A Type-2 Fuzzy System-based Approach for Image Data Fusion to Create Building Information Models

Hugo Leon-Garza [1], Hani Hagras [1], Anasol Peña -Rios [2], Anthony Conway [2] and Gilbert Owusu [2]

[1]The Computational Intelligence Centre, School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK.

[2]BT Labs, BT Plc, Adastral Park, Ipswich, UK.

Abstract

Building Information Modelling (BIM) is a standard digital process that fuses buildings information from different sources into a 3D model during their lifecycle. For new construction sites using BIM, it is possible to monitor the cost, schedule, and changes throughout the lifecycle; however, existing buildings do not have a BIM model. Manually creating the BIM models for existing buildings is a high-cost task, both in time and money, hence there is a need for extracting information from available paper-based documentation and fuse it into a BIM model. The struggle of facility management and utility companies to fully adopt a BIM process (due to their high volumes of paper-based documentation of existing buildings) has led to the research on creating these 3D BIM models from 2D floor plan images.

This paper presents a novel processing pipeline to extract 2D digital information from floorplans, fusing it into a 3D BIM model. The work focuses on fusing the available information to create the structure of the building in BIM format, which is considered the essential step before looking on working with other sources of data. In this process, we introduce a type-2 fuzzy logic based Explainable Artificial Intelligence (XAI) approach for the semantic segmentation step. The approach consists of using the outputs of type-2 fuzzy logic systems to classify a pixel as wall or background, by using information around and from the pixel of interest as the inputs to the system. After the semantic segmentation step, the output of the type-2 fuzzy logic goes through a noise removal process and finally a transformation from 2D to 3D by assigning the corresponding BIM tag to each identified element. The proposed type-2 fuzzy logic semantic segmentation approach produced comparable results (97.3%

mean Intersection over Union (IoU) performance metric value) to the opaque box model approach based on Convolutional Neural Network (CNN) (99.3% mean IoU performance metric value). However, the type-2 fuzzy XAI system benefits from being an augmentable and interpretable model, which means that human users can understand the decision process and modify the model using their expert knowledge.

Keywords—Building Information Modelling, BIM, semantic segmentation, type-2 fuzzy logic systems, convolutional neural network (CNN).

1. Introduction

Building Information Modelling (BIM) has become the standard digital process for working with new building sites during their lifecycle. During these phases, information from different sources is fused into 3D digital models that serve as a data-rich digital archive with all the information of every element in the building, including geometric and geographic data, relationships between elements and properties information [1]. The use of BIM in the construction industry is well established, and there are many benefits when managing the time, cost and quality of a project [2], [3]. However, for facility maintenance and utility companies, the biggest challenge is transitioning from their current paper-based process to a BIM process. One of the significant limitations of companies who are in the process of adopting BIM is the integration and fusion of information [4], where the more significant part of the information used by these companies remains in paper-based documents, i.e. paper drawings or CAD (Computer-Aided Design) floor plans instead of having a computerised 3D model.

Furthermore, there is a compatibility need [4] where the produced BIM data should be compatible with tools used by those companies. Moreover, the existing building data should be transformed to a compatible format; otherwise, companies will be forced to work with multiple tools and technologies for the different cases. This challenge is the *legacy data problem,* i.e. how are utility companies handling paper-based documentation of network assets in buildings that do not have 3D computerised BIM models?

This paper focuses on extracting all the information needed to create BIM models by processing and merging existing building structure data sources. The two main challenges are 1) How can we use the information available in the building's paper-based documentation and 2) how can we automatically fuse it.

Therefore, we present a processing pipeline that describes a series of steps to convert 2D floor plan images into computerised 3D BIM models. The proposed processing pipeline includes two tasks: the semantic segmentation task (i.e. the process of classifying each pixel in an image) and transforming the segmentation result to a 3D computerised BIM model. We present a type-2 fuzzy system based Explainable Artificial Intelligence (XAI) approach for the semantic segmentation task. The proposed type-2 fuzzy logic semantic segmentation approach produced comparable results (97.3% mean IoU performance metric value) to the opaque box model approach based on Convolutional Neural Network (CNN) (has a 99.3% mean IoU performance metric value). However, the type-2 fuzzy XAI system benefits from being an augmentable and interpretable model, which means that human users can understand the decision process and modify the model using their expert knowledge.

In Section 2, we discuss a brief overview of BIM. In Section 3, we present the proposed processing pipeline. Section 4 presents the deep learning approach using a Convolutional Neural Network (CNN) used to compare against the employed type-2 fuzzy semantic segmentation approach. The type-2 Fuzzy Rule-based System (FRBS) is presented in Section 5. Section 6 presents the semantic segmentation results and the image processing steps to create the BIM models using the extracted information. Finally, in section 7, we present our conclusions and future work.

2. Basic Concepts: BIM and Interval Type-2 Fuzzy Logic Systems

In this section, we introduce the concepts of BIM and Interval Type-2 Fuzzy Logic Systems. Creating a BIM file with the basic structure of a building floor is the goal of our system. In the first subsection we describe what BIM is and how does it help or benefit companies that make use of it. In the second subsection we describe interval type-2 fuzzy logic system which are used in the semantic segmentation step of our process.

2.1 A Brief Overview on Building Information Modelling

Building Information Modelling (BIM) is the standardised and open-source process to manage a building's lifecycle, from design to maintenance [5]. The main product of this process is a data-rich computerised 3D model that has relationships, properties and geometric information for every element in the building [6]. The 3D model allows the user to interact with the building as a whole or only select specific elements. The model is built in the design phase with information from different sources, it is monitored and updated during the construction phase, and the goal is to function as a historical archive during the maintenance phase. Having an accurate virtual representation with the correct position of all elements in the building would help planning and perform maintenance tasks more efficiently.

The advances in technology and software development that allow people to work with 3D models have enabled BIM adoption. The construction industry has led the way in adopting this process to use these digital models as centralised documentation of the site, moving to a paperless working environment and saving up to 35% [7]. The design phase has benefited from this digital approach, and some research has focused on using the relationship between elements to make indoor route planning [8] and then make changes if needed, useful for emergency exits planning. Other phases of the building's lifecycle have also received attention; for the construction phase, the main goal has been to take these large-size 3D models to the construction site and visualise them. Existing research shows the idea of developing frameworks to visualise the BIM models on-site [9], [10], and how to use Augmented Reality and mobile technologies to visualise these models [11], [12].

Companies in the construction industry have seen the benefits of BIM when managing the time, cost and quality of projects, and the trend is to adopt this new digital process. A large amount of information available in BIM allows construction companies to have a better understanding of the costs of each element and the complete project, identifying conflicts between elements in the early stages, and having a better time estimate of the project by better understanding how the construction of a section will affect others and which elements need to go before others [2], [3]. However, utility companies and maintenance companies are caught in between two processes. They are trying to adopt the new digital BIM process, but they still need to provide service to existing buildings that do not have BIM models.

For many buildings, the only available documentation exists in 2D floor plan paper documents, driving the goal of transforming legacy data to BIM models. Creating digital models for existing buildings is a high-cost task, both in time and money [13], and it might be one of the main reasons why BIM has a low adoption rate in these companies [13].

From a utility company point of view, using an accurate data-rich 3D model of the site could benefit the company in at least three ways: 1) have an updated and accurate record of the assets within the building, 2) provide on-site guidance for field engineers in their maintenance or repair tasks and 3) help field engineers to understand the cause of a fault before they get on site. However, field engineers usually have access to paper-based documentation only for existing sites, which can be outdated.

Gimenez et al. [13] reviewed different alternatives on how to create BIM models for existing sites. They divided the methods into two main categories: 1) BIM models from on-site data and 2) building documentation. The first category includes data sources such as aerial photographs and city building images; and scanning tools such as 3D laser scanners and mobile applications. Although most of these techniques may provide a highly accurate description of the building, they involve high-cost activities in the data acquisition process, e.g. training a field engineer to use a 3D laser scan, the time cost of sending the engineer to the site or the cost of acquiring the necessary tools. Expanding these costs for multiple buildings and sites might be too high for companies to absorb, hindering BIM benefits and slowing its adoption. Therefore, this work focuses on processing data sources with higher availability and lower cost (e.g. the use of sketches, CAD plans and 2D floor plan images). These data sources are included in the second category proposed by Gimenez et al. [13]. We propose a processing pipeline that takes the existing building documentation as an input and converts it to a BIM model. The advantage of using a fully automated processing pipeline combined with highly available information is a low-cost solution.

