5

Requirement driven feature integration for floor plan retrieval

In Chapter 4, a deep learning architecture for floor plan retrieval is proposed which was the first attempt of applying deep learning to the task of floor plan retrieval. A naive approach was introduced to learn a deep learning feature representation and study how the individual deep feature layers affect the performance of the system during retrieval. In the deep learning framework, a combination of convolutional, pooling, normalization, ReLU and fully connected layers in the Convolutional Neural Network (CNN) framework were used to obtain an effective feature representation from floor plan images. Quantitatively this framework gave encouraging results with a mean average precision of 0.56. However, certain key issues were observed during the retrieval process:

- The CNN framework is a simple network that mainly distinguishes by the global shape of a floor plan. It would fail while doing fine grain categorization of furniture components in the layouts. It would also not perform well if two layouts from different classes have almost similar shapes.

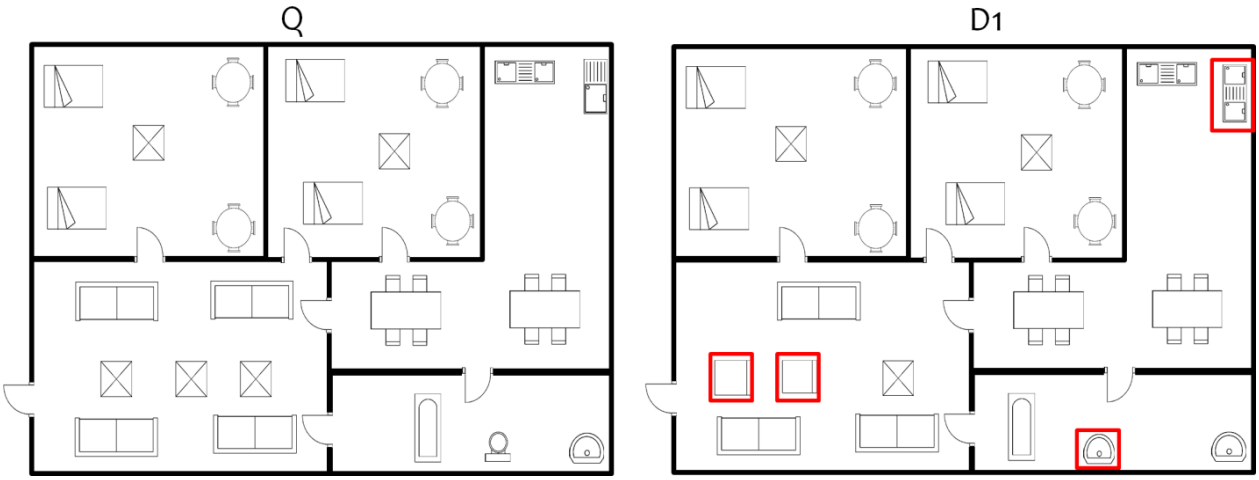


Figure 5.1. : Q depicts the query floor plan and D1 depicts a resultant retrieved floor plan from the database of floor plans. The objects enclosed in red boxes in D1 are furniture components with similar shape as the components in Q, but different types.

An example in support of the above argument is given in Fig. 5.1. Here, the shape of the square table and the small sofa (marked in red in D1) are quite similar, therefore, while retrieval through the deep framework, query floor plan Q and database floor plan D1 are deemed similar and D1 is ranked higher in the retrieved results, in spite of their dissimilarity. Moreover, the layout sub-categories in the ROBIN dataset are mainly proposed to be same in the global layout shape, which justifies the higher mean average precision value using DANIEL framework, but does not result into efficient qualitative results.

The other trait where the deep framework lacks is that there is no provision of a weighted combination of the features, which does not let the user give preferences to individual characteristics, according to the requirement. Choices cannot be set using the proposed deep framework, as the layers extract and combine the features implicitly without the provision of manipulating the weights by the user.

The hand-crafted approach, with non-deep features proposed in Chapter 3 was the baseline, and first of its kind framework for retrieval in floor plans. However, it had limitations like (i) the matching was done in two phase, i.e. layout matching followed by decor matching, and thus not a unified/ combined approach, (ii) a few key features were overlooked that define a buyer's requirement, e.g., room dimensions were not considered. In this Chapter, a framework for "fine-grained" retrieval of floor plan images from the repository is proposed.

The key contributions of this work are:

1. A novel end-to-end framework for extracting high level semantic features like the area and room-wise decor arrangement for the task of fine-grained retrieval

2. A technique to perform feature fusion to aggregate high-level semantic features to retrieve floor plans.

3. Extensive experimentation on a large dataset to demonstrate robustness and scalability. The technique proposed in this Chapter can find application in online property sale/rent scenarios where the buyer has preferred features related to the room semantics.

The Chapter is organized as follows: Sec. 5.1 gives a brief overview of the proposed framework. The fine-grained feature extraction process is presented in Sec. 5.2. The matching algorithm is described in Sec. 5.3. Details of the experimental findings are discussed in Sec. 5.4, followed by critical observations in Sec. 5.5. Section 5.6 summarizes this Chapter and leads the thesis towards the final approach.

5.1 BRIEF OVERVIEW

Figure 5.2 depicts the complete framework of the proposed technique. The entire framework has two phases: (i) feature extraction (solid line) and (ii) retrieval (dotted line). Structural analysis of a floor plan is performed. For this purpose, the layouts are segmented into rooms by identifying the walls separating all the rooms. Then, in a floor plan, the high-level semantic features like adjacencies between rooms, carpet area of rooms [Bannister, 2010] and also the number and type of furniture inside each room, termed as the Furniture Composition Record are extracted. During the retrieval phase, these features are extracted from a query floor plan and a weighted sum is computed of all the features extracted to generate a matching score for a particular layout as compared to another layout in the repository. By default, all the weights while matching are set to 1 to favor an equal sum of all the high-level semantic features. This score then serves as the parameter for fine-grained retrieval of similar floor plans. Additionally, any feature can be given a higher weight as compared to the other features during the matching process which further, aids in preferring a feature more during retrieval. Among the publicly available floor plan datasets, the *ROBIN* [Sharma *et al.*, 2017] dataset is chosen to validate the technique shown in this Chapter.

