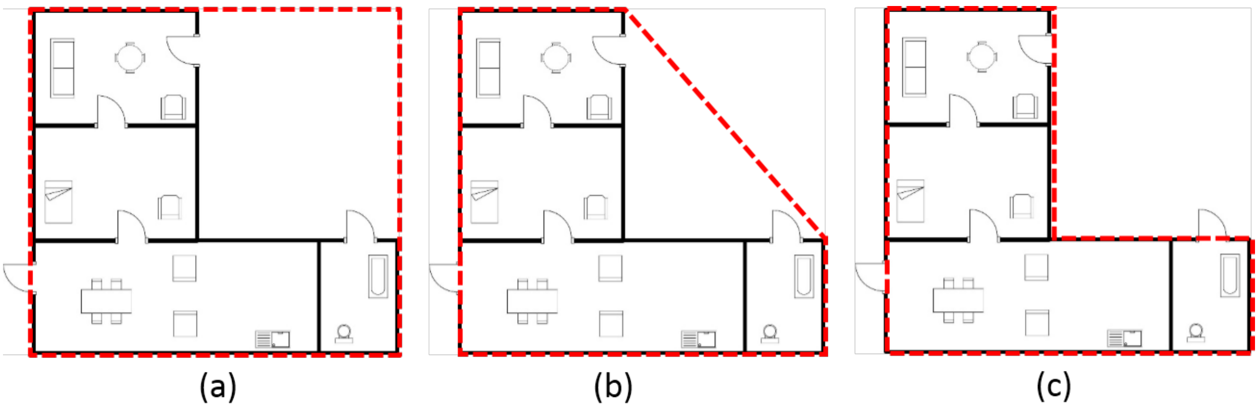


Figure 5.10 : An example of the boundary tracing process for different values of the shrinkage factor t.

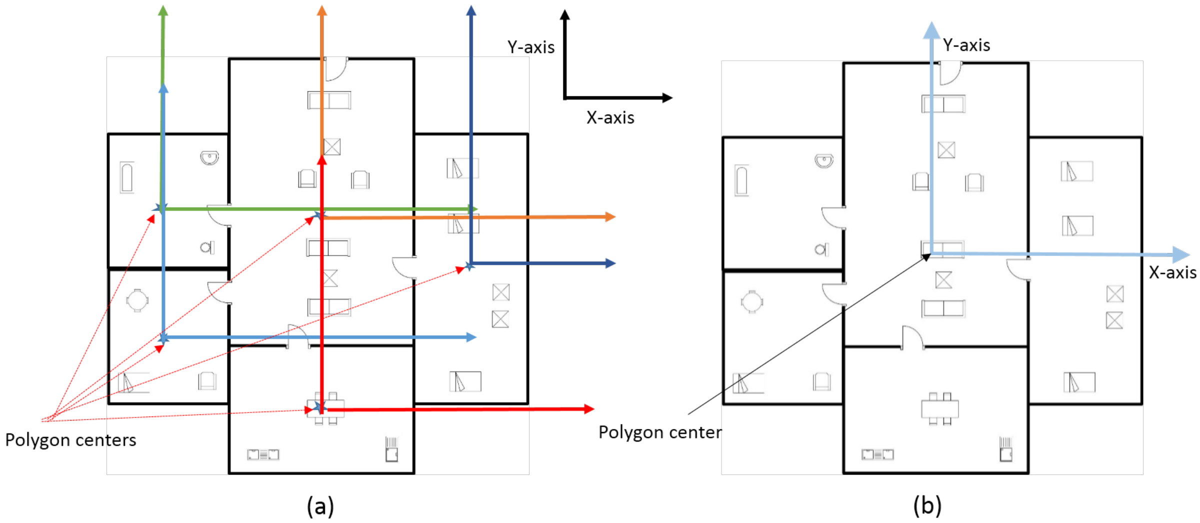


Figure 5.11 : An illustration of the global and local coordinate systems with respect to a floor plan image.

5.4.2 Coordinates systems

For defining the positions of rooms and decors present in the floor plan, we have defined two coordinate systems. The global coordinate system is to identify the global location of rooms with respect to the entire document. Local coordinate system is to define the relative position of decors with respect to each room.

Boundary Tracing

The origin of the global coordinate system is the polygon center, which makes the floor plan's boundary. A floor plan's boundary is traced to identify the center of the floor plan. Individual rooms' coordinates are plotted to trace the boundary, and the outer boundary is tracked, which encloses all the outer points since these points collectively make the floor plan image. However, by tuning the value of shrinkage factor t between 0 and 1 we can switch between a convex hull of those points and a more close-knit boundary. The shrinkage factor defines how closely the hull envelops the boundary points. For example, in Fig. 5.10 (a), the boundary traced is a convex hull of the floor plan image for the shrinking factor value $t = 0$, Fig. 5.10 (b) is the traced boundary for shrinking factor value $t = 0.5$ and Fig. 5.10 (c) is the close-knit boundary for $t = 0.8$. Therefore, by tuning the shrinking factor value, we obtain a close-knit boundary for the floor plan image.

Global and local coordinate systems

A global coordinate system defines the global position of all the rooms in a floor plan image (see Fig. 5.11 (b)). From the traced boundary obtained in the previous step, we calculate the global coordinate system's origin. Equation 5.3 and 5.4 lists the governing equations.

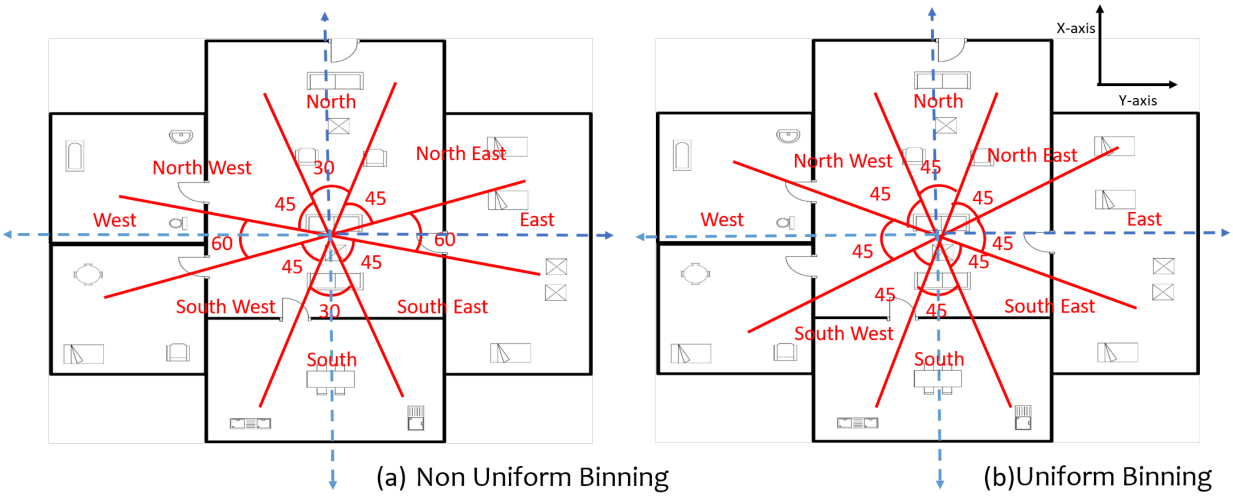


Figure 5.12 : A schematic representation of the non-uniform and uniform binning.