Moreover, using an explainable AI model allows end-users to modify and improve the model using their expertise. The disadvantage is that there is a limit of information to what can be extracted from a floor plan; the BIM model will never be as complete as when a laser scanner tool is used, e.g. from a 2D floor plan image, the height of the walls cannot be extracted. However, creating an initial

BIM building structure is essential as a base reference model for planning engineers. This initial structure can be extracted from 2D floor plan images by identifying essential elements such as walls, doors, and windows.

2.2 A Brief Overview on Interval Type-2 Fuzzy Logic Rule-based Systems

Fuzzy Logic (FL) use fuzzy sets, an extension of classical sets. In these sets, a numeric value called membership value (or degree of membership) is used to describe how much an input value belongs to a given set [14]. In classical sets, an input value either belongs or not belongs to a set, while in a fuzzy set, it can belong to multiple sets. This allows us to handle the uncertainty of an input that is close to the limits between sets.

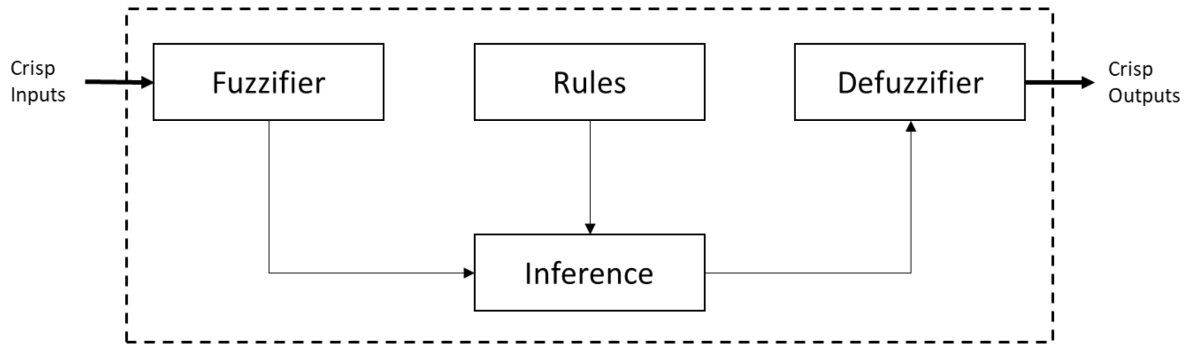


Figure 1: The basic components of a Fuzzy Logic System [15].

Fuzzy Logic Rule-based Systems (FRBS or Rule-based FLS) use fuzzy logic to address the imprecision of inputs and output variables by describing them with fuzzy sets that can be expressed in linguistic terms (e.g. small, medium, and large). The basic components of a FRBS are shown in Fig. 1. These components work together to map crisp inputs to crisp outputs [15]. The rules and inference components are responsible for mapping the input to the outputs by checking which rules are fired. A rule is fired if the input vector belongs to the antecedents in the rule, then the consequences are used to compute the output value. Human experts can define rules, or they can be extracted from data. Either way, each rule is defined by two sets of linguistic labels, antecedents, and consequences. To use these rules, it is necessary to transform crisp inputs to linguistic labels and linguistic consequences to crisp outputs. The fuzzifier and defuzzifier will be responsible for this, using membership functions. These functions are mathematical functions that define which values (and at which degree) belong to the fuzzy set associated with the membership function.

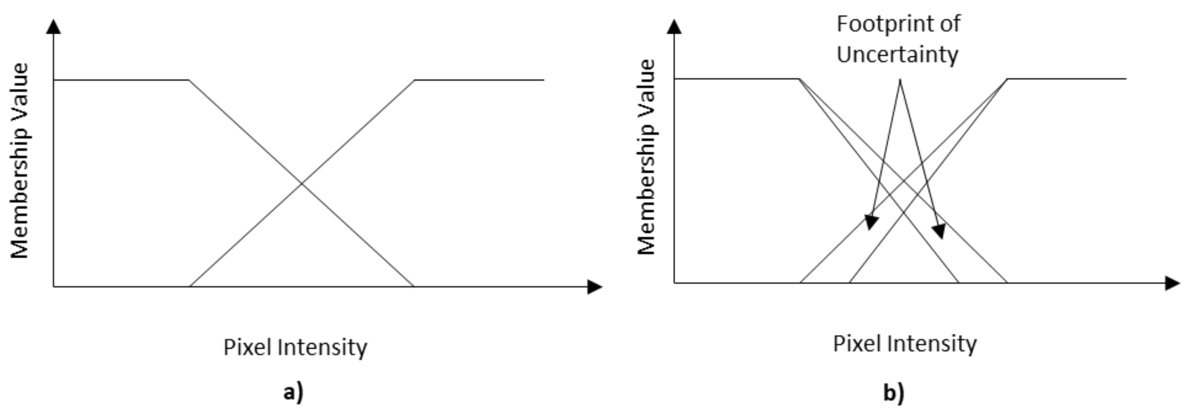


Figure 2: Examples of membership functions. a) Type-1 membership function and b) Type-2 membership function.

Fig. 2 shows two different types of membership functions. Fig. 2a describes a type-1 fuzzy set, and Fig. 2b describes a type-2 fuzzy set. Type-1 fuzzy sets were first introduced as a concept to handle the uncertainty of representing numeric values as linguistic terms, i.e. which numeric values belong to a specific term and the degree of belonging of each value [14]. Each linguistic term will have a type-1 membership function associated that will be used to define which values belong to it, and it will have smooth transitions at the limits, i.e. the degree of membership of values at the limits of the sets will start to decrease as the numeric values are further away. However, type-1 fuzzy membership functions use a crisp number to specify the degree of membership of each input value. Therefore we now have uncertainty on whether the precise membership function was correctly defined [14]. Type-2 fuzzy sets were then proposed to handle the uncertainty of using precise functions [16]. Type-2 membership functions can be seen as a type-1 membership function where the line defining the function was blurred, then at a given input value, the degree of membership is a range instead of a crisp value [15]. This range of values describing the degree of membership is called the Footprint of Uncertainty (FOU) of a function, as shown in Fig. 2b. Interval type-2 membership functions have for each point in the range an associated secondary membership of 1 [14]. Interval type-2 can be seen as a set of multiple type-1 membership functions that are together. Therefore the membership value is defined by two crisp membership values, one from the upper bound and another one from the lower bound. The proposed

FRBS uses Interval type-2 membership functions because the extra degrees of freedom from the type-2 fuzzy sets [14] results in a higher performance metric value.

Additionally, FRBSs are explainable AI models [17], i.e. they can be understood by the end-user and modified using the user's expertise. Our proposed FRBS will be used for classification tasks which means that there is no need for the defuzzification process where consequences are converted to crisp outputs. Instead, a similar approach to the one presented in [18] is used, where inputs are converted to linguistic labels and fired rules vote for a consequence, the consequence with the highest votes is the class assigned to it. The process of building the proposed FRBS is discussed in the following sections.

3. Extracting Information from 2D Digital Information to Generate 3D BIM Models

Having BIM models of infrastructure can bring many benefits, not just to the construction sector but also to utility and maintenance companies. This section introduces a processing pipeline to extract information from 2D digital information to merge it into a 3D BIM model. To do so, we introduce a type-2 fuzzy logic-based Explainable Artificial Intelligence (XAI) approach for the semantic segmentation step in the processing pipeline. The work focuses on fusing the available information to create the structure of the building in a standardised BIM format model. The process combines the information obtained from human expert knowledge, an optimised segmentation model, and external information (e.g. floor height and geographic location) to create the BIM model visualisation. The proposed process expands from the one proposed by Gimenez et al. [13], with the difference that it focuses on using semantic segmentation techniques.