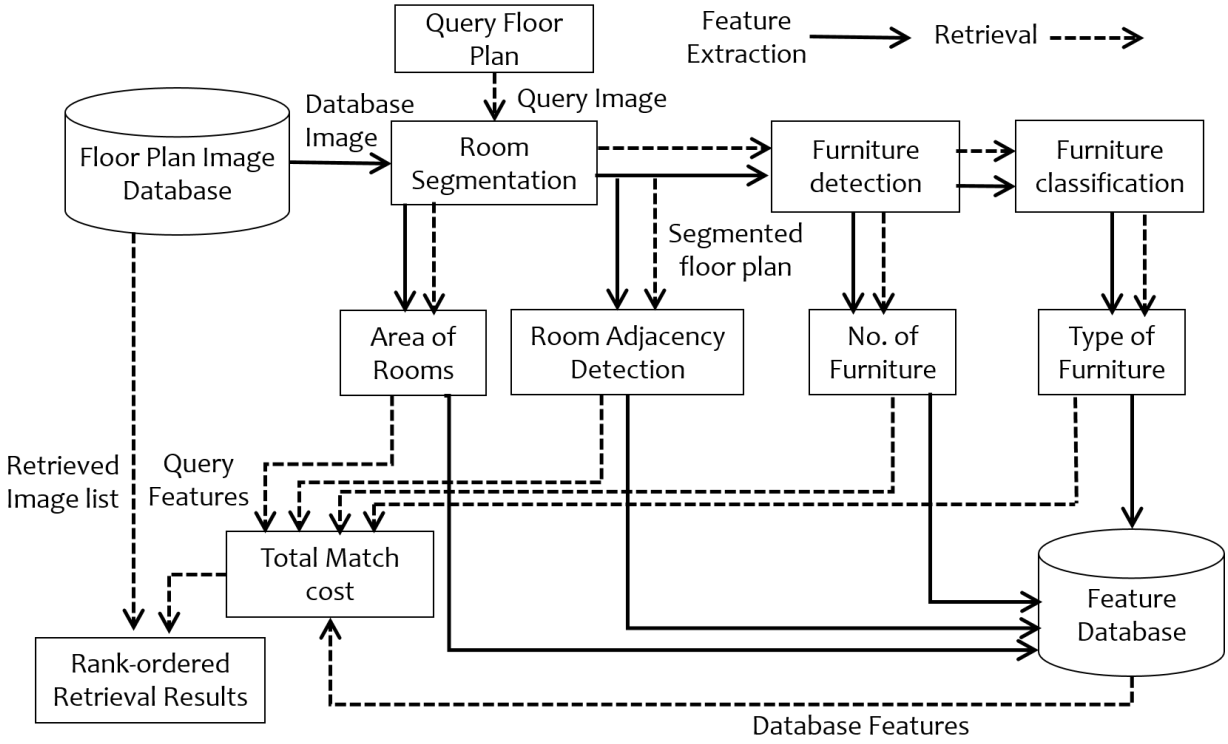


Figure 5.2. : Basic framework depicting extraction of high-level semantic features and weighted fusion for retrieval of floor plans.

5.2 HIGH LEVEL SEMANTIC FEATURES

As a preprocessing step, a morphological closing operation is carried out to segregate the layout boundary from furniture present inside the layout. The doors inside the plan are present in various orientations. They are detected and the gaps at their locations are closed to construct definite boundaries for the rooms. This recognizes the walls in the layout to represent the room structures and thus, delineates the rooms within the layout from each other. Next, connected component analysis using 8-neighborhood criteria is performed to label each room in the layout separately. The rooms in the layout are numbered to get a consistent ordering by their distance from a reference point situated at the top-left-most end of the floor plan image. A diagram showing the same is presented in Fig. 5.4.

Euclidean distance between the reference point taken to be as the top-leftmost point of the floor plan canvas and the centroid of each room is calculated. The closest room to the reference point, i.e., with the minimum distance to the reference point is labeled 1, and other rooms are labeled accordingly. Such an ordering makes the system rotation sensitive. Figure 5.3 (b) and (d) depicts the resultant room layout segmentation and labeling given a floor plan. Such an ordering helps in establishing a better correspondence between rooms while feature extraction and matching. The three features that are being extracted namely, Room Adjacency String, Carpet Area Ratio and Furniture Composition Record are described below.

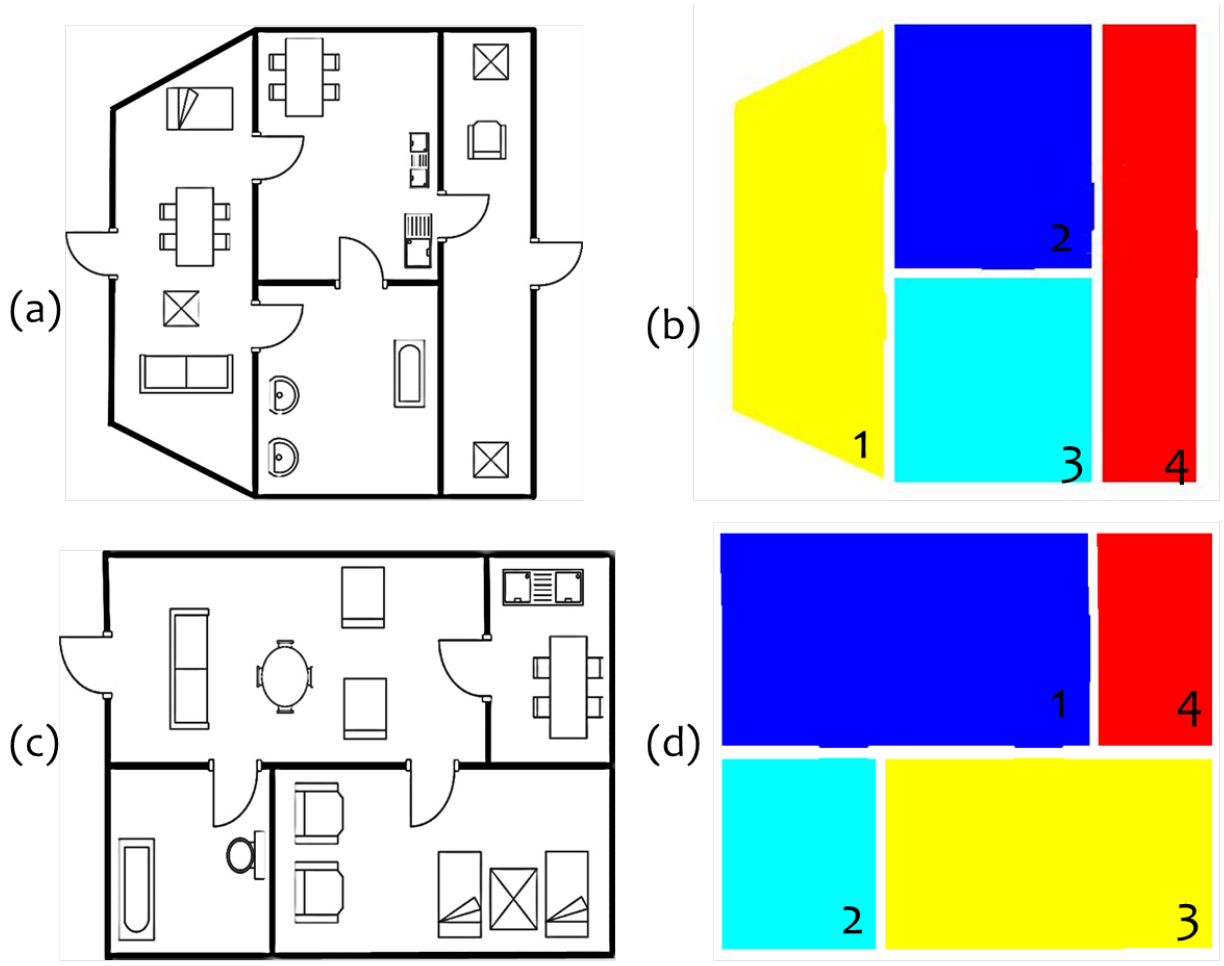


Figure 5.3. : Room level segmentation results on a pair of floor plan images from ROBIN dataset.

