JULY 2022 M.Sc. THESIS - ELECTRONICS AND COMPUTER ENGINEERING RABIA BAYRAKTAR

T.C.

HASAN KALYONCU UNIVERSITY

GRADUATE SCHOOL OF

NATURAL & APPLIED SCIENCES INSTITUTE

CULTURAL HERITAGE BUILDING SEGMENTATION AND CLASSIFICATION USING POINT CLOUD DATA

M.Sc. THESIS

IN

ELECTRONICS AND COMPUTER ENGINEERING

BY

RABİA BAYRAKTAR

JULY 2022

Cultural Heritage Building Segmentation and Classification

Using Point Cloud Data

M.Sc. Thesis In

Electronics and Computer Engineering

Hasan Kalyoncu University

Supervisor

Assist. Prof .Dr. Bülent HAZNEDAR

Rabia BAYRAKTAR

JULY 2022

	GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES INSTITUTE M.Sc. ACCEPTANCE AND APPROVAL FROM

Electronics-Computer Engineering M.Sc. (Master of Science) program student **Rabia BAYRAKTAR** prepared and submitted the thesis titled **"Cultural Heritage Building Segmentation and Classification Using Point Cloud Data"** defended successfully on the date of 04/07/2022 and accepted by the jury as a M.Sc. thesis.

Position	**Title, Name and Surname**	**Signature:**
	Department/University	
M. Sc. Supervisor	Assist. Prof. Dr. Bülent HAZNEDAR Computer Engineering Department Gaziantep University	
Jury Head	Prof. Dr. M. Fatih HASOĞLU Computer Engineering Department Hasan Kalyoncu University	
Jury Member	Assist. Prof. Dr. Ali Emre OZTURK Electrical - Electronic Engineering Department Hasan Kalyoncu University	

This thesis is accepted by the jury members selected by institute management board and approved by institute management board.

Prof. Dr. İbrahim Halil GÜZELBEY

Director

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Rabia BAYRAKTAR

ABSTRACT

CULTURAL HERITAGE BUILDING SEGMENTATION AND CLASSIFICATION USING POINT CLOUD DATA

BAYRAKTAR, Rabia

M.Sc. in Electronic - Computer Engineering

Supervisor: Assist. Prof. Dr. Bülent HAZNEDAR

July 2022

140 pages

The documentation and digital accessibility of historical buildings is an important issue. An integrated system that can be used in the restoration processes of cultural structures can be created by evaluating a historical building as a whole, by considering the region and period conditions in which it is located, and by realising an intelligent system that can replace existing manual systems.

In this thesis, automatic segmentation of structures using point cloud data belonging to cultural structures in Gaziantep province and mesh model creation studies were carried out from segmented point cloud data. An artificial intelligence-based learning model was used to segment historical buildings and mesh model studies were completed using poisson transformation methods.

Using the results obtained, the most comprehensive step of the studies carried out with the aim of developing the HBIM (Heritage Building Information Modelling) system has been completed.

Keyword: Artificial Intelligent, Restoration, Cultural Heritage, Segmentation, Point Cloud

ÖZET

KÜLTÜREL TARİHİ YAPILARIN NOKTA BULUTU VERİSİ KULLANILARAK SINIFLANDIRILMASI VE SEGMENTE EDİLMESİ

BAYRAKTAR, Rabia

Yüksek Lisans, Elektronik Bilgisayar Mühendisliği

Tez Danışmanı: Dr. Öğr. Üyesi. Bülent HAZNEDAR

Temmuz, 2022

140 sayfa

Tarihi binaların belgelenme ve dijital ortamda ulaşılabilirliği önemli bir konudur. Genellikle yapısal patolojiler ve sistemler tarihi yapılarla ilgili belgeleme için ve arşivleme konularında yetersiz kalmaktadır. Bir tarihi yapının bulunduğu bölge ve dönem şartları göz önüne alınarak bir bütün olarak değerlendirilebilmesi ve mevcut manuel sistemlerin yerini alabilecek akıllı bir sistemin gerçeklenmesi ile kültürel yapılardaki restorasyon süreçlerinde kullanılabilecek bütünleşik bir sistem oluşturulabilir.