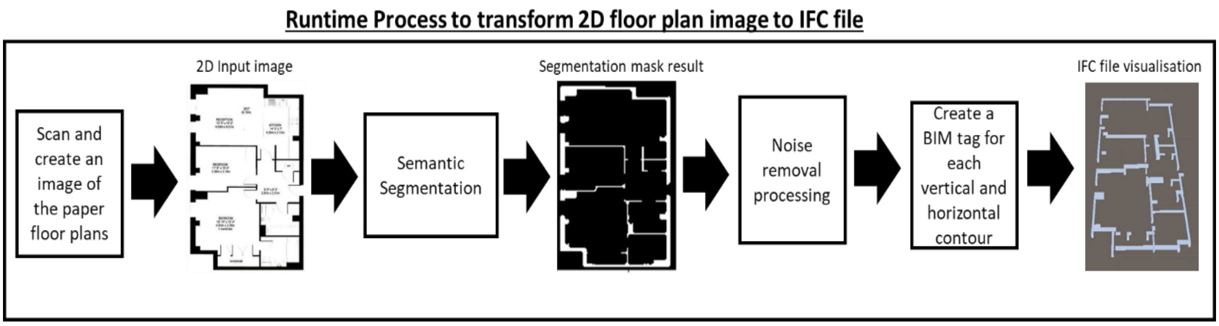


Figure 3: Proposed processing pipeline for converting 2D floor plan images to standardised BIM models.

Fig. 3 shows the proposed processing pipeline to transform 2D floor plan images, CAD plans or floor plan sketches (legacy data) to BIM models using semantic segmentation. The core steps of the process are:

1) **Digitalisation of Existing Documentation.** It transforms existing documentation (e.g. architectural drawings) from paper format to a digital format (i.e. digital 2D image). The transformation can be done using a scanner or a camera; however, we should consider that the higher the resolution, the better it will be for the segmentation process.
2) **Semantic Segmentation of the Digital 2D Images**. This is the main step of the processing pipeline, where elements are automatically identified. Semantic Segmentation allows us to do that by assigning a label to each pixel in the image. In this case, we are looking at identifying which pixels are part of a wall. In the following sections, an explainable AI fuzzy rule-based system approach is presented. This approach allows us to combine the knowledge obtained from optimising the model using a training data set and the human expertise of the end-user. The fusion of the knowledge happens in the rules, i.e. some rules are created from data, and others will be added or modified by the end-user. The model will automatically identify which pixels belong to the wall structures and which are background or part of another element. Fig. 3 shows an example of the resulting image; the segmentation mask result has white colour pixels where wall elements were identified.
3) **Noise removal process**. Some pixels in the floor plan images might be incorrectly classified, which will result in a segmentation mask with noise pixels. These need to be removed before creating a BIM representation for all the identified objects. The noise removal process uses image processing techniques such as median, dilation and erosion filters to remove isolated pixels incorrectly labelled. The idea is that large groups of pixels representing a wall will not be affected, and small groups (or individual pixels) that are not a wall will be changed and labelled as background.
4) **Conversion to BIM**. This final step aims to convert all the extracted information to a BIM model file using the IFC standard. Industry Foundation Classes (IFC) is a global open standard

for data exchange; used to describe and share construction and facilities management information [13]. By using this open-source standard, our model has high compatibility with most of the BIM tools available. During this step, we fuse the information extracted from the segmentation model and the available external information (e.g. floor height, the geographic location of the model, and structure material) to create a BIM model.

This section introduces the proposed processing pipeline and the result from combining different sources in each step. In the following sections, we present a detailed description of this process and its results.

4. The Proposed Interval Type-2 Fuzzy System for Semantic Segmentation

This section discusses using a Fuzzy Rule-based System (FRBS) as an interpretable and augmentable model for semantic segmentation of floor plans, which serves as an alternative to opaque (black) box segmentation models. Our approach is based on rules that combine local information and context information around the pixel of interest. We use a similarity value between image patches as the context information needed by the model. Each rule will vote for a label, and the one with the highest vote value will be assigned to the pixel of interest. A vote and label will be computed for every pixel in the input image.

The process to build this model consist of 3 main parts:

- The patch extraction and creation of visual words
- The rule modelling process to create rules
- The optimisation of the fuzzy model using the Big-Bang Big-Crunch Optimisation.

In the following subsections, we present a detailed description of these steps.

4.1 Visual Words Dictionary

The rules in the FRBS use two antecedents, one for the pixel intensity and one for context information, i.e. an antecedent that provides information about the surrounding of the pixel of interest. The use of similarity between image patches in a FRBS [19], [20] consists of combining the colour information of the pixel of interest and context information of the pixels around it to assign it a label. The context information is computed by calculating the distance between the patch with the pixel of interest and a list of pre-computed visual words. A patch-based approach improves the decision process of the FRBS by providing the rules with context information that is needed for tasks where colour segmentation is not enough.

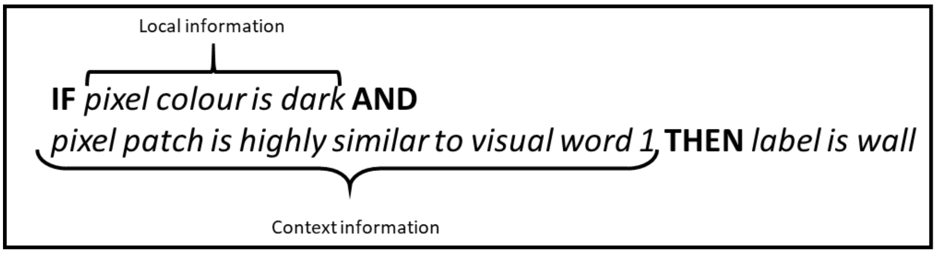


Figure 4: Example of a FRBS model rule and a description of the type of antecedents found in the rules.

Fig. 4 shows how the different information from the antecedents is combined to classify pixels. Information from the first antecedent is the pixel value, and there is no other process needed to extract it. For the second antecedent, the system computes the similarity of the image patch containing the pixels with a pre-computed dictionary of visual words, i.e. a list of numeric vectors representing the average look of different types of patches.

A patch-based approach for semantic segmentation was first introduced by [21], [22], where they combined the information from patches with a Markov Random Field model. The approach was then extended in [23], [24], where the authors trained a Support Vector Machine (SVM) model to segment floor plans using information from image patches. In [19], [20], the authors combined image patches with a Fuzzy Rule-based System to create an interpretable and augmentable model for semantic segmentation. This work relies on the idea of creating a visual vocabulary or dictionary, which will serve as our knowledge base and will be used to compute the similarity of new input patches to the model’s knowledge. The number of visual words used by our proposed model is considered a hyperparameter of our approach and it needs to be modified according to task. Values from 50 to 300 visual words were tested in our work. The model used with our training and testing data achieved the best performance value when using 100 visual words.

A training dataset of images is needed to create the visual words dictionary, and the process consists of the following steps:

1) **Divide the training image into patches.** The image is divided into sections by using a grid. An overlapping grid is used to avoid the location dependency of the object in the image [24] and to extract a more diverse set of image patches from a single image. After extracting the patches, they are transformed to a numeric vector representation following a row-wise approach and then Principal Component Analysis (PCA) feature selection is applied to it. Two main parameters can be changed in this step: the overlapping value in the grid (i.e. how many pixels overlap between patches); and the size of the image patches extracted.
2) **Cluster the extracted patches using the k-means algorithm**. The centroid of the computed clusters is the numeric vector used as a visual word. We have a visual word for each cluster; therefore, the size of our visual words dictionary determines the value of k in the clustering algorithm. This k is a hyperparameter of the model, and it can be optimised. It is essential to consider that the higher the value of k, the smaller the clusters and the more specific to a training set the model becomes. Additionally, when increasing the k value, the number of possible rules will also increase.

4.2 Rule Modelling Process

The FRBS is created using a data-driven process, i.e. all the rules and fuzzy sets membership functions are extracted from data, and there is no human intervention when creating the model's initial state. An initial set of rules can be created by computing all the possible combinations of antecedents and then optimised. However, the consequence of each rule cannot be computed just by finding the permutations. Assigning a consequence to each rule, i.e. determining the class each rule will vote for when it is activated, is a more complex process. We use a training dataset for the labelling process to find which label is the visual word related. The rule modelling process used in this paper is based on the concept of "weighted confidence" presented in [25] and the concept of "weighted scaled dominance" presented by [18].