5.2.1 Feature 1: Room Adjacency String (RAS)

Room segmenting and labeling is followed by determining the adjacencies between the rooms to mark which rooms share boundaries in a layout using the Room Adjacency String (RAS) feature. This feature can be a preference if buyers are keen to choose the arrangement of rooms in a layout, e.g., *the master bedroom should be near the kids' bedroom.* For this purpose, an adjacency matrix is constructed from the segmented and order-wise labeled layout. For k^{th} floor plan image in the database its topological graph is denoted as $G_k = (V_k, E_k)$, where V_k is the set of vertices also denoting the number of rooms in the layout and E_k is the edge-set as depicted in Fig.5.5 (b). The adjacency matrix, for the graph G_k is denoted as Adj_k which is a $|V_k| \times |V_k|$ matrix, where, $|\cdot|$ denotes cardinality of the set V_k. For each row of the adjacency matrix corresponding to each room, a unique binary string is computed, which represents the adjacencies for that particular room. E.g. the room adjacency of room 2 in the segmented layout in Fig. 5.5 (Example 1) (a) is highlighted in Fig. 5.5 (Example 1) (c). RAS of 1011 depicts Room 2 is adjacent with Rooms 1, 3 and 4.

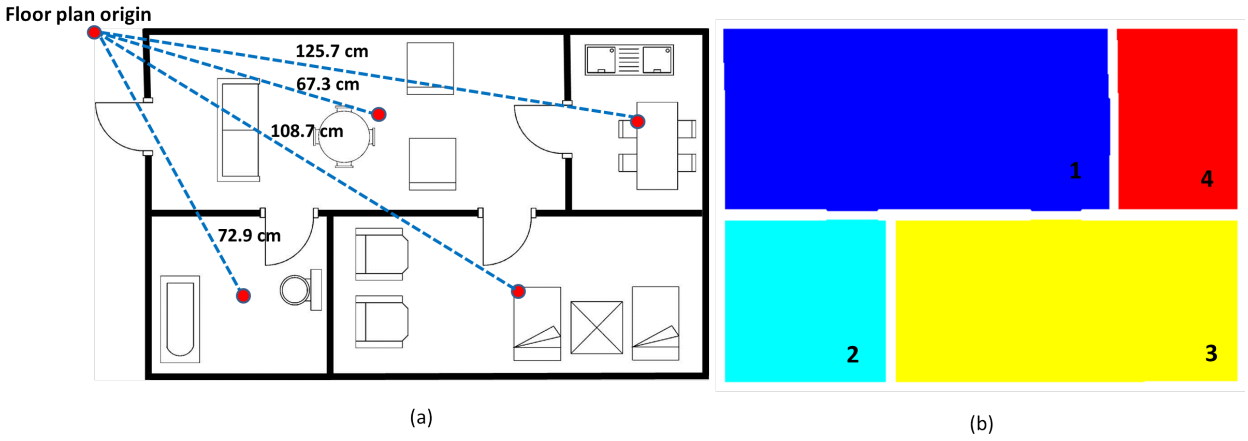


Figure 5.4. : Figure depicting the ordered labeling of rooms after segmentation. (a) Distance is calculated from a leftmost origin point up to all room centroids. Room with the lowest distance from the origin is ordered number 1 and the other rooms are labeled in similar fashion. (b) Subsequent room ordering results.

5.2.2 Feature 2: Carpet Area Ratio (CAR)

CAR is a unique feature that represents the ratios of carpet area enclosed within the walls i.e. the actual area to lay the carpet. This area does not include the thickness of the inner walls. The carpet area (Ar) of a room (L^k) is calculated using the formula for calculation of area of a polygon as specified in [Braden, 1986]:

$$Ar(L^k) = \left| \frac{(x_1y_2 - y_1x_2) + (x_2y_3 - y_2x_3) + \cdots + (x_ny_1 - y_nx_1)}{2} \right| \tag{5.1}$$

where, the pair $(x_i, y_i); \forall i = 1, \cdots, n$ denotes the n vertices of a room, taken in clockwise order, and the operator $|.|$ denotes the absolute value of the function. The motivation behind extracting this feature is the fact that usually users buying layouts have preferences in terms of the size of each room in the floor plan.

CAR ($A(\cdot)$) serves as a unique feature while a user is looking for layouts. The ordered rooms are labeled and the area between their boundaries and their respective ratios is stored which serves as the CAR. The area of a particular room (L^k) is then divided by the total area of the layout (L) to give an idea of the size of the particular room corresponding to the whole layout (refer Eq. 5.2). $Ar(L)$ is also calculated using Eq. 5.1, where (x_i, y_i) are the vertices of the polygon obtained by tracing the global layout of L.

$$A(L^k) = \frac{Ar(L^k)}{Ar(L)} \tag{5.2}$$

Figure 5.6 (b) and (d) show the areas corresponding to the different rooms in the layout shown in Fig. 5.6 (a) and (c) in the form of a pie-chart. Here, the inner entries are the actual areas of the rooms and the outer entries, represent the CAR feature of a room.