Bu tez kapsamında, Gaziantep ilindeki kültürel yapılara ait nokta bulutu verileri kullanılarak yapıların otomatik segmentasyonunun yapılması ve segmente edilen nokta bulutu verilerinden mesh model oluşturma çalışmaları yapılmıştır.

Tarihi yapıların segmentasyonu için yapay zeka temelli bir öğrenme modeli kullanılmış ve mesh model çalışmaları poisson dönüşüm yöntemleri kullanılarak tamamlanmıştır.

Elde edilen sonuçlar kullanılarak, Türkiye'nin kültürel mirası olan zengin içeriğe sahip tarihi yapılarının daha akıllı ve sistematik arşivlenebilmesi ve tarihi yapı üzerinde yapılacak çalışmaların başarılı restore edilebilmesi için, HBIM (Heritage Building Information Modelling) sistemi geliştirme amacı ile yapılan çalışmaların en kapsamlı adımı tamamlanmıştır.

Anahtar Kelimeler: Yapay Zekâ, Restorasyon, Tarihi Yapı, Segmentasyon, Nokta Bulutu.

To my family who supported and believed in me..

ACKNOWLEDGEMENTS

I would like to thank my supervisors Assist. Prof. Dr. Bülent Haznedar and Assist. Prof. Dr. Ali Emre Öztürk who guided me through the completion of this project and the writing process of this thesis.

A special thanks to my family for their unconditional support throughout my education life.

This thesis is written within the TUBITAK 1001 project (Grant number: 119Y038) and the authors would like to acknowledge The Scientific and Technological Research Council of Turkey (TUBITAK) for their support.

TABLE OF CONTENT

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF SYMBOLS AND ABBREVIATIONS

ANN	Artificial Neural Network
BIM	Building Information Modelling
CNN	Convolutional Neural Network
DBA	Discontinuty Based Algorithm
DL	Deep Learning
FCN	Fully Connected Network
FCRF	Fully Connected Conditional Random Field
HBIM	Heritage Building Information Modelling
KUDEB	Heritage Conservation Department of the Gaziabtep Metropolitan Municipality
MDRNN	Multi Dimensional Recurrent Neural Network
MLP	Multi Layer Perceptron
MLPNN	Multi Layer Perceptron Neural Network
NIN	Network in Network
RE	Reverse Engineering
RNN	Recurrent Neural Network
SCP	Semantic Compositional Parts
SGD	Scholastic Gradient Descent
SVM	Support Vector Machine

VGG	Visual Geometry Group
RBM	Restricted Boltzman Machine
DBN	Deep Blief Network
DAE	Denoising Auto Encoders
R-CNN	Region Based CNN
RPN	Regional Proposal Network

CHAPTER 1

INTRODUCTION

It is inevitable that cities and structures are in constant change, just like people and keep up with the times. Historical buildings gain importance in today's conditions as long as they can maintain their cultural identity of the time they were created. In the process of acquiring the cultural identity of heritage buildings, the changes and arrangements on the buildings show the experience of the heritage buildings and add value to them. Examining and researching the value to a heritage building according to the conditions of the period and analysing the characteristics specific to the building have increased the importance of the concepts of protection and regulation of cultural structures.

The buildings reconstructed and reorganised in the cities due to need, are in a continuous process of change and development depending on the new actions caused by social and economic changes. In this process of change, studies that are handled only one-focused problem-solving methods not suitable in zoning plans cause irreversible deterioration in the architecture of a city and building (Arabacıoğlu & Aydemir, 2007).