$$\begin{aligned} a_i &= x_i y_{i+1} - x_{i+1} y_i \\ A &= \frac{1}{2}\sum_{1}^{n} a_i \end{aligned} \tag{5.3}$$

Where, a_i in Eq. 5.3 is twice the signed area of the elementary triangle formed by (x_i, y_i) and (x_{i+1}, y_{i+1}) and the origin. A in Eq. 5.3 is the area of the polygon.

$$\begin{aligned} x_c &= \frac{1}{6A}\sum_{1}^{n} a_i(x_i + x_{i+1}) \\ y_c &= \frac{1}{6A}\sum_{1}^{n} a_i(y_i + y_{i+1}) \end{aligned} \tag{5.4}$$

In Eq. 5.4, (x_c, y_c) is the center of the polygon. The local coordinate system (see Fig. 5.11 (a)) identifies the relative positions of all decors with respect to each room. Center of each room, for a local coordinate system is computed using Eq. 5.3 and 5.4.

5.4.3 Binning

We have performed global and local binning or radial partitioning of the floor plan (see Fig. 5.12). The nonuniform binning angles were empirically determined. For identifying the direction of decor, the center of the surrounding bounding box is taken as the reference point. While for the rooms, their respective centers, obtained in the previous steps is taken as a reference point. As shown in the Fig. 5.12(a), the entire coordinate system is divided into 8 directions, north,

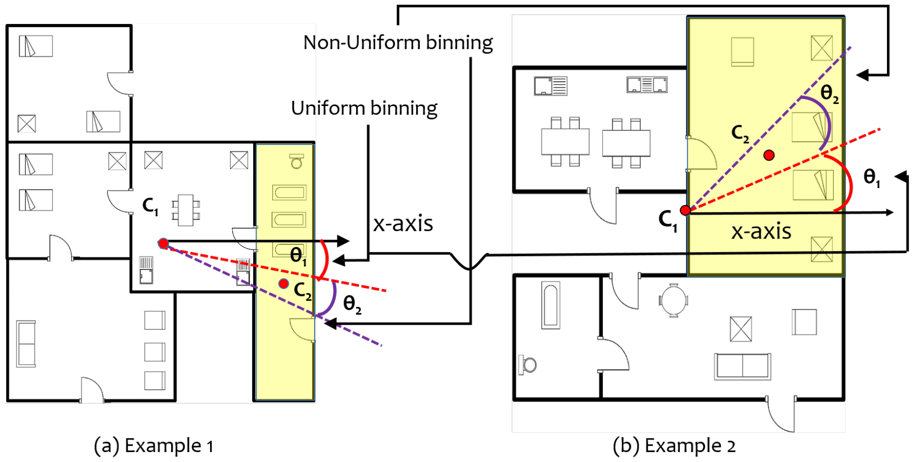


Figure 5.13 : A visual illustration for explaining the rationale of non-uniform binning.

north-east, east, south-east, south, south-west, west, north-west, in the clockwise direction. The binning depicted in Fig. 5.12(a) is a nonuniform binning, while in Fig. 5.12(b) is a uniform binning.

The rationale behind non-uniform binning is to provide a piece of more realistic direction information for rooms and decors. The idea of taking the direction from the center of the surrounding polygon may misguide the framework about the actual position of a room. E.g., if a room is located in the west direction and stretches towards the north, its center will lie in a northwest direction even if the room is in the west. To avoid these kinds of ambiguities, binning is done non uniformly, and the angles are empirically taken. Figure 5.13 highlights examples for the above rationale. The highlighted room (Fig: 5.13 (a)) is more toward the east direction. However, it is also extended towards the south. With nonuniform binning, we try to increase the east direction span, showing a purple line and an arc where $(\theta_1+\theta_2)$ is the angle of nonuniform binning. The red line and arc show the span of uniform binning, making the room fall in the southeast direction and creates ambiguity. Here θ_1 is the angle of uniform binning, C_1 and C_2 are the centers for floor plan and room, respectively.

5.4.4 Navigation

Navigation in the indoor environment by avoiding obstacles is an integral part of SUGAMAN. We have proposed a grammar-based model that yields navigational directives to navigate the house for a natural movement from one door to the other door of each room. The algorithm is divided into two parts. First, we create a data structure that stores the room labels and their respective entries and corresponding indexes. The room information and the door coordinates are obtained from semantic segmentation in the earlier stages (see Sec. 5.3.1). If a door is shared between two rooms, that door will be present in both room's door structures, and the index will represent the door's identity. Next, we identify the entry room and the corresponding door and generate a Depth First Search (DFS) ordering of the region adjacency graph of the floor plan taking the entry room as the start node. After that, a path to the next room is generated, avoiding obstacles by checking the visibility from the first door to the other. We also create a door-based adjacency matrix (AM_D), which stores the rooms' shared doors.

Creating door structure

The room coordinates, room labels, and door coordinates are obtained in the semantic segmentation. After that, an index i_d is assigned to each door. We have checked whether a given door i_d belongs to a particular room or not. We have performed an inside-outside test between the bounding polygons of the doors and rooms to achieve belongingness. The door structure contains each room with its corresponding doors having marked with their index i_d. As shown in Fig. 5.14(b), room and door information is stored in a door structure.