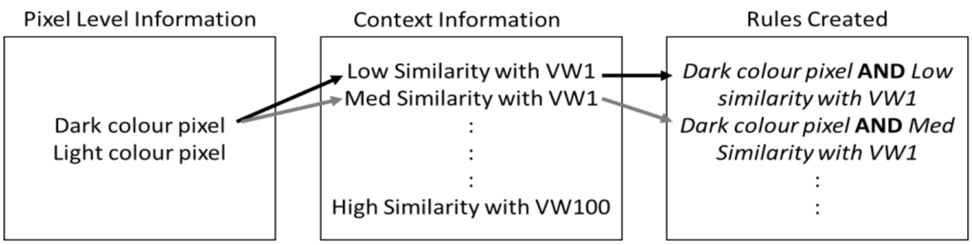


Figure 5: Computing all the possible combinations for the antecedents to create the initial set of rules. Pixel level information has 2 possible antecedents and Context Information has 300 possible antecedents.

The process of computing all possible combinations to create the initial set of rules is shown in Fig. 5. The next steps are followed when creating the rules:

1) Define the different antecedent variables. In this case we have “Pixel Level Information” and “Context Information”, these are the only two variables since it is the information that is currently extracted from the image.
2) For each of the antecedent variable define all possible linguistic values that are part of the antecedent. For “Pixel Level Information”, since the images are grayscale images, only two linguistic values were defined, one for dark colour pixel and one for light colour pixel. The number of linguistic labels in context information is defined by the number of visual words in our approach. As mentioned before, different values were tested but the best performance metric was achieved with 100 visual words. The similarity with each visual word is described with 3 linguistic values (low similarity, some similarity and high similarity). This means that for context information there will be 300 possible antecedent values (3 linguistic values multiplied by 100 visual words).
3) To generate the initial set of rules, all possible combinations are computed using the sets of possible values for each antecedent variable. Rules will only have one value from each antecedent variable set.

The rule modelling comprises the steps detailed in the following subsections:

4.2.1 Assigning labels to rules

First, we create all the possible rules based on two constraints. The first constraint is that rules should only have two antecedents. The second constraint is that one of the antecedents is always the local pixel information. In this case, the pixel intensity value and the other antecedent will be the context information provided by the similarity value between the input patch and patches in the visual words dictionary. In a FRBS model with 100 visual words, the initial total number of rules will be 600, the similarity to each visual word is described with three linguistic labels (similar, somehow similar, not similar), and the pixel intensity value is described by two linguistic labels (dark and white). When extracting the consequence of the data, we make use of all the rules. In the following stages, the number of rules can be optimised.

For the rule modelling phase, we have a dataset with M rows. Each row will be a training pattern called *t(m)*, *m=1, 2, …, M* that consist of a vector *x(m)* and a class *c(m)*. The vector *x(m)* has all the needed information related to the pixel, and this includes the pixel intensity value and the similarity of the image patch centred at the pixel to the patches in the visual words dictionary. For example, if the model uses a dictionary of 100 visual words, the vector *x(m)* will contain the numeric value of the pixel intensity and 100 numeric values. Additionally, *r(m)* will also have the linguistic value *c(m)*, which is the expected class for the pixel information *x(m)*.

For each rule, we compute the firing strength *f(m)* using the membership functions of each antecedent. This value measures the vector *x(m)* belonging to the fuzzy region of that rule. The FRBS model uses interval type-2 fuzzy logic, which means that two values define the firing strength *f(m)*, the lower ($\underline{f(m)}$) and the upper ($\overline{f(m)}$) bounds of the interval type-2 membership functions.

Datasets for semantic segmentation are highly unbalanced because most of the pixels of an image will not be part of the class of interest. In this work, we are trying to identify pixels that are part of a wall in floor plan images. Most of the pixels in the floor plan image will either be background or another element. To handle the unbalanced data, we adopted the approach of “weighted scaled dominance” presented in [18], [26]. This method gives classes that are a minority in the dataset a fair chance when competing with majority classes. The method uses “scaled firing strength” ($fs(m)$, also defined by the lower $\underline{fs(m)}$ and upper $\overline{fs(m)}$ bounds), instead of the regular firing strength, this is computed by dividing the firing strength of the rule by the summation of firing strength of all the rules with the same consequence [18]. This step is repeated for every training pattern in the dataset. The scaled firing strength will be used in the next step to compute the scaled confidence and scaled support values.

4.2.2 Scaled Support and Confidence for Solving Rule Conflicts

After extracting the labels from the data, it might be possible that some rules will have two or more consequences assigned; these are considered conflicting rules. To solve the conflicts in these rules, i.e. decide which of the multiple consequences assigned to the same set of antecedents is the most appropriate, we use the confidence and support values. These values have been previously used to evaluate the rules and solve the conflicts in [26], [27]. Confidence is the value that measures how likely is it for a set of antecedents to have a specific consequence. The lower and upper bounds define it because of the use of interval type-2 fuzzy logic. Equation 1 computes the confidence for class q, the summation of the firing strength of the rule for all the training patterns with an expected class q is divided by the summation of the firing strength of the rule for all training patterns. The scaled confidence is the same measure value but computed using the scaled firing strength, which helps handle the unbalanced data.

$$c\left(\widetilde{A_q} \Rightarrow C_q\right) = \frac{\sum_{x_s \in Class\ C_q} fs^{jt}(x_s)}{\sum_{j=1}^{m} fs^{jt}(x_s)} \qquad (1)$$

The support metric is the measured value of the coverage of training patterns for a given rule. A rule can have high confidence, but only a couple of training patterns in the dataset supported the combination of antecedents and consequences. This means we do not have enough data to support that rule, and the support measure will show this. Equation 2 is used to compute the support for class q. The summation of the firing strength of the training patterns with an expected class q is divided by the total number of training patterns. As with the confidence measure, scaled firing strength allows us to compute the scaled support and handle the unbalanced data.

$$s\left(\widetilde{A_q} \Rightarrow C_q\right) = \frac{\sum_{x_s \in Class\ C_q} fs^{jt}(x_s)}{m} \qquad (2)$$

Finally, we combine confidence and support in a measure called scaled dominance. The dominance value is a simple multiplication of both measures. To solve the conflict in rules with multiple consequent classes, we select the highest scaled dominance value.

4.3 The Optimisation of the Fuzzy Model using Big Bang – Big Crunch

The membership functions in a fuzzy logic system are the mathematic functions that model the linguistic labels in the system. A change to these functions will influence how the system interprets each linguistic label, and therefore the performance may decrease or increase. Finding the optimal membership functions is one of the crucial steps in optimising the fuzzy rule-based systems. In this subsection, we describe how the Big Bang – Big Crunch (BB-BC) algorithm can be used to optimise the membership functions that describe the pixel intensity and patch similarity antecedents.

The BB-BC optimisation algorithm presented in [28] is an evolutionary optimisation technique based on the big bang theory in physics. The main advantage of BB-BC is the high convergence speed and the low computation time. In the work of Erol and Eksin [28], the algorithm was compared on benchmark problems to other Compact Genetic Algorithms (C-GAs) and BB-BC outperformed C-GAs in convergence speed and computational time. In another study [29], BB-BC achieved equal or better results than Genetic Algorithms (GAs) and Particle Swarm Optimisation (PSO) in convergence speed and execution time, and it proved to be less dependent on the randomised initialisation of the generations. Additionally, BB-BC has a simple implementation and it has already been successfully implemented for the optimisation of fuzzy logic systems [30]–[32]. This optimisation process consists of a Big Bang phase and a Big Crunch phase. The first part explores the universe by applying random changes to the current best solution. In the second part, the optimisation process converges towards the new best solution by evaluating each candidate and selecting the best according to a predefined performance metric.

BB-BC and other evolutionary algorithms have been successfully implemented in the optimisation of fuzzy logic systems in different areas such as finance [18], PID controller [33] and machine vision [34]. Implementing it is to encode the membership functions to a vector of values that can be optimised. The main differences in the literature review implementations are in the encoding process and the meaning of each element of the encoded vector. In this work, the encoding process used has previously been tested in [27]. In this process, a list of numbers is created; each number is a point in the fuzzy set needed to represent the shape of the membership functions, then changes to this list of points are made following the BB-BC algorithm to find the best possible combination. Fig. 7 shows the optimised points in the set. Some constraints need to be followed when optimising type-2 membership functions so that the upper function is always the upper function. This process has two main assumptions: 1) the first and last membership functions are respectively the left and right shoulder function, 2) all the other membership functions are triangular shape functions. The initial state of the membership function for both fuzzy sets can be seen below in Fig. 6.