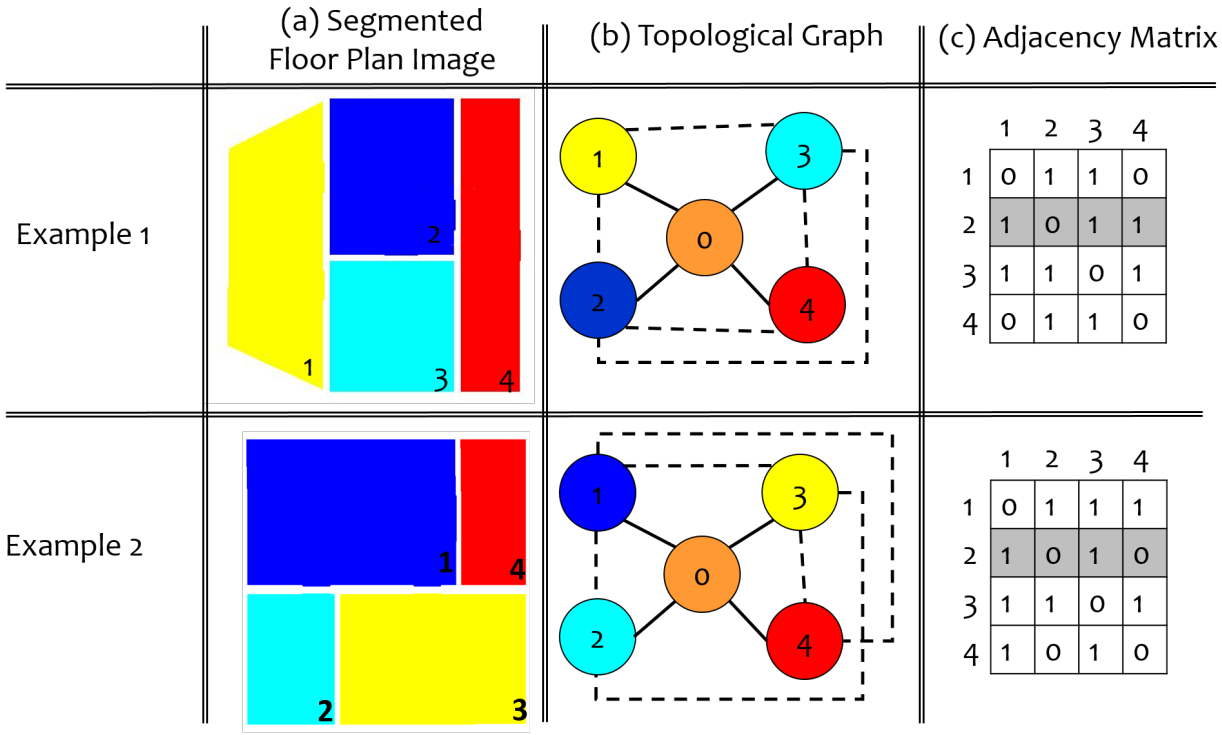


Figure 5.5. : Example of Room Adjacency String (RAS) feature: (a) Segmented rooms (b) Topological Graph (c) Adjacency matrix (Adjacency string corresponding to Room 2 is highlighted).

5.2.3 Feature 3: Furniture Composition Record (FCR)

The third feature that is computed is the Furniture Composition Record (FCR) that comprises of number and type of various furniture inside a particular room. A furniture is localized inside a room according to their position in the layout, by detecting the blobs inside the floor plan image. Further, their count (C) is stored corresponding to each room. This helps to compare the rooms according to the consumers priority of being overcrowded with decor or not. Next, the various furniture are classified by performing a connected component analysis on the detected blobs and taking the area of individual components inside the different furniture symbols. Then a ratio of the largest 3 area components is taken, and a unique signature is obtained for each furniture which can be further used for categorizing them as shown in Fig. 5.7. Algorithm 3 describes each step of the furniture categorization process in detail where the input is a room from a floor plan image without the walls. The set $\mathscr{T}$ is generated as the output of the furniture categorization step. In this step, each blob in a room is categorized and their signature is compared with the already stored furniture template signatures and assigned a type T. The union of the type of all the furniture inside a room then constitute the set $\mathscr{T}$. This set finally contains the type of furniture inside each room.

Furniture type gives a unique signature to a room. For example, a room with a bed, table, armchair etc. would primarily be a bedroom as shown in Fig. 5.8. So, while comparing a pair of layouts, it is worth to compare features of the similar type of rooms, instead of all possible rooms. Hence, categorizing a room based on type of furniture can prove helpful while matching. The feature extraction accuracies of all the 3 features is listed in Tab. 5.1. Accuracy of a particular feature is calculated by taking ground-truth information for each floor plan of the dataset in terms

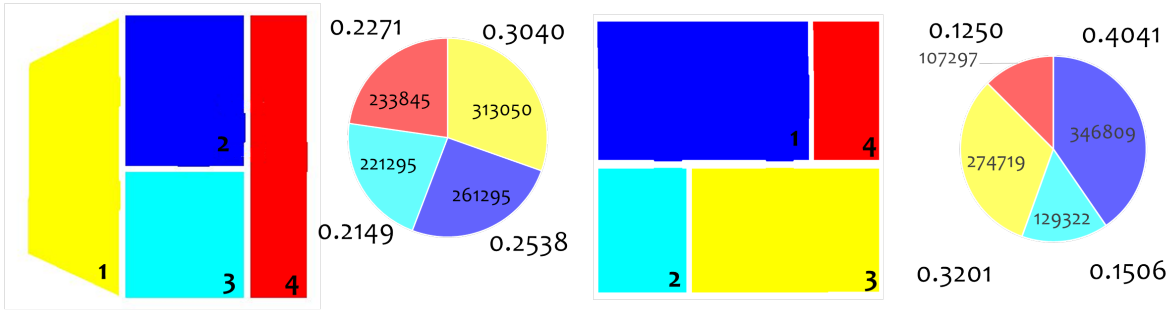


Figure 5.6. : Carpet Area Ratio (CAR) feature: (a), (c) Segmented, ordered layout, while (b), (d) CAR feature corresponding to Example 1 and 2 respectively.

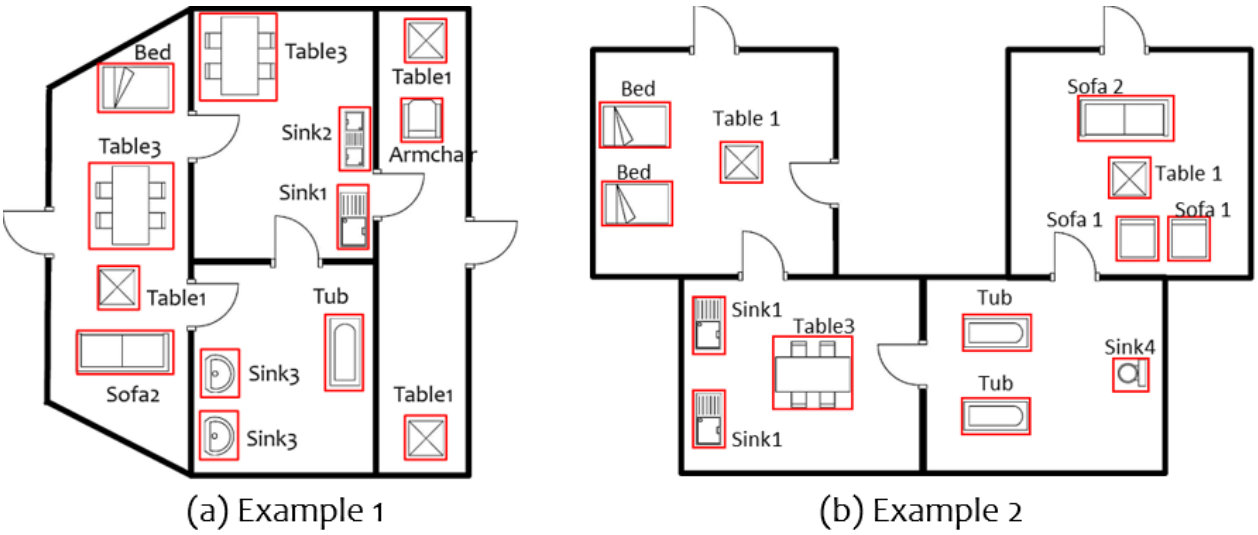


Figure 5.7. : Furniture detection and categorization in each room.

of the adjacencies, areas and furniture composition that each room in a floor plan has and further, taking a ratio of the features extracted from floor plans while running the algorithm and the ground-truth results. It was observed that each feature is extracted to a perfection using the steps mentioned above, except the furniture categorization feature, which performs to an accuracy of 89%. This accuracy while feature extraction helps in identifying the user's requirement efficiently while feature fusion and retrieval.