Algorithm 3 Room-to-Room traversal within a floor plan

for $i \leftarrow 1, N_r - 1$ **do** $\triangleright$ N_r: No of rooms
  $Backtrack \leftarrow 0$
  $c_r \leftarrow i$ $\triangleright$ c_r:Current room
  $n_r \leftarrow i+1$ $\triangleright$ n_r:Next room
  if $AM_D(c_r, n_r) == 1$ **then**
    if $Backtrack \neq 1$ **then**
      $DC \leftarrow \textsc{ClassifyDecor}(R_i)$ $\triangleright$ R_i:Room Image
      $V_L \leftarrow \{DC\}$ $\triangleright$ V_L: Vertex list
      $D \leftarrow \textsc{DetectDoorCentroid}(R_i)$
      $V_L \leftarrow V_L \bigcup D$
      $R_i^{new} \leftarrow \textsc{RemoveDoors}(R_i)$
      $B \leftarrow Blobs(R_i^{new})$
      $Corners \leftarrow \textsc{HarrisCorner}(B)$
      $C_S \leftarrow \textsc{Strongest}(Corners)$
      $V_L \leftarrow V_L \bigcup C_S$
      for $j \leftarrow 1, N_{V_L}$ **do** $\triangleright$ N_{V_L}: No. of elements in V_L
        for $k \leftarrow 1, N_{V_L}$ **do**
          $visible \leftarrow \textsc{Visible}(V_L(j), V_L(k), R_i^{new})$
          if $visible$ **then**
            $AM_N^i(j,k) \leftarrow \mathrm{ED}(V_L(j), V_L(k))$
          else
            $AM_N^i(j,k) \leftarrow 0$
          end if
        end for
      end for
      $D_E \leftarrow R_{c_r}(Entry)$
      $D_X \leftarrow R_{n_r}(Entry)$
    end if
    $Path(i) \leftarrow \textsc{Disjkstra}(AM_N^i, D_E, D_X)$
  else
    $Backtrack \leftarrow 1$
    $D_E \leftarrow R_{c_r}(Exit)$
    $c_r \leftarrow c_r - 1$
    goto Step 6
  end if
end for
return *Path* $\triangleright$ Path to go to every room from the entry =0

Path Finding

DFS search is performed over the floor plan image's region adjacency graph, taking the entry room as starting node. The door connected to the entry room's outer wall is considered the entry door and stored in the door structure. Here, the entry door for the house is detected by the the algorithm discussed in S. Goyal, C. Chattopadhyay, and G. Bhatnagar [2018]. Algorithm 3 describes the process of room to room navigation by obstacle avoidance. The route in each room is stored in the form of coordinates (x, y) of the movement and included in the description for narration of the path. Algorithm 3 traverses the rooms starting from the first room in the DFS graph by checking if there is a door shared between them. This is checked by door-based adjacency matrix (AM_D). Suppose they do not share an entry, the algorithm backtrack and explore other rooms. Also, it determines the route across the rooms for navigation. In Alg. 3, line 2 declares the flag, if the algorithm has to enter into backtracking. Line 4 describes the loop which traverses room to room, finding the path. Line 6, algorithm checks if there is a shared door between the current room and the next room and continues traversal between rooms if there is a shared door. Line 7 directs the algorithm to further processing if backtracking is not required. Line 8 to 8, detects the coordinates of the bounding box of decor items and centroid of doors of the current room and include them in a vertex list. In 9, doors are removed from the room image because they are not required for avoiding obstacles. Lines 10 to 15 detect the corner points in the room image using Harris corner detector after detecting the blobs and include top 1000 strongest corners in the vertex list. Line 16 to line 22 describes the construction of adjacency matrix for navigation (AM_N). It checks the visibility between every point in the vertex list and includes the Euclidean distance between them in AM_N as the weight at $AM_N(V_L(\mathrm{j}),V_L(\mathrm{k}))$. Visibility between two points is checked by filling the line between equal intervals in those points and checking if there is a black pixel present. If there is a black pixel present, then there must be an obstacle between those two points, and hence those points are not visible. $AM_N(V_L(\mathrm{j}),V_L(\mathrm{k}))$ will have a 0 in that case. Line 26 and 27 define the entry (D_E) and exit (D_X) door for the current traversal, where the entry door is the entry of the current room and exit door is the entry of the next room. Line 29 evaluates a route (P^i) for current traversal by applying Dijkstra's shortest path algorithm over AM_N taking D_E and D_X as start and end nodes. Line 31 to 34 defines the backtracking process if there is no shared door found between the current room and next room. The algorithm will backtrack in the DFS path and find the navigation path between corresponding rooms. The route for i^{th} room (P^i) is a set of coordinates containing the start point, endpoint, and intermediate turns that a person has to make for obstacle avoidance. Figure. 5.14 describes the entire process for the input image Fig. 5.14(a). The checkered box (inset) depicts AM_D, where the dark box represents a 0, and a white box represents a 1.

Figure. 5.14(b) shows the door structure created in the previous step and traversal order with backtrack step. Figure 5.14(c) shows the DFS search graph generated over the region adjacency matrix to obtain the order to traverse each room. Figure 5.14(d) shows the local coordinate system fitted over every point in a route while traveling through the floor plan and also shows the direction of movement by arrows. Figure 5.15(a) represents the door to door path generated for navigation, avoiding obstacles in each room in the input image. Figure 5.15 shows some other examples describing the path generated with various floor plan images.

5.4.5 Proximity based sentence model

The XML file's parsing yields 5 types of information for each room, defined in separate sentences; room name, area, neighboring rooms, global position, and contained decors with their relative position in the room. For that purpose, we defined the sentence model having 6 rules based on proximity, as shown in Tab. 5.1. The first sentence (S_1) of every floor plan description is a

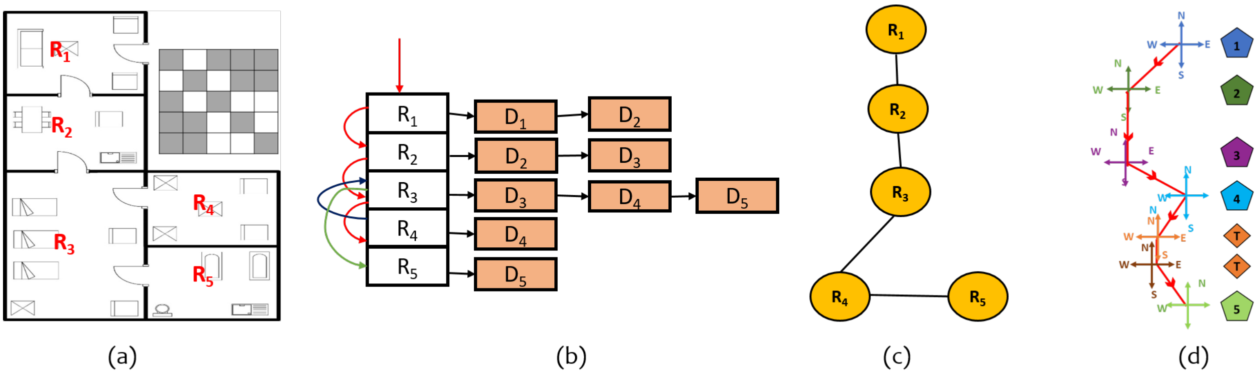


Figure 5.14 : Illustration of path detection process.