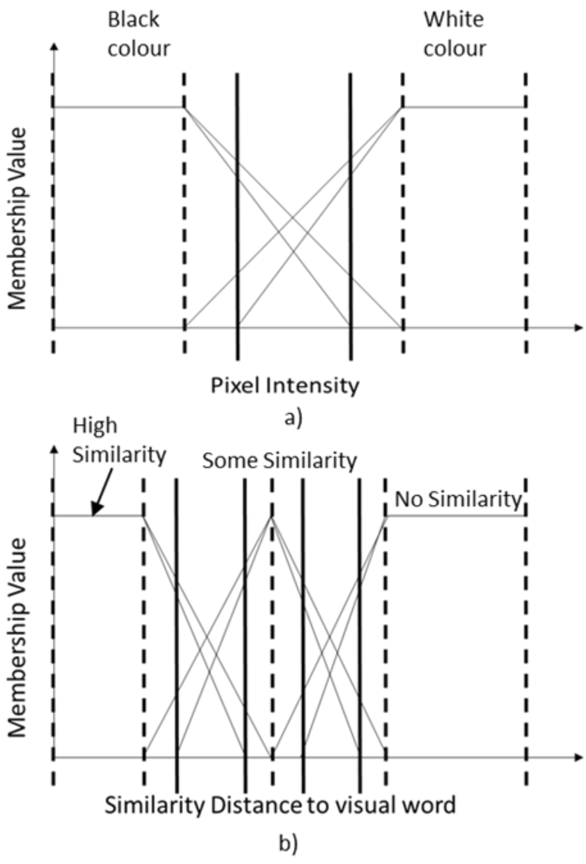


Figure 6: Initial state of the membership function before optimisation. Dashed vertical lines show optimisation points for type-1 fuzzy membership functions, and solid vertical lines show the additional points needed for type-2 fuzzy membership functions. (a) Pixel Intensity input fuzzy sets. (b) Similarity Distance to visual word input fuzzy sets.

In Fig. 6, membership functions can be seen for both input fuzzy sets before optimisation. The fuzzy set was equally divided for the initial state of the membership functions, and the membership functions have the same coverage. The vertical lines in Fig. 6 show the selected points for optimisation. The BB-BC will change to the points and move them to the left or the right, always keeping the same order. The changes to the points will become smaller as the generations advance. These points describe the shape of the membership functions. Dashed lines show which points are needed for type-1 fuzzy membership functions, and solid vertical lines show which points need to be added when working with interval type-2 fuzzy membership functions.

The number of points to be optimised per fuzzy set can be calculated using Equations (3) and (4)

$$Number\ of\ parameters\ for\ Type\ 1 = M + 2 \qquad (3)$$

$$Number\ of\ parameters\ for\ Type\ 2 = M * 3 \qquad (4)$$

M is the number of membership functions in the fuzzy set. After computing the number of points, a vector is created, and the points can be initialised randomly or with an equal distance between them. There are two critical things to consider: 1) each fuzzy set will need to have its own vector, and 2) a point in the vector cannot be lower than the previous point or greater than the next point.

After understanding the encoding process, it is possible to apply the BB-BC optimisation process following the next steps:

1) Initialise a population of N candidate solutions. Each candidate has a vector for each fuzzy set that describes the membership functions that belong to that set.
2) Use the training data and the rule modelling techniques (described in the following subsection) to create a Fuzzy Rule-based System for each candidate solution.
3) Evaluate the performance of each candidate FRBS.
4) Select the best candidate solution of the generation and compare it to the historic best solution.

5) Start a new population and create a new set of N candidate solutions. All of them should be created using the equation $x^{new} = x_c + \frac{lr}{k}$. The historic best solution is x_c, l is the limit of the search space, k is the iteration number, and r is a random number between -1 and 1.

6) Repeat steps 2 – 5 until the maximum number of iterations are completed or until other stopping criteria are completed.

Fig. 7 shows the membership functions with the best performance metric after completing the multiple-optimisation process.

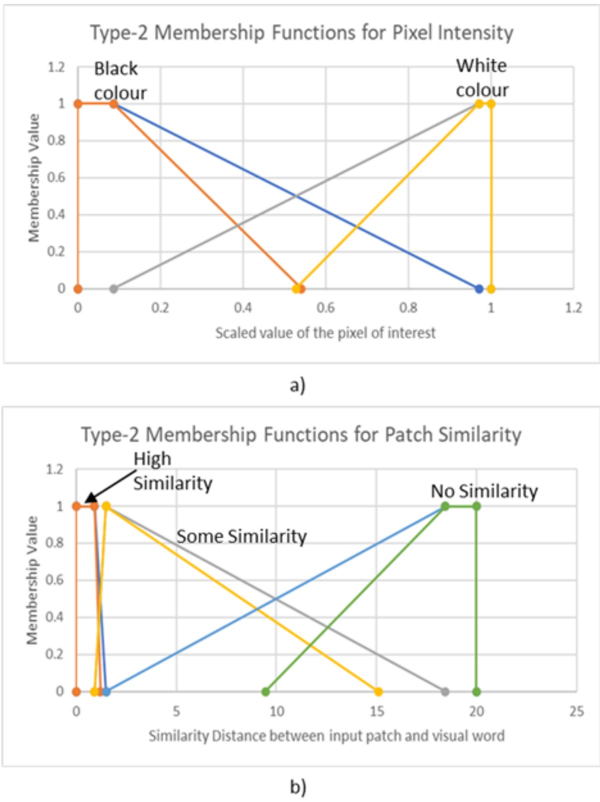


Figure 7: Optimised membership functions used in the FRBS model from [20].

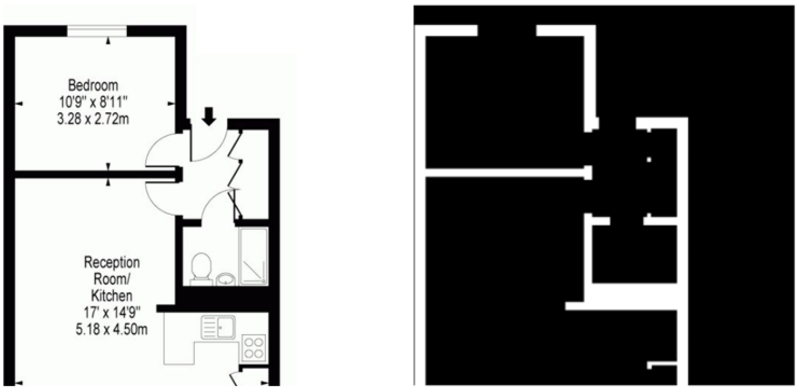
The previous process described the optimisation of the membership functions used when transforming crisp inputs to linguistic labels. For the rules, it is possible to optimise the size of the rules (number of antecedents per rule) and the size of the rule base (total number of rules available). This work focused on optimising the rule base size since the rules were built with only two antecedents. Thus, there was no need to optimise the number of antecedents. The process to optimise the size of the rule base is as follows:

1) **Create a list with all the possible rules**. Sort that vector so that the difference in antecedents is the lowest possible. For example, rule 1 should have as antecedents *low pixel intensity* and *no similarity with visual word 1*, rule 2 should be *low pixel intensity* and *some similarity with visual word 1*. The antecedent that changes is *similarity with visual word 1*. Since *no similarity* and *some similarity* are antecedents close to each other in the fuzzy set, the difference between the rules is low. Each rule in this list will have an index number associated and will be used during the optimisation process.
2) **Create a list with N random index numbers from the list of all possible rules**. N is the maximum number of rules that the rule base is allowed to have. The objective is to find the N rules that together have the best performance metric value. This list is a candidate solution, and it is possible to create as many candidate solutions as needed.
3) **After candidate solutions are created, the BB-BC optimisation algorithm (see the six steps previously described) can be used to find the optimal set of N rules**. In this case, when we perform the changes from step 5 of the BB-BC algorithm, the goal is to change the index number we are looking at for another index number. At the beginning of the BB-BC process, the value $\frac{lr}{k}$ is larger, which results in significant rule changes. As the BB-BC generations advance, the value $\frac{lr}{k}$ becomes smaller until there are no changes in the rule index value.