Considering the thousands of years of historical history of our country, it is seen that there are different types of buildings with protection problems, from large-scale public buildings built with a wide variety of building materials and construction systems to examples of civil architecture built. Each of these structures is a cultural heritage for the next period and generation (Örmecioğlu & Unay, 2006).

The purpose of preserving historical buildings is to ensure that the attributes of each of the cultural heritage belonging to the period they were created are safely transferred to the future without deterioration.

It is under intense threat to the seismic movements and natural disasters that occur during the transfer of historical buildings from generation to generation, and the rapid urbanization that has emerged recently.

There are many studies under the heading of deterioration in cultural structures and their detection methods.

According to these studies (Örmecioğlu & Unay, 2006), the diversity of building materials and construction techniques used in historical buildings makes it difficult to understand the structural integrity of the building. The materials used in masonry systems do not show homogeneous properties that cause the formation of points of different strengths on the same structural element and makes it difficult to model their structural deformations (Unay, 2002).

When the restoration process of Historic Buildings was scanned retrospectively, the first principles for restoration began to be established in the late 19th century and were set forth in the "Five Principles of Contemporary Restoration" published in 1883 by Camillo Boitto. According to Boitto, “if additions or repairs are necessary, it should be based on concrete data and respect the visual integrity and form of the building” (Alanyurt, 2009).

Documentation and digital accessibility of historical buildings is an important issue. Generally, structural pathologies and systems are insufficient for documentation and archiving of historical buildings.

Restoration and maintenance of historical buildings with high architectural value are at least as complex and important as documentation. Managing cultural heritage values and structure bears great responsibility because improper restoration can result in irreversible damage. Previous research results show that 3D scanning or digitisation of heritage structures alone is insufficient to accelerate and improve the current workflow of heritage management, use and maintenance. However, BIM (Building Information Modelling), a new method for design and construction, can be the solution to address these challenges and smarter management and restoration of heritage assets.

Building Information Modelling (BIM) is a holistic building design, construction, and management method that provides interdisciplinary teamwork, supported by parametric modelling and simulation technologies, which predicts the design, construction, delivery and maintenance of physically constructed structures to be more efficient and human-oriented.

BIM embeds semantic information along with geometric information in an object-oriented approach into a 3D parametric model that can be used for effective management of building, as the BIM key product, and asset data throughout the building lifecycle, from the earliest concept design to construction, operation, and use.

With the use of the BIM approach in cultural heritage, the historic buildings can be viewed, analysed, and information exchanged by many users in a collaborative manner through integrated building lifecycle processes. To create this BIM integrated collaborative system for the restoration projects of heritage buildings, which can be considered an automatic booklet of building elements, the segmentation of the structure is required.

The concept of segmentation, which emerged in the 1970s, based on classifying and making sense of each pixel in the pictures, today emerges as the inclusion and qualification of each point in a certain class in the point cloud data. The concept of segmentation can be defined as dividing the data, which is considered as a whole, into various regions with semantic categories and grouping the existing attributes in the data (Alanyurt, 2009). In other words, segmentation is the division of an image into its constituent parts.

The image segmentation process, which aims to group pixels according to their common characteristics, minimises the diversity suitable for the selected criterion within the obtained group (Yu, et al., 2018; Chang, et al., 2015; Dube & Zell, 2011).

There is no universal segmentation method that can be applied to all images, and no segmentation method is perfect. In other words, as in image enhancement and restoration problems, the methods designed for image segmentation and the performances of these methods vary from image to image and depending on the application. Because of these difficulties, automatic image segmentation has become one of the most difficult applications of image processing (Qi, Su, Mo, & Gubias, 2017). Semantic segmentation is based on the separation of 2D images and segmentation studies have been done with 2D data for many years. However, today it has been concluded that studies with 2D data are insufficient to make sense of the data, and studies have continued with 3D data such as 3D Voxel Grids and 3D point clouds.