general sentence stating the number of rooms (N_r) in the floor plan. In S_2, DT is a determiner that takes its value from the set {a, an}. In addition, O_i is the object which takes its value from level 1 nodes (Room names), where value i varies from 1 to N_r. In S_3, AREA takes its value from the *RoomArea* tag when XML file is parsed. In S_4, s takes its value from the set {s,ϕ}, which is a proximity-based value depends upon its previous word. Value s is chosen if the word in proximity (room) is plural and ϕ otherwise. Also, AUX is an auxiliary verb, which takes its value from {is, are}, depending upon its proximity word, and NR_j takes its value from Neighbors tag (neighboring rooms) when the XML file is parsed. Here, the value of j varies from 1 to NN_r, which is the number of neighboring rooms. In S_5, LOC is the room's global position, which takes its value from the set {North, North East, East, South East, South, South West, West, North West} described by binning. In S_6, the value of k varies from 1 to DC i.e. decor count. Here, C is the count of individual decor item, D takes its value from the Decor tag in XML file, s takes its value from {s,ϕ} and $DLOC$ is the relative location of decor in the room which takes its value from {North, North East, East, South East, South, South West, West, North West} described by binning.

S_7 is the sentence narrating the navigation, where N_{step} is the number of steps to be taken. We took the Euclidean distance between the first coordinate and the next coordinate in the route to calculate the number of steps and calibrate the distance into steps (10 pixels= 1 step). Also, DIR is the direction in which the person has to move, for which the local coordinate system is being fit to every coordinate of the route. It takes its values from the set {North, North East, East,

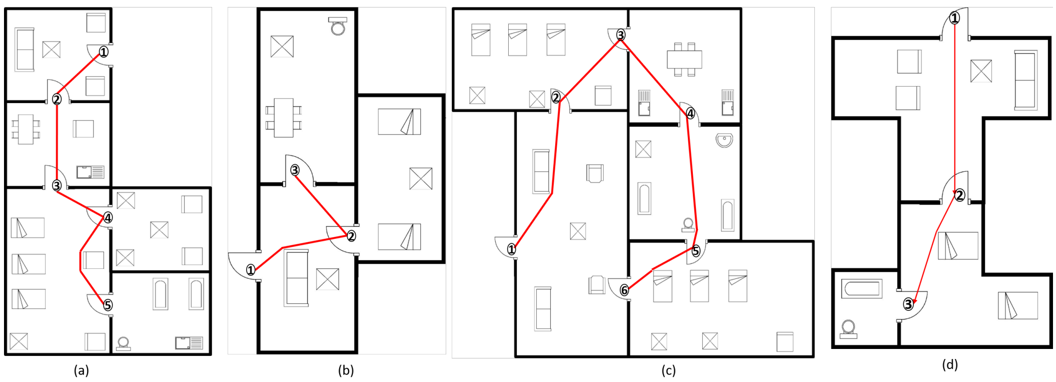


Figure 5.15 : Examples illustrating path detection by avoiding obstacles.

Table 5.1 : Sentence model based on proximity

Sentence	Rule
S_1	This floor plan has N_r rooms
S_2	There is **DT** O_i
S_3	It has an area of **AREA**
S_4	Its neighboring room{s} **AUX** NR_j
S_5	It is located in the **LOC**
S_6	This room has $\{\mathbf{C}\,\mathrm{D}\{s\}\ \textit{at the}\ \mathbf{DLOC}\}_k$
S_7	$\{\{$ Go N_{step} steps in **DIR** direction $\}_{N_m}\}_{N_r}$
S_8	There is a door and a room. $\{\mathrm{S9}\}_{if\ dead\ end}$ $\{\mathrm{S7}\}_{else}$.
S_9	You have to turn back.

South East, South, South West, West, North West}. The number of coordinates returned in the navigation route inside a room is the number of turns a person will have to make. Here N_m is the number of turns inside a room, and N_r is the number of rooms. S_8 describes the door and room found after navigating through the previous room. If the room has only one entry and hence a dead end, the person will turn back and navigate further (S_9), else he will go straight and explore the other rooms by entering (S_7).

5.5 ANALYSIS OF INTERMEDIATE STEPS

We have performed our experiments on a hardware platform with the following configurations. The system has an Intel core i7 (8^{th} generation), with a 1.87 GHz processor. It has a memory of 8 GB where, implementation has been done on Matlab 16a.

Table 5.2 : Classifiers results of room annotation learning by BoD feature.

Variant	Training (R) (%)	Testing (R) (%)	Testing (S) (%)	Training (R+S) (%)	Testing (R+S) (%)
linear svm	89.2	80.00	66.51	89	76.41
Quadratic SVM OVA	89.0	78.67	64.62	90.8	76.92
Cubic SVM OVO	88.6	79.56	55.19	88.7	74.70
Medium Gaussian SVM	88.7	78.44	58.96	89.3	75.21
Quadratic SVM OVO	88.2	80.44	65.57	89.7	75.90

5.5.1 Room annotation learning and Classification

For this task, a dataset of 1355 room images divided into 70% and 30% for training and testing, respectively. The two features proposed (see Sec. 5.3.4 and Sec. 5.3.5) are used to train a multi-layered perceptron (1 hidden layer with 10 neurons).

Table 5.3 : Classifier-Results of room annotation learning by LOFD feature.

Variant	Training (R) (%)	Testing (R) (%)	Testing (S) (%)	Training (R+S) (%)	Testing (R+S) (%)
Linear	90.1	78	66.04	89.1	73.4
Quadratic	87.8	79	60	91.6	78.4
Cubic	87.8	77.1	63	91.2	76.6
Medium Gauss	88.3	76.7	58	90.5	75.3
Quadratic	87.1	77.1	60	90.9	74.2
Complex Tree	88.3	76.4	65.6	88.7	73