The BB-BC optimisation algorithm was applied as described in the previous three steps, and it is possible to reduce the size of the rule base. When reducing the size of the rule base, it might happen that some inputs will not fire a rule; hence, there will be no prediction for those input vectors. The prediction of those input vectors is computed with the most similar rule in the rule base to handle these cases. The most similar rule is the closest rule in the sorted list of all possible rules (see previous step 1 of rule base size optimisation), i.e. the rule with the most similar antecedents [26].

5. Experiments and Results

In this section, we present the results for the proposed approach for the semantic segmentation tasks. Additionally, a description of the process followed to transform the semantic segmentation to a standard 3D BIM model and VR visualisation of these results.



a) b)

Figure 8: Example images from the test set. a) input image and b) expected segmentation result for the input image.

Our training and testing dataset have 168 images similar to the ones in Fig. 8. A visual comparison with the data used in [35], [36] shows that these images are more straightforward because there is less noise in the input images (i.e. fewer pixels that are not background or wall). However, there are still four main challenges for the segmentation models: 1) remove the text, 2) remove other elements such as the kitchen or bathroom elements, 3) remove door elements and 4) remove window elements.

a) b)

Figure 9. Segmentation results for the proposed FRBS model. a) before the noise removal process and b) after the noise removal process.

Fig. 9 presents the results of the proposed type-2 fuzzy semantic segmentation. In order to establish the efficiency of the proposed method, we have compared it with Convolutional Neural

Networks (CNNs), U-Net specifically, which has emerged as the state-of-the-art for semantic segmentation [37]. This architecture, introduced by [38], consists of a contracting path and an expansive path. The contracting path has blocks where the network performs operations of convolution, ReLU activation and max pool. The block always ends with a max pool operation that will reduce the dimensionality of the input image. The feature map extracted in each block is the input of the next block and an input set of features in the expansion path. The expansion path consists of blocks of deconvolution (or Transpose Convolution) layers that will help bring the set of features back to the original image dimensions. In the end, this state of the art architectures, take an input image with a width, height and number of channels and output an image with same dimensions but the number of channels will be equal to the number of classes, and only the channel corresponding to the predicted class is activated [39].

One of the advantages of using more complex networks is using them (or parts of them) to create new architectures. This is known as transfer learning and has already been used for semantic segmentation in [40]. Our work implemented transfer learning using the VGG-16 model (in the Keras section of the Tensorflow python library) as our U-Net implementation contraction path for floor plans.

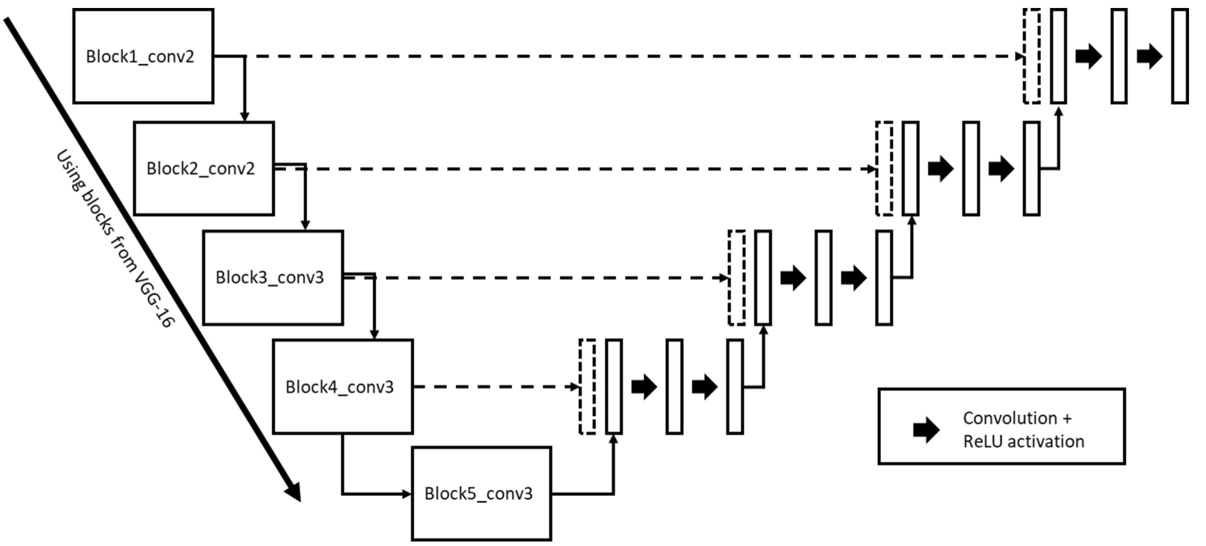
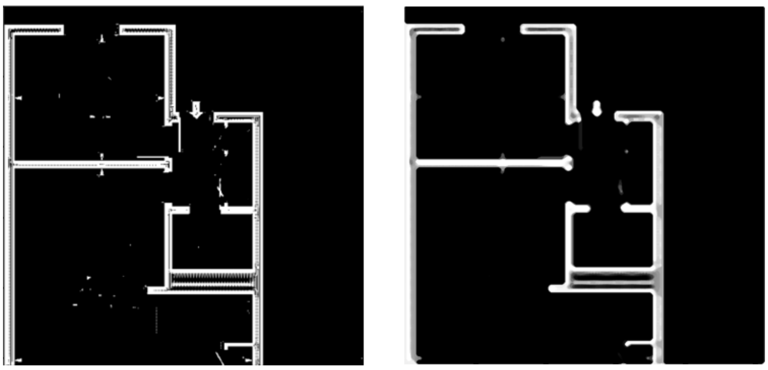


Figure 10. CNN U-Net architecture used in our deep learning approach for segmenting floor plan images.

Fig. 10 shows the implemented architecture of CNN. As described before, the contraction path is based on the VGG-16 model, and we take the output of specific blocks. The expansive path is blocks of first a deconvolution operation, then a concatenation operation with the feature set the output of the block at the same level in the contraction path, followed by convolution and ReLU operations.

To numerically evaluate the results from our models, we used the Intersection over Union (IoU) metric, and we used the mean of the IoU between wall pixels and background pixels. We used the IoU metric as it is considered as a main standard performance metrics for the semantic segmentation task. The membership functions of the initial version of the proposed type-2 fuzzy model were constructed using expert knowledge only, and the rules used were all the 600 obtained from all the possible combinations of antecedents. This model achieved a performance value of 94% mean IoU. The next step was to create an optimised version of the proposed model using the abovementioned BB-BC algorithm, the shapes of the membership functions and the number of rules were modified. The optimised type-2 fuzzy model used the membership functions shown in Fig. 6, the number of rules was reduced to 400 and it achieved a mean IoU metric of 97.5%. The number of rules can be further reduced, however, the performance of the model will be affected where an optimised model with 200 rules achieved 96.7% mean IoU. The adjustments to the fuzzy rule-based system made by the BB-BC algorithm improved the final performance showing the benefits of the optimisation step. In Fig. 9, the visual results for the FRBS model are shown, there are two main things to mention: 1) the main problem of the model is text and 2) the model can remove other elements such as the kitchen and bathroom elements.



a) b)

Figure 11. Segmentation results for the CNN U-Net architecture model. a) before noise removal and b) after noise removal.

Fig. 11 shows the results for our U-Net CNN approach. After visually comparing these results to the ones from our fuzzy approach in Fig. 9, there are two significant observations to mention: 1) the deep learning approach is efficient at removing text and 2) wall (or structure) elements that are large, are not entirely labelled as a wall; instead some of those pixels are labelled as background. The mean IoU metric value for the CNN is 99.3%, which is higher than the performance of the FRBS, and visually we can say that most of the difference comes from the capability of the CNN to identify text. Our main conclusion from these visible results is that our proposed FRBS rules are still substantially based on colour, even though we included patch similarity information. This explains most of the visual errors of the FRBS in Fig. 9.

However, CNNs are black box models that are not interpretable or augmentable, i.e. a human end-user will not understand the system's decision process and will not be able to modify it [17], [41]. Additionally, these models need large numbers of training data patterns to perform well. Obtaining data is costly, especially for semantic segmentation where the label needs to be pixel by pixel [39]. On the other hand, the FRBS model has the advantage of being that a human end-user will be able to trace and understand the decision process and will also be able to modify the model by changing, adding, or removing antecedents in the rules of the model. This allows the model to be improved by using expert knowledge without training (or optimising) it from scratch [17].