5.3 FINE-GRAINED MATCHING AND RETRIEVAL

Matching of a pair of floor plans using the extracted features is essential for retrieving similar layouts to the query layout. In the following sections the procedure to calculate matching score corresponding to various high level semantic features is described. Let, L_i^k and L_j^k represent

Table 5.1. : Percentage Accuracy of the extracted high level semantic features

Feature	Extraction Accuracy
Room Segmentation/ Room Adjacency	100%
Carpet Area	100%
Number of Furniture	100%
Type of Furniture	89%

k^{th} room in two different layouts (L), where i and j range from 1 to n, i.e. the total number of floor plans in the dataset, while k ranges from 1 to total number of rooms (V) in a layout. If there is a difference in the number of rooms in the layout then the minimum number of rooms between the layouts is taken for comparison. Algorithm 4 describes the matching procedure in a step-by-step fashion.

5.3.1 RAS Matching Score

Edit distance, represented as $e(\cdot,\cdot)$ [Levenshtein, 1966] is used, as the measure to differentiate between a pair of adjacency strings corresponding to same numbered rooms in two different layouts as described in Sec.5.2.1. As an example room 2 for layout 1 in Fig. 5.5 (Example 1) (b) has RAS 1011 and room 2 for layout 2 in Fig. 5.5 (Example 2) (b) has RAS 1010. In this case, the two adjacency strings differ only at the fourth bit whereas, rest bits are similar. Therefore, edit distance, i.e. the operations required to convert one string to another results to 1. Similarly, in case of room 4 for both the layouts the RAS are 0110 and 1010 respectively, differing in the first

Algorithm 3 Furniture detection and categorization

Input: Room image without outer walls (I), Furniture Template Signatures (F)
Output: Furniture Count (C), Furniture Type Set ($\mathscr{T}$)

C=0, $\mathscr{T}$=$\{\phi\}$
$\mathscr{B}$=Morphological fill operation(I)
$C = |\mathscr{B}|$ ▷ $|\cdot|$: *Cardinality*
for $j = 1$ to C **do**
  $\mathscr{C} = CC(\mathscr{B}_j)$ ▷ CC()=Connected Component
  for $k = 1$ to $|\mathscr{C}|$ **do**
    $\mathscr{A}_k = Area(c_k)$, where $c_k \in \mathscr{C}$
  end for
  $A = Sort_{desc}(\mathscr{A})$
  $S(\mathscr{B}_j) = \{(A_1/A_3), (A_2/A_3), 1)\}$ ▷ $S(\cdot)$: Signature
  for $k = 1$ to $|F|$ **do**
    if $S(\mathscr{B}_j) == F_k$ **then**
      $T(\mathscr{B}_j) = T(F_k)$ ▷ $T(\cdot)$: Type of Furniture
    end if
  end for
  $\mathscr{T} = \{\mathscr{T} \cup (T(\mathscr{B}_j)\}$
end for

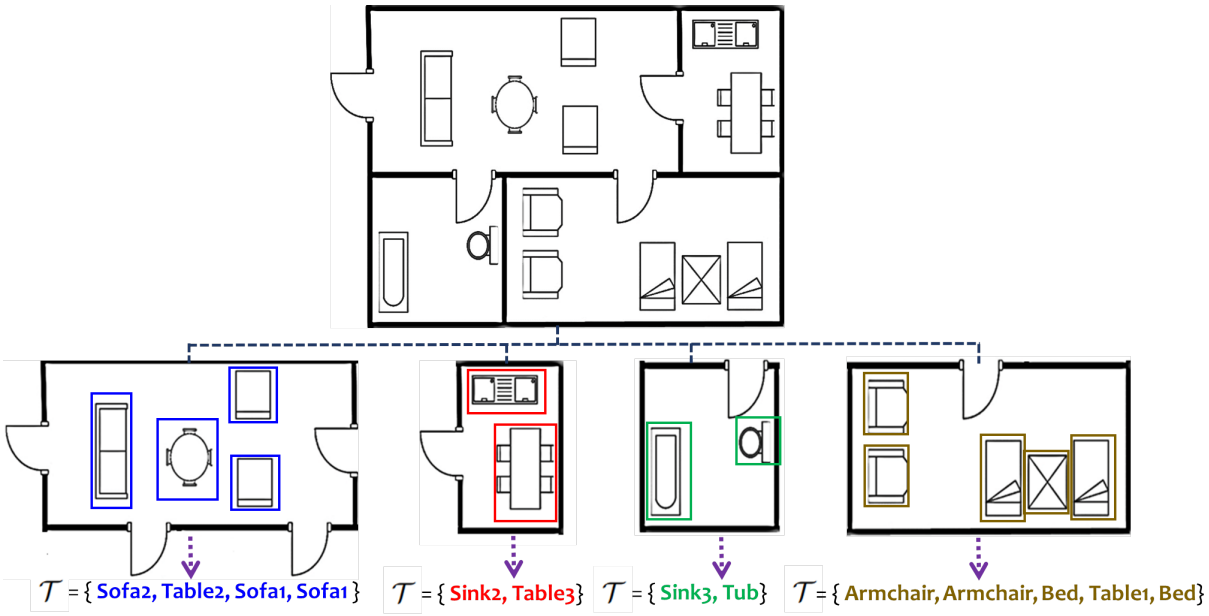


Figure 5.8. : Furniture detection and categorization in each room. Illustration underneath each room gives the array $\mathcal{T}$ containing the type of furniture in each room.

and second bit, resulting in an edit distance of 2. The RAS matching score (ρ) is computed as:

$$\rho(i,j,k) = \frac{e(L_i^k, L_j^k)}{max(V_i, V_j)} \tag{5.3}$$

Here the matrix ρ corresponds to the difference in room adjacencies between all similar numbered rooms in two layouts. Thus, higher ρ value suggests dissimilarities in adjacencies between two rooms. In a similar manner adjacency string matching of the rest of the rooms of a layout as compared to another layout is done. For normalizing this particular score in the range [0,1], it is divided by the factor equal to the maximum number of rooms between layout i and j, denoted as V_i and V_j. The idea behind normalizing in this manner is, that the maximum edit distance in the adjacency string can be obtained only if one of the strings is completely dissimilar from another, and also has a different length as compared to another. In that case the maximum operations taken to convert one string to the other would be equal to the maximum length between both the strings.

5.3.2 CAR Matching Score

The next feature for matching between two layouts is Carpet Area Ratio, $A(\cdot)$ of a particular room in both the layouts as described in Sec. 5.2.2. The CAR matching score (ψ) is computed as:

$$\psi(i,j,k) = A(L_i^k) - A(L_j^k) \tag{5.4}$$

Here the matrix ψ corresponds to the difference in Carpet Area Ratios between all similar numbered rooms in two layouts. The feature area ratio is already normalized in the range [0,1].