BoD classifier

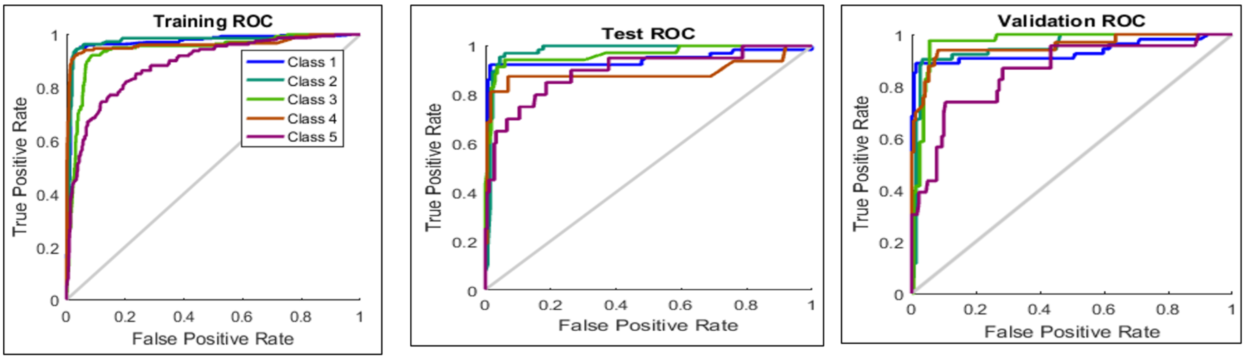
Table 5.2 shows the comparative analysis of different classifiers using BoD classifier in which the first column shows the training results using 1355 room images taken from ROBIN dataset (denoted as R), and the linear SVM gives the highest training accuracy. In the second column, testing is done using samples taken from ROBIN. In the third column, test results are shown for SESYD samples (denoted as S) by a trained ROBIN image model, which is comparatively low. Experiments are done by training the classifier using mixed samples from both datasets using 1940 images and training and testing accuracy shown in the fourth and fifth columns. It can be easily noted that the testing accuracy is considerably increased with this model. Figure 5.17(a) shows the experiment of room annotation learning with the neural network using BoD classifier, showing the training, testing, and validation accuracy achieved. In addition Fig. 5.16 (a) shows the ROC curves for training, testing and validation.

It is indicative that we achieved the best validation at 30^{th} epoch of the training cycle. It is clear from the test ROC that class 5 samples have minimum testing accuracy. The curve moves much away from the upper left corner and goes below the diagonal line (moving towards the false positive axis). We can also see that class 1 samples have maximum training accuracy from the training ROC curve. The curve remains concentrated in the upper left corner (moving towards the true positive axis). Low training and testing accuracy for class 5 samples result from its less number of samples.

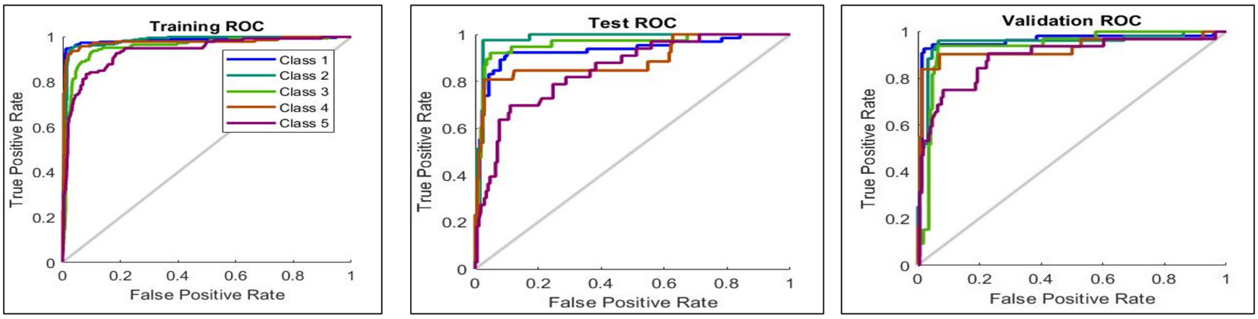
LOFD classifier:

Figure 5.16(b) shows the the ROC curves for training, testing and validation for neural network using LOFD classifier, in which it is clear that ROC curve for class 5 moves maximum towards false positive axis, because of less number of training samples for class 5. Furthermore, for class 1 and 4 it remains on true positive axis due to more number of samples in the training data. Figure. 5.17(b) (column 1) shows the performance curve for neural network in which best validation performance is achieved at epoch 49. The training, testing, and validation accuracy obtained by the neural network are 88.3%, 81.3% and 85.2%, respectively.

Experimental results with other supervised classifiers using the LOFD feature are shown in Tab. 5.3. For training the classifiers, as shown in Tab. 5.3, first we divided 1355 samples from ROBIN(R) dataset in 70% (training) and 30% (testing). Training and testing accuracy are shown



(a) ROC curves for BOD



(b) ROC curves for LOFD

Figure 5.16 : Training, testing and validation ROC curves for BoD and LOFD features.

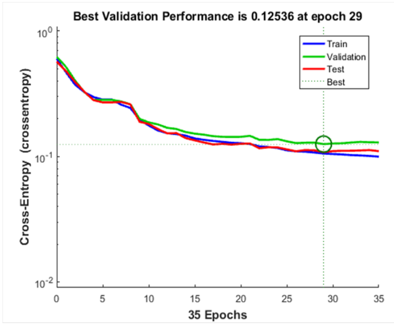
in the first and second columns, respectively. We tried testing the sampled from SESYD (denoted as S) dataset with this trained model. However, the testing accuracy statistics (column 3) are not up to the mark. Hence, we mixed the samples from both datasets. Taking 500 samples from SESYD and 1355 samples of ROBIN, making it a collective dataset of 1855 images, other models were trained and training and testing are shown in column 4 and column 5, respectively. The best performing classifier is linear Support Vector Machine (SVM), one versus one, for ROBIN dataset and quadratic SVM (one versus all) for mixed samples making LOFD a highly accurate feature descriptor for room annotation learning in floor plan images.

Table 5.4 : Performance analysis of text generation algorithm using ROUGE score.

ROUGE	Average Recall	Average Precision	F score
ROUGE-1	0.5061	0.2715	0.3445
ROUGE-2	0.1545	0.5707	0.07616
ROUGE-3	0.0535	0.01093	0.01483

5.5.2 Description synthesis

All the reference corpus available in the A-ROBIN dataset and the generated descriptions were tokenized using the "Penn Treebank tokenizer'' Marcus *et al.* [1993] and utilized during the evaluation. We have compared the machine-generated description of the floor plan with human-written descriptions in A-ROBIN. Three metrics evaluate the generated description,



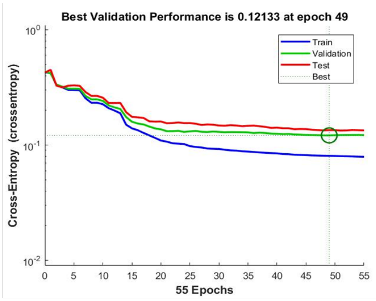
Output class

259	2	2	5	17
9	206	7	2	18
4	4	121	1	43
1	2	2	138	9
2	4	9	7	75

Target class

Training (%)	Testing (%)	Validation (%)
84.2	87.7	82.8

(a) Experimental results for BoD



Output Class

260	1	3	2	4
5	231	5	6	9
4	4	125	2	30
2	4	1	131	6
3	4	12	4	91

Target Class

Training (%)	Testing (%)	Validation (%)
88.3	81.3	85.2

(b) Experimental results for LOFD

Figure 5.17 : Performance analysis of multi-layered perceptron by BoD and LOFD.