To segment both 2D image and 3D point cloud data, a series of operations must be performed on the raw data. Point clouds or mesh can be defined as irregular or unformatted data. For this data to be organized in a certain format, it must be converted into pictures called 3D Voxel Grids or a scene created by a series of images (Wang Z. , Liu, Qian, & Xu, 2012). 3D Voxel Grids can be defined as a 3D cube consisting of points in the simplest sense (Torlig, Alexiou, Foncesa, Querioz, & Ebrahimi, 2018).

When voxelization is completed on an image, it is considered a whole and each Voxel on the image defines a point. In this way, the picture turns into a point cloud (Point Cloud) consisting of voxels. After this transformation is completed, it can be used in deep network architecture. As it is known, whatever we define as data are concrete facts that contain information about an object and can be recorded.

While making the transformations, it is necessary to preserve the original integrity of the data and to prevent any deterioration or change in the information it contains. Many different methods have been developed to prevent this corruption that may occur in the data.

In a study by Z. Wang et al. (Wang Z. , Liu, Qian, & Xu, 2012), segmentation and detection operations were performed on an aircraft from depth image data of aircraft. According to this study, two methods called Voxel-based segmentation and Pixel-based accurate segmentation are recommended to use the segmentation of the data taken as images. Completion of these methods is performed in three steps called data pre-processing, segmentation, and detection.

A different method can be developed for each problem for segmentation and detection, or existing methods can be adapted to the solution of the problem. The important thing here is the process of separating the data taken as a picture to be used in the realization of both methods, first into 3D Point Cloud and then Point Cloud into Voxels. In this process, a 3D point cloud corresponding to the 2D depth image taken from the Kinect is obtained. Then, a voxel Grid is created on the point cloud and the plane equation, plane normal, and plane coefficient are obtained for each voxel using the least-squares estimation technique. With these values obtained, a regular format of the data is created, and the data become ready for use (Wang Z. , Liu, Qian, & Xu, 2012).

Unlike 2D image data, processing and interpreting 3D point cloud data with high density and efficiency is a more complex process. However, it is more advantageous than working with 2D image data. One of the areas, where these advantages can be used, is the segmentation of the building and classification of the basic building elements during the restoration of cultural structures.

3D point cloud data has many disadvantages besides its advantages. Because of the high density, the point cloud contains high noise and mostly open field data consists of sparse and scattered points. The methods to be used in the processes of organizing these points and determining a certain threshold value are as important as processing the data. Being able to manage this process well and choosing the right methods to be used with the right parameters directly affects the results to be obtained from the study.

Point cloud data of an object obtained using laser scanners contain the *x, y, z* coordinate values of each point belonging to the object in the simplest sense. In addition to these values, it may occasionally include *RGB (Red, Green, Blue)* colour codes or normalized x, y, and z coordinates defined as *Nx, Ny, Nz.*

With the introduction of deep learning and artificial intelligence concepts into our lives, learning architectures developed for automatic segmentation and studies in this field have gained momentum recently. Some of these studies are, AlexNet (Alom, et al., 2018) uses 8 layers of CNN, 5 conv layer , 2 FCHL (Fully Connected hidden layers) , just one FCOL.

AlexNet's closest competitor, the deep learning architecture presented by Krizhevsky et al. (Krizhevsky, Sutskever, & Hinton, 2012) lags behind AlexNet, which has 84.6% learning accuracy with 73.8% learning accuracy. These results enabled AlexNet to achieve great success in ILSVRC-2012. This study is the first study where a computer vision application that can be qualified as automatic surpasses manual methods.

VGG is a CNN network improved by VGG at the University of Oxford (Pak & Kim, 2017). VGGNET, which consists of different configurations in the proposed CNN network and is one of the Deep-CNN models, consists of 16 weight layers and its difference from its competitors is that it combines several different processes in the convolution layer in a single layer.