It is understood that interpretable and augmentable characteristics of FRBS are not enough to choose that model over CNNs. However, it is a different approach that might be necessary due to internal or external policies related to the accountability of decisions and companies should be aware and choose the model that better fits their needs. An example scenario where an augmentable and interpretable model might be preferred over the higher performance of the CNNs is when there is a conflict between network assets and the BIM model created. An engineer will need to solve the conflicts, understanding why the model is built that way will help fix it, and it might also help the engineer improve the model by modifying the rules that were fired. However, a company might prefer to have the model with the highest possible performance deployed in a remote server and let the model do all the work. In this case, the CNN becomes a better option.

To evaluate and provide an example of how human expertise can be used to augment and improve the model, we made manual changes to our optimised rule base as a human expert will do. In our optimised rule base, we found that 17 rules with the antecedent *high pixel intensity* had a consequence label of *wall*. As human experts, we understand that all the wall elements in the floor plans to be segmented have *low pixel intensity*. The output label for the previous 17 rules was changed to *other,* resulting in an improvement of 1% in the mean IoU metric value for one of our testing data sets and no improvement in the other data set. This is an example of how it is possible to combine the information obtained from the optimisation process and the human knowledge of the user into an explainable model that can automate the segmentation process of the floor plan images. Furthermore, the end-user will understand how model predictions are made and make additional changes to the rules if needed. This is the main advantage of an explainable AI model: the end-users transparency and the possibility to improve the optimised model by combining it with human knowledge.

5.1 Noise Removal and Conversion to BIM

Since the segmentation models are not perfect, a noise removal process is needed to remove the incorrectly classified pixels using median blur, dilation, and erosion filters on the segmentation mask result. The goal of using these filters is to remove small groups (or single) pixels that were incorrectly labelled as wall elements. The assumption is that wall elements will consist of a large group of wall pixels, and a large group of pixels will not be affected by the noise removal filters.

Once the segmentation mask result passes through the noise removal process, we separated the identified wall elements and created a standardised BIM element for each wall. Our approach for this process assumes that walls are straight lines. The steps for this process are as follows:

1) Use a Sobel filter to identify vertical lines and horizontal lines.
2) Find the position of contours for each of the vertical and wall elements.
3) Create standard BIM element with the contour dimensions using IfcOpenShell python library [42].

This process transforms a 2D segmentation mask into a 3D BIM model. Fig. 12 illustrates the results in a virtual environment, which can be used for remote planning of the on-site network. Furthermore, it is possible to complement the resulting BIM model by fusing external information into the BIM model creation process. One example of external information sources includes the geographic location of the floor plan. The geolocation is stored in the model and can later be placed in Augmented Reality applications for on-site visualisation. Finally, there is information that currently cannot be extracted from 2D images (e.g. the height of the walls); however, the end-user can manually add it.

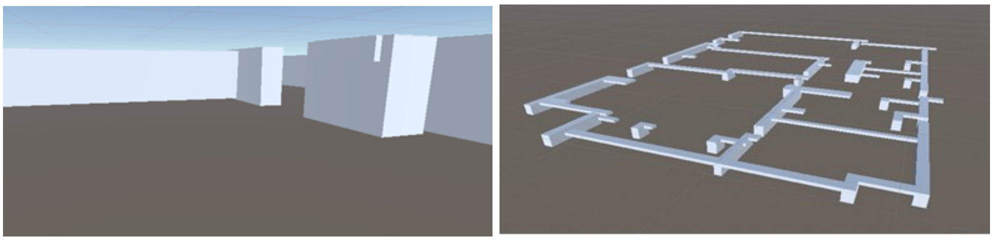


Figure 12. First-person view of the resulting BIM model using Oculus Quest VR headset (Left) and Top view of the model (Right).

6. Conclusions and Future Work

Building Information Modelling is the process of creating and managing information on a construction project across its lifecycle [43]. A Building Information Model is the digital description of every aspect of an asset and fuses information assembled collaboratively and updated during the project. For new buildings, this fusion of information is a natural part of the construction project lifecycle, where engineers, surveyors, planners, architects, and designers work together. However, for existing buildings built before BIM adoption, this is a high-cost manual process. Therefore, there is a need to transform legacy data from existing sites to BIM models for facility management and utility companies to work under the same standard and format as the construction companies do. This will facilitate the management of the complete lifecycle of a building, and it will become the first step towards creating an intelligent virtual representation of existing buildings. The first step in this digital transformation should use any data resources available, including paper-based documentation of the building structure.

This paper proposed a pipeline that covers the necessary steps to extract building structure information from 2D floor plans to create 3D BIM models.

The most crucial step within this processing pipeline is the semantic segmentation of the floor plan images. This document discussed two different approaches: one using a CNN and one using a FRBS. Each of them has its advantages and disadvantages. The deep learning approach has a mean IoU performance metric value of 99.3%, significantly better than the 97.5% mean IoU value of the fuzzy rule-based system approach. However, the CNN has the disadvantages of being a black-box model, which means that we have no (or a small) understanding of how the decision process is done, and consequently, it is impossible for a human to modify the model.

On the other hand, the FRBS approach is an interpretable and augmentable model, i.e. the end-user can understand the decision process by observing which rules are fired, and they can also modify the rules by adding, removing, or changing the antecedents, which results in a modified model. This allows human users to add expert knowledge to the system or update the model with new features without the need for the complete training process; however, the cost of this is that the explainable AI model has a lower performance metric when compared to a deep learning approach. Each company should decide which model meets its needs better. The CNN provides higher performance with a fully automated process (no human intervention), while the FRBS provides lower performance compared to the CNN but allows for human or external intervention, keeping human users in the loop and working alongside them.

Our future work includes two main goals: the first one is improving the performance metric for both segmentation models and test these models with more complex floor plans. This would allow us to include more context information in our explainable AI model by adding more antecedents in our rules. The second goal is to start fusing other information sources to the BIM model generated with the proposed pipeline, e.g. transform existing companies' assets to a BIM equivalent, and add to the existing BIM model (e.g. network assets around the building).

References

[1] J. J. McArthur, “A Building Information Management (BIM) Framework and Supporting Case Study for Existing Building Operations, Maintenance and Sustainability,” *Procedia Eng.*, vol. 118, pp. 1104–1111, 2015, doi: 10.1016/j.proeng.2015.08.450.

[2] D. W. M. Chan, T. O. Olawumi, and A. M. L. Ho, “Perceived benefits of and barriers to Building Information Modelling (BIM) implementation in construction : The case of Hong Kong,” *J. Build. Eng.*, vol. 25, 2019, doi: 10.1016/j.jobe.2019.100764.

[3] Y. Y. Al-Ashmori *et al.*, “BIM benefits and its influence on the BIM implementation in Malaysia,” *Ain Shams Eng. J.*, vol. 11, pp. 1013–1019, 2020, doi: 10.1016/j.asej.2020.02.002.

[4] Y. Arayici, “Towards building information modelling for existing structures,” *Struct. Surv.*, vol. 26, no. 3, pp. 210–222, 2008, doi: 10.1108/02630800810887108.

[5] Associated General Contractors of America, “The Contractor’s Guide to BIM.” https://www.engr.psu.edu/ae/thesis/portfolios/2008/tjs288/Research/AGC_GuideToBIM.pdf (accessed Jun. 18, 2019).

[6] Cooperative Research Centre for Construction Innovation, “Adopting BIM for facilities management,” 2007. http://www.construction-innovation.info/images/CRC_Dig_Model_Book_20070402_v2.pdf (accessed Jun. 18, 2019).

[7] J. Zak and H. Macadam, “Utilization of building information modeling in infrastructure’s design and construction,” in *IOP Conference Series: Materials Science and Engineering*, 2017, vol. 236, no. 1, doi: 10.1088/1757-899X/236/1/012108.

[8] T. A. Teo and K. H. Cho, “BIM-oriented indoor network model for indoor and outdoor combined route planning,” *Adv. Eng. Informatics*, vol. 30, no. 3, pp. 268–282, 2016, doi: 10.1016/j.aei.2016.04.007.