Table 5.5 : Performance analysis of text generation algorithm using BLEU score.

BLEU-1	BLEU-2	BLEU-3	BLEU-4
0.6418	0.4673	0.3448	0.2103

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) proposed in, Lin [2004], Bilingual Evaluation Understudy (BLEU) proposed in, Papineni *et al.* [2002] and Metric for Evaluation of Translation with Explicit Ordering (METEOR) proposed in, Denkowski and Lavie [2011]. The textual description generated by our framework is then compared with the descriptions in A-ROBIN to evaluate their agreement with human-written descriptions. Table 5.4 depicts the average recall, average precision, and F score for ROUGE-1, ROUGE-2, ROUGE-3. As the value of n in n-gram comparison increasing, the ROUGE precision score decreases, which is also clear from Tab. 5.4. Table. 5.5 depicts the BLEU score and Tab. 5.6 METEOR score for the description generated, which demonstrates a high correlation with human judgment.

ROUGE

ROUGE is a set of metrics designed to evaluate the text summaries. The generated summary can be evaluated against a set of reference summaries. In our work, we have compared the generated descriptions with available human written descriptions using n-gram ROUGE by the following equation.

Table 5.6 : Performance analysis of description synthesis using METEOR score.

Average Recall	Average Precision	F1	F mean	Final Score
0.555	0.218	0.313	0.450	0.184

$$\frac{\sum_{S\in\{RS\}}\sum_{gram-n\in S} Count_m(gram-n)}{\sum_{S\in\{RS\}}\sum_{gram-n\in S} Count(gram-n)} \tag{5.5}$$

where, RS stands for reference summaries, n stands for the length of the n-gram, $gram-n$, and $Count_m(gram-n)$ is the maximum number of n-grams co-occurring in the candidate summary and the set of reference summaries. In Tab. 5.4, comparison with three types of ROUGE-n is shown, ROUGE-1, ROUGE-2 and ROUGE-3. It can be seen that average recall is decreasing with increasing n-gram in ROUGE. This behavior is natural as ROUGE-1 compares on the uni-gram basis in the candidate reference corpus, which is the word matching. ROUGE-2 compares on a bi-gram basis, taking a set of two words at a time. However, ROUGE-3 compares on a tri-gram basis, which is by considering 3 words at a time. Since ROUGE-1, ROUGE-2, and ROUGE-3 use uni-gram, bi-gram, and tri-gram comparisons, respectively, the decreasing nature of average precision is natural. Machine-generated descriptions have a fixed pattern for words to be used and the information to be displayed. However, human-written descriptions can have any sequence and use of words and phrases.

BLEU

BLEU metric analyses the co-occurrences of n-grams between a machine translation and a human-written sentence. The more matches, the better is the candidate translation is. The score ranges from 0 to 1, where 0 is the worst score, and 1 is the perfect match. In Tab. 5.5, we have given 4 types of BLEU score, for 4 values of n-gram. They first compute n-gram modified precision score (p_n) by the following equation,

$$p_n = \frac{\sum_{C\in\{Cand\}}\sum_{gram-n\in C} Count_{clip}(gram-n)}{\sum_{C'\in\{Cand\}}\sum_{gram-n'\in C'} Count(gram-n')} \tag{5.6}$$

Where, $Count_{clip}$ limits the number of times a n-gram to be considered in a candidate ($Cand$) string. Then they computer the geometric mean of the modified precision (p_n) using n-gram upto length N and weights W_n which sums up to 1. A brevity penalty(BP) is used for longer candidate summaries and for spurious words in it, which is defined by the following equation:

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{\frac{1-r}{c}}, & c \le r \end{cases} \tag{5.7}$$

Where c is the length of candidate summary and r is the length of reference summary. Then BLEU score for corpus level given equal weights to all n-grams is evaluated by the following equation:

$$BLEU = BP.exp^{\sum_{i=1}^{N} W_n log(p_n)} \tag{5.8}$$

Here W_n is the equally distributed weight in n-grams. E.g., in case of BLEU-4, the weights used are $\{(0.25),(0.25),(0.25),(0.25)\}$. The proposed dataset perform well on BLEU score as shown in Tab. 5.5.

METEOR

METEOR is a metric used for evaluating machine-generated summaries with human written summaries by checking the goodness of the order of words in both. METEOR score combines precision, recall and fragmentation (alignment) in the sentences. It is a harmonic mean of the uni-gram precision and uni-gram recall given an alignment, and calculated as:

$$PN = \frac{1}{2}\left(\frac{no\,of\ chunks}{matched\ uni-grams}\right) \tag{5.9}$$

$$METEOR = \frac{10PR}{R+9P}(1-PN) \tag{5.10}$$

Where PN is the penalty imposed on the basis of larger number of chunks, P is the uni-gram precision, R is the uni-gram recall and METEOR is the final score obtained by multiplying the harmonic mean of uni-gram precision and uni-gram recall with the penalty imposed. Table. 5.6 shows the METEOR score we have obtained while experimenting on A-ROBIN dataset. It can be said that the generated descriptions are very close to the human written descriptions. Also, it is clear that the descriptions collected in the A-ROBIN dataset are grammatically correct and close to the descriptions generated by the proximity-based grammar model.

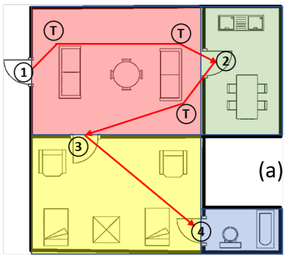
5.6 QUALITATIVE RESULTS

In this section, we describe the qualitative results. These result shows the generated descriptions for samples from A-ROBIN dataset, along with the navigation information in the narrative form.

5.6.1 Examples of description synthesis

In SUGAMAN, rooms are labelled into one of the 5 classes using the trained model. Room annotations and semantic information are stored in an XML file, which is parsed for description synthesis. Figure 5.18 presents the resultant description for 3 floor plan images. The description is formatted in the following way for ease of reading, (i) the first word of the first sentence about any room is in **bold** face, (ii) in the floor plan image, every room is highlighted with a different color and the same color is used to highlight the room name in the first sentence about the room, (iii) in the floor plan image, the turning points are marked with 'T' and sequence of traversal of doors are marked with their respective numbers. There are two types of descriptions synthesized for a given floor plan. The first kind of description is named as General description (GD), which contains information like name, area, global position in the floor plan, relative position of decors, and neighboring rooms in terms of its accessibility by a door is described for each room in the final output description, along with a room having a door opening to outside of the house is also described. The other one is Navigation description (NV), which contains navigation information from room to room avoiding obstacles. If a room has only one door, it is a dead end. Hence the navigating person will turn back.