Algorithm 4 Fine grained retrieval using high level semantics

Input: Query Image (q), Database Image (d)
Output: Matching Score ($M(q,d)$)

for all d=1 to n **do**
  for all k=1 to min(V_q, V_d) **do**
    $\rho(q,d,k) = e(L_q^k, L_d^k)$ ▷ $e(\cdot)$: edit distance
    $\psi(q,d,k) = A(L_q^k) - A(L_d^k)$ ▷ $A(\cdot)$: Area
    $\phi(q,d,k) = C(L_q^k) - C(L_d^k)$ ▷ $C(\cdot)$:furn-count
    $\theta(q,d,k) = S(\mathscr{T}(L_q^k), \mathscr{T}(L_q^k))$ ▷ $\mathscr{T}(\cdot)$: furn-type
    $Normalize(\rho, \psi, \phi, \theta)$
  end for
  $M(q,d) = \sum_k(\rho(q,d,k)) + \sum_k(\psi(q,d,k)) + \sum_k(\phi(q,d,k)) + \sum_k(\theta(q,d,k))$
  $M(q,d) = M(q,d)/4$
Return $M(q,d)$
end for

Therefore, no post normalization required. A low value of ψ suggests high similarity in area feature of two rooms.

5.3.3 Furniture Composition Record Matching Score

The room decor component has two sub-components, namely the number of furnitures inside two same numbered rooms and type of furnitures inside the rooms.

Number of furnitures

Number of furnitures, $C(\cdot)$ inside a particular room is computed using the technique discussed in Sec. 5.2.3 and corresponding number of furnitures score, ϕ, is calculated as:

$$\phi(i,j,k) = \frac{C(L_i^k) - C(L_j^k)}{max(C(L_i^k), C(L_i^k))} \tag{5.5}$$

Here the matrix ϕ corresponds to the difference in number of furnitures between all similar numbered rooms in two layouts. This helps us compare the rooms according to the consumers priority of being overcrowded with decor or not. The difference of the number of furnitures corresponding to ϕ should be low in both the layouts if they are similar.

Type of furnitures

This particular feature categorizes type of room, for example a bedroom, bathroom or a kitchen based on its furniture arrangement. It helps us distinguish between two same numbered rooms, with different furniture types. Firstly, the classification of the furnitures in each room is done and their classified labels are stored in an array with rows corresponding to each room number and the subsequent column corresponding to the labels of the furniture in that particular room. Motivation behind matching this feature is that a buyer might specify that the third room in the

layout should be a bedroom and hence, while retrieval one should retrieve floor plans with buyer's preferred placement of room in the complete floor plan. Thus, floor plan having a kitchen as the third room should be retrieved later as compared to a floor plan having the third room as the bedroom. In the below mentioned equation, the matrix θ corresponds to the difference in type of furnitures between all similar numbered rooms in two layouts. Using this, the furniture label strings are compared for similar numbered rooms in two layouts and the penalty is incremented if label strings do not match, i.e. if dissimilar furniture is found between both rooms. The more similar the two rooms are in their furniture types the lower would be the value of θ. The type of furniture in a particular layout is represented as $T(\cdot)$, string comparison is represented as $S(\cdot,\cdot)$ and the type of furniture score is represented as θ, and is computed as follows.

$$\theta(i,j,k) = \frac{S(\mathscr{T}(L_i^k), \mathscr{T}(L_j^k)}{max(C(L_i^k), C(L_i^k))} \tag{5.6}$$

5.3.4 Cumulative Match Score (CMS)

After computing the various feature matching scores for individual rooms in a query layout, CMS between a pair of layouts is obtained by adding the feature scores corresponding to all the common rooms present in both the layouts. CMS for a feature (ρ, ψ, ϕ, θ in our case) F is denoted as F^+ and is computed as:

$$F^+(i,j) = \frac{\sum_{k=1}^{min(V_i,V_j)} F(i,j,k)}{min(V_i,V_j)} \tag{5.7}$$

Using this particular equation the corresponding cumulative feature scores ρ^+, ψ^+, ϕ^+ and θ^+ are obtained. Thus, finally, a matching score for a complete layout is obtained by taking a sum of individual features for all the rooms.

5.3.5 Total Match Score

Total Match Score, M for a pair (query and database) of layouts is calculated using the weighted sum of four extracted feature scores namely RAS, CAR score, number and type of furniture score (FCR). Weights are assigned to each feature component to investigate the effect of individual feature during retrieval. This allows user to give more preference to a certain feature while retrieval. In such a case, the matching score has additional individual weight coefficients η_1, η_2, η_3 and η_4 all in the range $[0,1]$, where the default value of the coefficients is 1. In this case, the Match Score M is obtained, as shown in the equation below:

$$M(i,j) = 1 - \frac{\rho^+(i,j) + \psi^+(i,j) + \phi^+(i,j) + \theta^+(i,j)}{4} \tag{5.8}$$

It is to be noted that this score has individual components normalized in the range [0,1]. Also, the Total match score is divided by a factor 4 to be finally normalized in the range [0,1].

As discussed before, the weight coefficients can be incremented and decremented to set preferences to some features while diluting the effect of others, by tuning the values of η_1, η_2, η_3

and η_4 in the equation mentioned below :

$$M(i,j) = 1 - \frac{\eta_1 \rho^+(i,j) + \eta_2 \psi^+(i,j) + \eta_3 \phi^+(i,j) + \eta_4 \theta^+(i,j)}{4} \tag{5.9}$$

The next section describes the qualitative and quantitative assessment of the proposed framework.