[9] X. Wang, P. Ed Love, and P. R. Davis, “BIM + AR: A framework of bringing BIM to construction site,” *Constr. Res. Congr. 2012 Constr. Challenges a Flat World, Proc. 2012 Constr. Res. Congr.*, pp. 1175–1181, 2012, doi: 10.1061/9780784412329.118.

[10] H. Leon-Garza, H. Hagras, A. Peña-Rios, G. Owusu, and A. Conway, “A Fuzzy Logic Based System for Cloud-based Building Information Modelling Rendering Optimization in Augmented Reality,” 2020.

[11] C. Sydora and E. Stroulia, “Augmented reality on building information models,” in *2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*, 2018, pp. 1–4, doi: 10.1109/IISA.2018.8633637.

[12] Y. C. Lin and Y. C. Su, “Developing mobile- and BIM-based integrated visual facility maintenance management system,” *Sci. World J.*, vol. 2013, 2013, doi: 10.1155/2013/124249.

[13] L. Gimenez, J. L. Hippolyte, S. Robert, F. Suard, and K. Zreik, “Review: Reconstruction of 3D building information models from 2D scanned plans,” *J. Build. Eng.*, vol. 2, pp. 24–35, 2015, doi: 10.1016/j.jobe.2015.04.002.

[14] A. Bilgin, H. Hagras, A. Malibari, M. J. Alhaddad, D. Alghazzawi, “Towards a linear general type-2 fuzzy logic based approach for computing with words,” *Internationl Journal of Soft Computing*, vol. 17, no. 12, pp. 2203–2222, May 2013

[15] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems Introduction and New Directions*, 1st editio. Prentice-Hall, Inc., 2001.

[16] H. A. Hagras, “A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots,” *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 524–539, 2004, doi: 10.1109/TFUZZ.2004.832538.

[17] H. Hagras, “Toward Human-Understandable, Explainable AI,” *Computer (Long. Beach. Calif).*, vol. 51, pp. 28–36, 2018, doi: 10.1109/MC.2018.3620965.

[18] D. Bernardo, H. Hagras, and E. Tsang, “A genetic type-2 fuzzy logic based system for the generation of summarised linguistic predictive models for financial applications,” *Soft Comput.*, vol. 17, no. 12, pp. 2185–2201, 2013, doi: 10.1007/s00500-013-1102-y.

[19] H. Leon-Garza, H. Hagras, A. Peña-Rios, G. Owusu, and A. Conway, “Type-1 Fuzzy Rule-based System using Patch-based Approach for Semantic Segmentation in Floor Plans,” 2021.

[20] H. Leon-garza, H. Hagras, A. Peña-rios, A. Conway, and G. Owusu, “An Interval Type-2 Fuzzy-based System to Create Building Information Management Models from 2D Floor Plan Images,” 2021.

[21] D. Larlus, J. Verbeek, and F. Jurie, “Category level object segmentation by combining bag-of-words models with dirichlet processes and random fields,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 238–253, 2010, doi: 10.1007/s11263-009-0245-x.

[22] T. J. Burns and J. J. Corso, “Robust unsupervised segmentation of degraded document images with topic models,” *2009 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. CVPR Work. 2009*, vol. 2009 IEEE, pp. 1287–1294, 2009, doi: 10.1109/CVPRW.2009.5206606.

[23] L. P. D. Las Heras, J. Mas, G. Sánchez, and E. Valveny, “Wall patch-based segmentation in architectural floorplans,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, no. September, pp. 1270–1274, 2011, doi: 10.1109/ICDAR.2011.256.

[24] L. P. De Las Heras, J. Mas, G. Sánchez, and E. Valveny, “Notation-invariant patch-based wall detector in architectural floor plans,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7423 LNCS, pp. 79–88, 2013, doi: 10.1007/978-3-642-36824-0_8.

[25] H. Ishibuchi and T. Yamamoto, "Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining," vol. 141, pp. 59–88, 2004, doi: 10.1016/S0165-0114(03)00114-3.

[26] D. Bernardo, H. Hagras, and E. Tsang, "A Genetic type-2 fuzzy logic based system for financial applications modelling and prediction," *IEEE Int. Conf. Fuzzy Syst.*, 2013, doi: 10.1109/FUZZ-IEEE.2013.6622310.

[27] H. Leon-Garza, H. Hagras, A. Peña-Rios, A. Conway, and G. Owusu, "A Big Bang-Big Crunch Type-2 Fuzzy Logic System for Explainable Semantic Segmentation of Trees in Satellite Images using HSV Color Space," *IEEE Int. Conf. Fuzzy Syst.*, 2020.

[28] O. K. Erol and I. Eksin, "A new optimization method: Big Bang-Big Crunch," *Adv. Eng. Softw.*, vol. 37, no. 2, pp. 106–111, 2006, doi: 10.1016/j.advengsoft.2005.04.005.

[29] H. Tang, J. Zhou, S. Xue, and L. Xie, "Big Bang-Big Crunch optimization for parameter estimation in structural systems," *Mech. Syst. Signal Process.*, vol. 24, no. 8, pp. 2888–2897, 2010, doi: 10.1016/j.ymssp.2010.03.012.

[30] R. Chimatapu, H. Hagras, A. Starkey, and G. Owusu, "A big-bang big-crunch type-2 fuzzy logic system for generating interpretable models in workforce optimization," *IEEE Int. Conf. Fuzzy Syst.*, vol. 2018-July, 2018, doi: 10.1109/FUZZ-IEEE.2018.8491662.

[31] T. Kumbasar and H. Hagras, "A big bang-big crunch optimization based approach for interval type-2 fuzzy PID controller design," *IEEE Int. Conf. Fuzzy Syst.*, 2013, doi: 10.1109/FUZZ-IEEE.2013.6622301.

[32] T. Kumbasar and H. Hagras, "Big Bang-Big Crunch optimization based interval type-2 fuzzy PID cascade controller design strategy," *J. Inf. Sci.*, vol. 282, no. October, pp. 277–295, 2014, doi: 10.1016/j.ins.2014.06.005.

[33] A. Sakalli, T. Kumbasar, E. Yesil, and H. Hagras, "Analysis of the performances of type-1, self-tuning type-1 and interval type-2 fuzzy PID controllers on the Magnetic Levitation system," 2014, doi: 10.1109/FUZZ-IEEE.2014.6891615.

[34] B. Yao, H. Hagras, D. Alghazzawi, and M. J. Alhaddad, "A big bang-big crunch optimization for a type-2 fuzzy logic based human behaviour recognition system in intelligent environments," *Proc. - 2013 IEEE Int. Conf. Syst. Man, Cybern. SMC 2013*, pp. 2880–2886, 2013, doi: 10.1109/SMC.2013.491.

[35] S. Dodge, J. Xu, and B. Stenger, "Parsing floor plan images," *Proc. 15th IAPR Int. Conf. Mach. Vis. Appl. MVA 2017*, no. December, pp. 358–361, 2017, doi: 10.23919/MVA.2017.7986875.

[36] J. H. Yang, H. Jang, J. Kim, and J. Kim, "Semantic Segmentation in Architectural Floor Plans for Detecting Walls and Doors," *Proc. - 2018 11th Int. Congr. Image Signal Process. Biomed. Eng. Informatics, CISP-BMEI 2018*, 2019, doi: 10.1109/CISP-BMEI.2018.8633243.

[37] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks."

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9351, pp. 234–241, 2015, doi: 10.1007/978-3-319-24574-4_28.

[39] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, "What's the Point: Semantic Segmentation with Point Supervision," Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.02106.

[40] A. A. Pravitasari *et al.*, "UNet-VGG16 with transfer learning for MRI-based brain tumor segmentation," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 3, pp. 1310–1318, 2020, doi: 10.12928/TELKOMNIKA.v18i3.14753.

[41] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, pp. 818–833, 2014, doi: 10.1007/978-3-319-10590-1_53.

[42] T. Krijnen, "IfcOpenShell," 2017. http://ifcopenshell.org/ (accessed Sep. 20, 2020).

[43] S. Hamil, "What is BIM?," *NBS*, 2021. https://www.thenbs.com/knowledge/what-is-building-information-modelling-bim (accessed Sep. 10, 2021).