Since fewer parameters are used in these processes combined in a single layer, linearity between parameters decreases and the decision mechanism used in the model is more distinctive. It was first presented in (ILSVRC)-2013, and test accuracy of 92.7% was achieved with this method (Pak & Kim, 2017).

PointNet (Qi, Su, Mo, & Gubias, 2017) ,which is an end-to-end deep neural network architecture that allows working point-blank on the Point dataset and can be used for classification, part segmentation, and semantic segmentation using Point Cloud, is one of the pioneering studies in this field.

In this study , in which classification and segmentation processes were performed on 3D point cloud dataset of the selected heritage buildings in Gaziantep in Turkey, 90%-92% test accuracy and 98% training accuracy were achieved. However, the authors, who stated that PointNet could not capture local geometries over time, presented the PointNet++ architecture as a new study (Qi, Yi, Su, & Guibas, 2017).

In this study developed by Yi et al, a hierarchical grouping was made to identify local features. More details on the point cloud can be captured using point-to-point metric calculations. According to the results obtained, PointNet++ performs better than PointNet.

In this study, the segmentation of the historical structure, which is the most comprehensive step to create a BIM model, will be made using AI and DL solutions, and the results will be examined.

By using the above-mentioned segmentation methods and the PointNet architecture, which has a wide place in the literature, the *x*, *y*, *z*, and *Nx*, *Ny*, *Nz* coordinates of the 3D poin cloud data were processed. For the classification and segmentation of historical buildings with 5 labels, a deep learning network to be trained using x, y, z coordinates, and their normalized values from 3D point cloud data has been created and trained.

The results obtained were recorded by repeating with different tests, validation, and training data. In the PointNet network used, laser scanning data of Gaziantep historical buildings, which have been deformed over time, have been used, unlike the original study.

Because of the need for a large dataset for the training of Deep Learning Networks, synthetic data was created with the methods found in the literature to expand the dataset and included in the learning network.

In this study, 3d mesh model creation studies were also conducted from segmented point cloud data. Studies on creating 3d models in BIM models are summarised as follows: The HBIM model is semantically expressed as a 3d solid model. This model should contain parametric and semantic information. It is necessary to use point cloud data to create a BIM model of a structure, and this point cloud data is obtained using TLS (Terrestrial Laser Scanning) (Arayici, 2008).

Hichri et al. detailed the preprocessing steps of BIM modeling from the point cloud (Hichri, Stefani, De Luca, Veron, & Hamon, 2013). Macher et al. similarly present a semi-automatic approach for creating 3D models of historical buildings from point clouds (Macher, Landes, Grussenmeyer, & Alby, 2014) . They separate the buildings according to their structural features and states that an evaluation should be made according to these features.

Because of a Building Information Model is required to be parametric, Dore and Murphy (Dore & Murphy, 2013) proposed a method called GDL (Geometric Description Language).This method offered illustrative ideas for BIM model building and reconstruction of structures (non-primitive shapes are modelled using NURBS, meshing and Boolean operations).

Similarly, Oreni et al. (Oreni, et al., 2014) introduce a workflow using NURBS and vector profiles to model the geometry of historic monument components. In addition, Garagnani and Manferdini (Garagnani & Manferdini, 2013) present a Revit plug-in (GreenSpider) that matches a spline to imported 3D points.

Unlike other systems in BIM systems; The difference between the methods and models used in modeling the features of today's buildings and historical buildings is considered as an important issue. Fai and Sydor (Fai & Sydor, 2013) thus defend for differentiate the ‘characteristic ’ and the ‘specific’, and constitute ‘characteristic (typical)’ BIM objects that can be deformed to acquire ‘certain (specific) ’ ones for a given historic monument.

While this approach addresses the concerns of heritage experts regarding the unicity of objects in historic buildings (and therefore in HBIM models), it raises the need to create libraries of 'typical' BIM objects that can be deformed (Brumana, Oreni, Raimondi, Georgopoulos, & Bregianni, 2013).