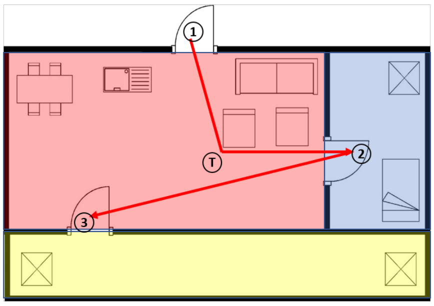
Figure. 5.18(a),(b),(c) are examples where the descriptions are successfully generated for the floor plan images. For example, in Fig. 5.18(a), the GD correctly describes the number of rooms, their connectivity count, and the decors' arrangements inside each room. On the other



(a)

GD:. In this architectural floor plan, there are 4 rooms. **There** is an ENTRY. It has an area of 39.5485 sq. ft. It has a door opening to the outside. Its neighboring rooms are BEDROOM, KITCHEN. It is located in the North side of the house. There is 1 large sofa at the West side of the room, 1 round table at the East side of the room, 1 large sofa at the East side of the room. **There** is a BEDROOM. It has an area of 34.64195 sq. ft. Its neighboring rooms are ENTRY, BATHROOM. It is located in the South West side of the house. There is 1 armchair at the West side of the room, 1 bed at the West side of the room, 1 coffee table at the South West side of the room, 1 bed at the South East side of the room, 1 armchair at the East side of the room. **There** is a BATHROOM. It has an area of 5.2914 sq. ft. Its neighboring room is BEDROOM. It is located in the South East side of the house. There is 1 large sink at the South West side of the room, 1 tub at the East side of the room. **There** is a KITCHEN. It has an area of 16.80245 sq. ft. Its neighboring room is ENTRY. It is located in the North East side of the house. There is 1 twin sink at the North side of the room, 1 dining table at the South side of the room.

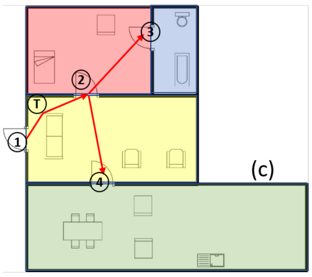
NV: Go 7 steps in North East Direction. Go 65 steps in East direction. Go 6 steps in South East Direction. There is a door and a room. You have to turn back. Go 9 steps in South West Direction. There is a door and a room. Go 70 steps in South East Direction. There is a door and a room.



GD: In this architecture floor plan, there are 3 rooms. **There** is an ENTRY. It has an area of 89.07 sq. units. It has a door opening to outside. Its neighbouring rooms are BEDROOM and HALL. It is located in the North side of the house. There is 1 dining table at the north east side of the room, 1 sink at the north side of the room, 1 large sofa at the north west side of the room, 2 small sofa at the west side of the room. **There** is a BEDROOM. It has an area of 42.76 sq. units. Its neighboring room is ENTRY. It is located in the west side of the house. There is 1 coffee table at the north side of the room, 1 bed at the south side of the room. **There** is a HALL. It has an area of 35.20 sq. units. Its neighboring room is ENTRY. It is located at the south side of the house. There is 1 coffee table at the west side of the room, 1 coffee table at east side of the room.

NV: Go 24 steps in South Direction. Go 27 steps in East Direction. There is a door and a room. You have to turn back. Go 51 steps in South West Direction. There is a door and a room.

(b)



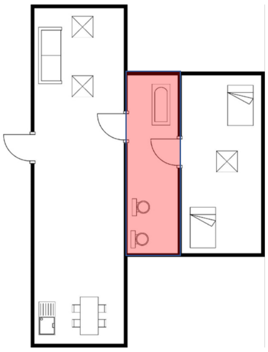
(c)

GD: In this architectural floor plan, there are 4 rooms. **There** is a ENTRY. It has an area of 49.66 sq. units. It has a door opening to outside of the house. Its neighbouring rooms are BEDROOM and KITCHEN. It is located in the west side of the house. There is 1 large sofa in the west side of the room, 1 armchair in the south of the room, 1 armchair in the south east of the room. **There** is a BEDROOM. It has an area of 37.45 sq. units. Its neighbouring rooms are BATHROOM and ENTRY. It is located in the north west side of the house. There is 1 bed in the south west side of the room, 1 small sofa in the north side of the room. **There** is a BATHROOM. It has an area of 20.34 sq. units. Its neighbouring room is BEDROOM. It is located in the north east side of the house. There is 1 tub in the south of the room, 1 large sink in the north of the room. **There** is a KITCHEN. It has an area of 81.22sq. units. Its neighbouring room is ENTRY. It is located in the south side of the house. There is 1 dining table in the west side of the room, 1 small sofa in the south side of the room, 1 small sofa in the north side of the room, 1 sink in the south west side of the room.

NV: Go 5 steps in North East Direction. Go 7 steps in North East Direction. There is a door and a room. Go 41 steps in North East Direction. There is a door and a room. You have to turn back. Go 38 steps in the South Direction. There is a door and a room.

Figure 5.18 : Generated descriptions for three floor plan images from A-ROBIN dataset.

hand, the NV part of the description guides the user to navigate each room, starting from the entry. It can be observed that starting from the entry door (labeled as 1), the first obstacle to go to the kitchen (as per the DFS navigation) is a sofa. Hence, the user has to take a turn (marked as "T") and then proceed. The directional information are obtained from the nonuniform binning technique discussed in Sec. 5.4.3. In Fig. 5.18(a), also the significance of "backtracking" can be understood. Once someone reaches the kitchen, then the next room to visit is the bedroom. Since there is no direct connection (as per Alg. 3, line 6, $AM_D(c_r, n_r) \neq 1$). Thus, the current room (c_r) is changed from kitchen to entry. However, the navigation door should now be availed 2, even though the current room is the entry. The switching of the door index, as per door-to-door connectivity, is taken care by the line 31-34 of Alg. 3. We have tested Alg. 3 for various configurations and achieved correct results. Figure. 5.19 shows a failure case. The reason behind this failure is that the decor recognition framework has miss-classified the large sink as sink and tub as a large sofa. The NV part of this example is not shown here as the GD is incorrect. As a result, the bathroom is labeled as the kitchen. The limitations of the proposed framework are discussed next.