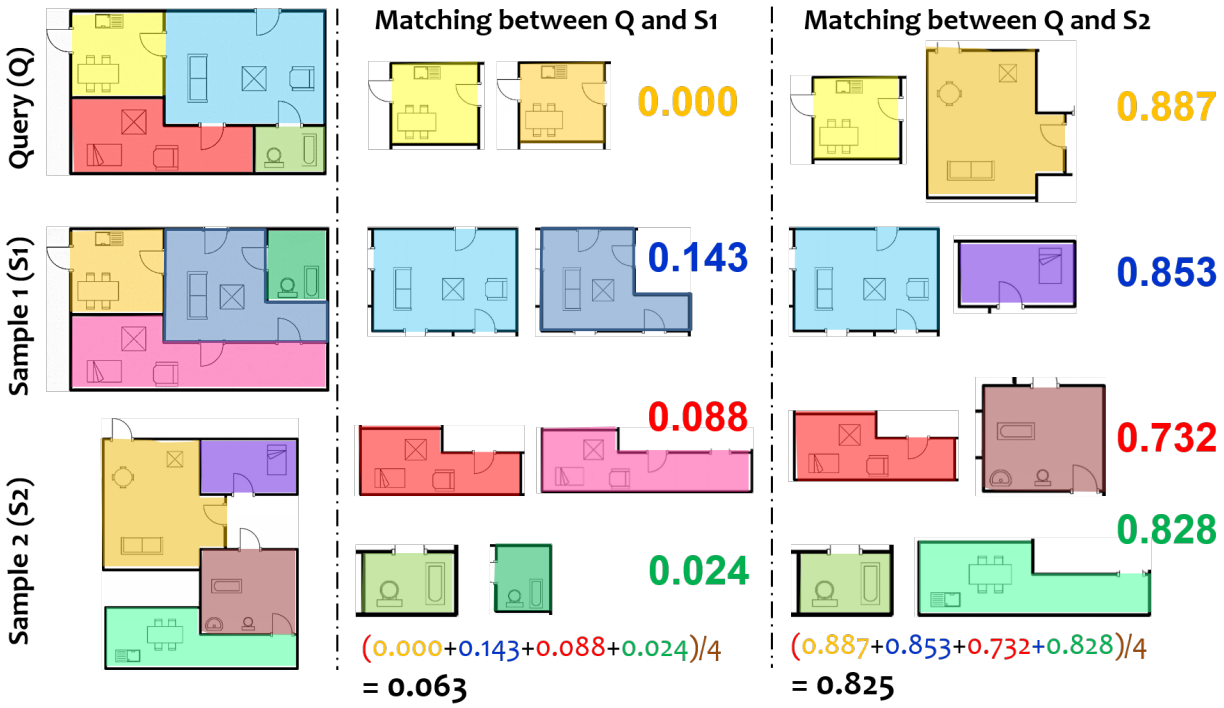


Figure 5.9. : Distance score corresponding to each room of the query layout (Q) and the database layouts (S_1 and S_2). Attributed to the difference in features, lower the score, more similar the floor plans are. Hance, finally the score is subtracted from 1 to make the final greater Matching score to represent the floor plans that match more. Color codes are used to highlight the room correspondence.

5.4 EXPERIMENTS AND RESULTS

The experiments are performed on the *ROBIN* dataset that contains three categories of layouts each with 17 subcategories containing a total of 510 floor plans. The dataset has 11 different categories of furniture inside each layout. The experiments were performed on a machine with Intel Xeon I7 processor with 16 cores.

Details of the matching process are shown in Fig. 5.9. There is a high degree of similarity (in terms of FCR) between the individual rooms of the query floor plan (Q) and that of sample floor plan S1. However, minor differences lie in the area feature (CAR) that can be observed by looking at the second room's ("blue" color) area in both the layouts. The two layouts also differ in their adjacency feature, which can be observed by looking at the fourth room (highlighted in "green"). In the case of Q, the fourth room is adjacent to the third and the second room. On the other hand,

Figure 5.10. : Retrieval results for four different query architectural floor plans taken from the *ROBIN* dataset. Note that the floor plans highlighted in red are incorrectly retrieved as global layout shape differs from query.

in the case of S1, it is adjacent only to the third room. However, database sample floor plan S2, as can be visually seen, differs in many respects as compared to Q. The total scores, as per Eq. 5.9, also justify these differences. For example in the fourth room (color coded in "green") in both Q and S2, all the features, namely the FCR, CAR as well adjacencies differ resulting in a large cost of 0.828. Such observations between Q and database floor plan S2, justify high dissimilarity score attributed to the different global shapes and thus, different CAR, RAS and FCR features of both the layouts. Hence, S2 is ranked later than S1 during retrieval.

5.4.1 Qualitative results

First of all the architectural layouts are segmented into rooms followed by detecting the various types of furniture present in each room. Thus, a semantic analysis of a floor plan is

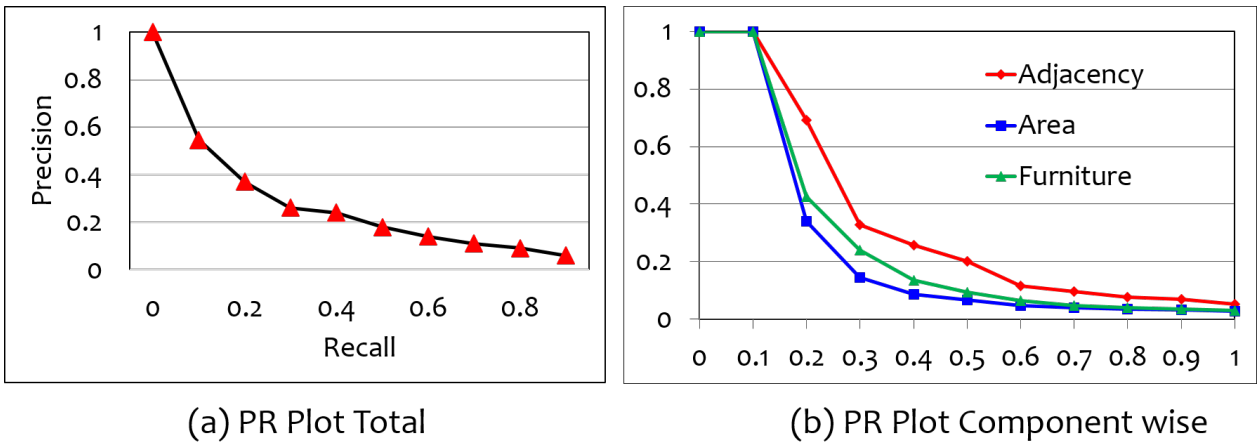
obtained which can be further used for processing. The fine-grained matching techniques proposed in this Chapter helped in the rank ordering of the matched layouts based on the differences in the adjacencies between corresponding rooms, the area of the rooms, number, and type of furniture inside the rooms. The rank-ordered retrieval results for six such queries are shown in Fig. 5.10. The first rank ordered result retrieved from the layout database is the query image itself. This is due to the fact that the match score between the retrieved layout and the query layout is maximum. The subsequent results differ in the scores in the area, adjacencies and decor features inside each room.

As is evident from Fig. 5.10(d) the rank one result is the query layout as mentioned above. Justification of the rank 2 retrieved result can be seen by observing its differences/ similarities with the query floor plan. For example, note that the adjacencies of the first room in the query layout are with room 2 and 4, that is exactly similar to the adjacencies shared by room 1 in rank 2 retrieved result. Thus, on the adjacency parameter, both layouts match completely. Although the first room is slightly bigger area-wise in the query, the furniture type and number are approximately similar in both the layouts. Hence, the match score between the query layout and the retrieved rank 2 layout is high which attributes to the result. An interesting observation is that the rank 3 result retrieves layout from a different sub-category of the dataset as it differs in the global layout shape. It is to be noted that such a retrieval is due to the fact of various common features between the query and rank 3 result which can be analyzed similarly as shown in Fig. 5.9.

5.4.2 Quantitative results

The performance of the system is quantified using a Precision and Recall (PR) metric (see Fig. 5.11). The precision values are averaged over all the query images for a particular recall value. Given a query, retrieved layouts should belong to the same sub-category of layouts as the query, i.e., it should maintain the global shape and also keeping in mind the preference set by the property seeker during requirement driven querying. Thus, if the global shape of the layouts differ then that is considered as an incorrectly retrieved result.

Figure 5.11 (a) refers to the overall PR plot according to the match score (see Eq. 5.8), i.e. giving equal weights for FCR, CAR and RAS scores and retrieving the results. As shown in



(a) PR Plot Total (b) PR Plot Component wise

Figure 5.11. : Precision and Recall (PR) plot corresponding to a) Cumulative sum of all the features with equal weights b) Preferring one feature more than the others while retrieval.

Fig. 5.11 (a), given a query sample the rank 1 result is the query floor plan 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 as compared to the query layout. As discussed in Sec. 5.3.5, while retrieval a user can have a preferred feature. For example, a user might be fixated over preferring area feature over the others while retrieving. The performance of the approach mentioned in this Chapter is analyzed by simultaneously increasing the weight of a feature, while decreasing weights of other features to study the impact of individual high level semantic features during retrieval, thus obtaining PR plot as in Fig. 5.11 (b), corresponding to preferring different features during retrieval. In Fig. 5.11 (b), "red" line corresponds to the PR plot given high weight to adjacency feature during retrieval. Similarly, if the user wants the furniture type and number to be very similar to the query layout then weight of the cost of types of furniture during matching can be increased. The PR plot in Fig. 5.11 (b) with the "green" line depicts the quantitative retrieval results using such an approach. Similarly, the "blue" line in the same plot depicts the results obtained when the area parameter is given more weight. The area under the PR plot is the highest when the adjacency feature ("red" line) is given the highest weight as compared to other features. Because layouts, which are falling in the same category of a global shape generally have similar room adjacencies. Even though their areas and furniture arrangement might differ a little. Thus, the retrieval accuracy is higher when adjacency parameter is favored as is evident from its PR plot.

5.5 DISCUSSION

The proposed approach, efficiently performs all the processing steps given in Fig. 5.2. As observed, the system generally succeeds in retrieving the samples from the same sub-category as the query sample, only faltering at some occasions, where, our proposed FCR and CAR features overpower the RAS feature. For run time analysis of this approach, the time taken (averaged over 510 samples) in seconds at every step was computed. Segmentation into rooms was a computationally expensive step owing to a number of intermediate processing steps. The average time taken by the segmentation step was 22.543 secs. Furniture detection and Furniture classification took 3.968 secs and 5.815 secs respectively. The four feature scores proposed by us namely CAR, RAS, No. of furnitures and Type of Furnitures take 4.253 secs, 3.453 secs, 3.234 secs and 5.736 secs respectively. It was noted that these feature scores took more time in layouts with 5-rooms, due to higher number of doors and inner components owing to the increased rooms. Calculating the cumulative matching score between two layouts and thereby performing the retrieval costed 12.257 secs at an average.

Due to a lack of availability of the implementation of such a composite framework of fine grained retrieval of floor plans in the literature, the results couldn't be compared with other retrieval based approaches. The current approach is motivated by the idea proposed in Chapter 3 in the area of floor plan retrieval [Sharma *et al.*, 2016a]. The key difference between the work discussed in this Chapter and that proposed earlier in this thesis are: (i) previous work does not establish room-wise uniform ordering and correspondence, (ii) the earlier approach took only adjacency and decor arrangement in the whole layout as the parameters during retrieval. In this Chapter, there is an inclusion of high level semantic features like ordering of rooms, area and furniture arrangement in a particular room to aid better fine grained retrieval of floor plans in the *ROBIN* dataset. For making a fair comparison, the results of the baseline method [Sharma *et al.*, 2016a] on *ROBIN* dataset are reproduced and the mean average precision values (MAP) are compared with the newly proposed method in this Chapter. As shown in Tab. 5.2, the approach proposed in this chapter has led to an increase in the mean average precision value as compared to the previous approaches.

The proposed method also has the edge over the approach proposed in Chapter 4 [Sharma

Table 5.2. : Comparison of techniques on ROBIN and SESYD Dataset

Approach	MAP (ROBIN)	MAP (SESYD)
Baseline [Sharma *et al.*, 2016a]	0.2479	1.0
OASIS with RLH [Chechik *et al.*, 2009]	0.0956	1.0
OASIS with BOW [Chechik *et al.*, 2009]	0.1892	0.99
Proposed	**0.3047**	1.0

et al., 2017], where a Convolutional Neural Network based deep framework for retrieval in floor plans is proposed. This deep framework although quantitatively performs better than the method proposed in this Chapter. However, it has a few drawbacks. For example, there is no provision of giving preferences to individual features during retrieval as the layers extract and combine the features implicitly without the provision of manipulating the weight by the user. On the other hand, the provision of a weighted combination of the features proposed in this Chapter lets the user give preferences to individual features, according to his requirement (justified by Fig. 5.11 (b)), thus making it more relevant from a user's point of view. Moreover, the CNN framework is a simple network that mainly distinguishes by the shape of an object or a layout. It would not perform well while doing fine-grain categorization of furniture components, as well as if two layouts from different classes have almost similar shapes.

As the proposed technique deals with similarity in floor plan images thus, the results are also compared with the Online Algorithm for Large Scale Image Similarity Learning (OASIS) [Chechik *et al.*, 2009] coupled with the Run Length Histogram technique (RLH) [de las Heras *et al.*, 2013] and Bag of Words technique (BOW) [Lazebnik *et al.*, 2006b]. These algorithms were chosen for comparison as they are widely popular in object retrieval in document images. It was observed that the proposed feature matching yielded a much better average precision value as compared to OASIS approach (See Tab. 5.2) because OASIS technique focuses on the difference of similarity values between relevant and irrelevant image pairs while ignoring the similarity value between highly similar images. As already discussed, reproducing the approaches mentioned in Tab. 5.2 on the SESYD public dataset yielded a perfect MAP value of 1 due to the low interclass similarity in the samples.

In [Ahmed *et al.*, 2014], the authors proposed a sketch based floor plan retrieval framework, where query layouts are represented as sketches using only primitive geometric shapes such as rectangles. In contrast, in this work, the room decor represented by complex geometric primitives is considered. Furthermore, room decor labeled fine-grained analysis was missing in the work by [Ahmed *et al.*, 2014]. This framework is rotation sensitive to allow relative position of rooms as a feature. Customers may follow Vastu tips and prefer "*master bedroom should be towards the south facing the wall, so that sleeping person's head is facing south*". Making the retrieval rotation invariant will fail to serve this purpose.

5.6 SUMMARY

The proposed framework for fine-grained retrieval of floor plans uses high-level semantic features like area, number, type of furniture and also adjacencies between the rooms. The inclusion of the high-level features during retrieval is a novel idea that can find application in scenarios where the user has specific choices in terms of the fully furnished floor plans. Also, weighted feature fusion

helps in weighing features when a user is querying, and the user is fixated over certain features to be present during retrieval mandatorily.

Till this point in the thesis, only query by example in the form of image has been used. However, the primary concern in this case is that the user must have an image of a floor plan available for querying. In a real scenario, a user may not have a digitized floor plan image to search for similar floor plans. A more intuitive mode of querying is to scribble the requirement as a sketch and query accordingly. Therefore, in the next Chapter, a sketch based input query is discussed, where, the user can draw the layout and the internal furniture components as well. In the current world of smart, hand-held devices this provides an easy mode of search for the end users. Thus, the next Chapter deals with changing the mode of query from images to sketch, to ease querying and make the application more viable.