GD: In this architecture floor plan, there are 3 rooms. **There** is an ENTRY. It has an area of 91.73 sq. units. It has a door opening to outside. Its neighbouring room is KITCHEN. It is located in the west side of the house. There is 1 dining table at the south side of the room, 1 sink at the south west side of the room, 1 large sofa at the north west side of the room, 2 coffee table at the north side of the room. There is a KITCHEN. It has an area of 32.16 sq. units. Its neighboring rooms are ENTRY and BEDROOM. It is located in the north side of the house. **There** is 1 large sofa at the north west side of the room, 2 sink at the south east of the room. There is a BEDROOM. It has an area of 35.20 sq. units. Its neighboring room is KITCHEN. It is located at the east side of the house. There is 1 coffee table at the west side of the room, 1 bed at north east side of the room, 1 bed at the south west side of the room.

Figure 5.19 : Generated descriptions for three floor plan images from A-ROBIN dataset.

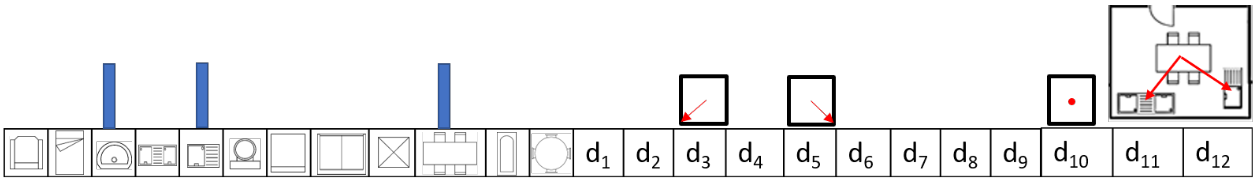


Figure 5.20 : An illustration of incorrect creation of the LOFD feature.

5.6.2 Limitations of proposed BoD and LOFD features

The proposed features highly depend on the decor classification. An inaccurate decor classification algorithm affects the accuracy of BoD and LOFD features. Figure 5.20 depicts an incorrect creation of LOFD feature. Due to the morphological operation and joining of the nearby blob, the twin sink is classified as a large sink. As a result, the classification model confused this room image and labeled the kitchen as a bathroom. Hence, we require a highly accurate decor classification algorithm. Reducing its dependency on decor classification or a better decor classification algorithm will further improve its accuracy. Moreover, the descriptions collected through Google form at present are scripted. However, there is substantial variability in reality while describing a floor plan like incomplete sentences, out-of sequence, room description, etc. A learning model, which can collate all this information and generates a single description to be used for experiment purposes forms a unique scope for future work.

5.7 COMPARATIVE ANALYSIS

5.7.1 Decor identification

Table 5.7 shows the comparison between existing techniques proposed in D. Sharma, C. Chattopadhyay and G. Harit [2016]; Hu [1962]; Ojala *et al.* [2002] and ours. All the decor items in each room image is classified in one of the 12 decor categories. The maximum accuracy obtained for a particular symbol is depicted in **bold** face numbers. Performance between ours and D. Sharma, C. Chattopadhyay and G. Harit [2016] is comparable, except for the large sink and tub, where

Table 5.7 : Comparison of recognition accuracy (%) of ours vs. others.

Symbol	D. Sharma, C. Chattopadhyay and G. Harit [2016]	Hu [1962]	Ojala *et al.* [2002]	Ours
Bed	**99.01**	80.39	44.60	98.5
Arm Chair	**100**	77.77	63.88	**100**
Coffee Table	99.15	0.004	11.44	**99.57**
Dining Table	**98.76**	77.77	66.66	**98.76**
Small Sofa	**100**	96.77	0.00	**100**
Large Sofa	98.06	0.009	65.44	**99.35**
Small Sink	83.33	0.00	72.22	**88.88**
Twin Sink	**95.23**	0.00	71.42	**95.23**
Sink	**100**	0.00	63.73	**100**
Large Sink	55.69	0.00	51.89	**67.08**
Tub	61.16	74.75	61.16	**97.08**
Round table	0.00	0.00	**100**	82.35

SUGAMAN is superior. The technique proposed in D. Sharma, C. Chattopadhyay and G. Harit [2016] could not detect the round table, while SUGAMAN achieves 82.35% accuracy. In addition, for the technique proposed in Hu [1962], recognition accuracy is very low. However, with our technique implemented using the LBP feature, the round table could be recognized with 100% accuracy, while comparatively much lower accuracy for others, which lowers down the overall system's mean accuracy.

5.7.2 Access navigation

Table 5.8 shows the comparative analysis of the various state-of-the-art indoor navigation approaches with ours. We have compared the existing approaches with ours in terms of the problems in navigation they have addressed. In the work Xu *et al.* [2016], the authors have addressed the problem of obstacle avoidance same as SUGAMAN. However, proposed work in Liu and Zlatanova [2011] and Zlatanova *et al.* [2013] have not dealt with this issue. Moreover, the technique proposed in Xu *et al.* [2016] requires creating a network that involves checking all the edges of the triangle created (Delaunay triangulation), which increases the algorithm's complexity. Taking the problem of backtracking of the path when a dead end is encountered, SUGAMAN has dealt with it, while others don't provide a solution for the same. Additionally, the proposed work in Xu *et al.* [2016] requires an empirical calculation of a threshold value for Delaunay triangulation, which is not required in others and SUGAMAN as well. The proposed algorithm is better in terms of problems addressed and complexity. It provides the shortest door to door path for indoor navigation while avoiding obstacles and without any requirement of manual intervention.

Problem / Method	Obstacle Avoidance	Backtracking	Threshold Calculation
Ours	Yes	Yes	Not Required
Xu *et al.* [2016]	Yes	No	Required
Liu and Zlatanova [2011]	No	No	Not Required
Zlatanova *et al.* [2013]	No	No	Not Required

Table 5.8 : Comparative analysis of various path finding approaches

5.8 SUMMARY

This chapter's primary objective is to generate description and navigation information from floor plan images for the visually impaired. We have proposed novel features, BoD and LOFD, for automatic room label learning. A proximity-based grammar model is also proposed used to synthesize the description. The proposed algorithm also generates navigation information in the form of narration. We have also offered a novel description dataset, A-ROBIN, and made it publicly available for the DAR community. In the next chapter, two advanced deep learning models, Description Synthesis from Image Cues (DSIC) and Transformer-Based Description Generation (TBDG) are proposed for end-to-end textual description generation from floor plan images utilizing the paragraph annotations proposed in BRIDGE dataset.