Parametric as-built building models, consisting of millimetre accurate vector models are very rare. The time required to model an object to this level of detail renders it unfeasible for all but the most critical elements in a project.

Mechanical parts design and industrial design are modelled to this level of detail, but existing buildings are mostly modelled to a higher-order detail and the parametric information included is limited (Arayici, et al., 2017; Xiong, Adan, Akinci, & Huber, 2013).

The manual modelling process is flawed and inaccurate. Having an accurate and fast way of capturing detail in a parametric environment would enable an analysis of specific attributes of materiality, geometry and interrelationship with other building elements. By association, other parametric values can be implanted in these models. This will in turn enable classifiers to be searched for in models, similar to the machine learning algorithms that identify features in images (Pu & Vosselman, 2009; Arayici, et al., 2017).

Considering that the correct segment size is achieved, the data can be further processed towards BIM objects modelling by meshing the points up and using the neighbouring ratio of vertices via Voxelizing the different types of interdependencies to identify building components such as openings for instance. All of this takes place in 3D space.

The relationship between the semantic feature extraction and the points belonging to the structure in the direction of gravity can be determined. However, as the number of scanned and interrelated points increases, accuracy becomes an important problem in semantic inference (Pătrăucean, et al., 2015; Wang, Cho, & Kim, 2015).

Since 2003, the principal investigator of this study undertook research on point cloud data processing and modelling for cultural heritage and refurbishment.

The focus of the previous research was to build a concept of the building data integration system based around the 3D laser scanning technology to develop VR models of heritage buildings for environmental and context analysis. This involves producing CAD models and integrating with other systems such as GIS, VR displays, and nD modelling to enhance the refurbishment and renovation process for cultural heritage (Lee, et al., 2003).

The major themes and case studies in the research have been: (i) heritage documentation for St Fagan's Museum in Cardiff; (ii) 3D VR modelling for buildings and surroundings (Arayici, Hamilton, & Gamito, 2006); (iii) forward & reverse prototyping for buildings (Arayici, Hamilton, Gamito, & Albergaria, 2005); (iv) building feature line extraction for building surveying and refurbishment (Arayici, Hamilton, & Gamito, 2005); (v) environmental and historical simulation of the Waterloo battlefield; and (vi) IFC (Industry Foundation Classes) data production from point cloud data for the building data integration system (Arayici, 2007) by a semi-automated approach developed in the research (Arayici, 2008).

In this study, the segmented entities achieved through the deep learning approach are dealt with separately in the 3D environment for 3D BIM extraction. Mathematical modelling for BIM extraction uses object recognition and constructive solid geometry methods. As a result, an accurate 3D BIM model of the heritage building is prodeced.

CHAPTER 2

BACKGROUND & LITERATURE REVIEW

2.1. SEGMENTATION & CLASSIFICATION

2.1.1. 3D Geometric Data

In typical convolutional architectures, segmentation with 2D images is a research area that has been the subject of many studies to obtain high accuracy in estimation processes by calculating weights.

In a study by Dube et al. (Dube & Zell, 2011), one of these studies, an airplane detection study was conducted using the Randomized Hough transform method on the dept image data.

In this study (Dube & Zell, 2011), Hough transforms was used for detection and a noise model was created to find the metrics to be used in this transform. With the mobile application in which the created model is used, the aircraft in a depth picture can be detected in less than a millisecond. In this study, it is sufficient to use 2D depth image data only for detection without segmentation.

However, recently, researchers have been conducting research and studies on the use of image Grids and 3D voxel data, since studies with 2D images are insufficient due to the lack of depth. Point Clouds or Meshs can be defined as irregular or unformatted data. In order for this data to be organised in a certain format, it must be converted into pictures called 3D Voxel Grids or a scene created by a series of images (Qi, Su, Mo, & Guibas, 2017).

3D Voxel Grids can be defined as a 3D cube consisting of points in the simplest sense (Wang Z. , Liu, Qian, & Xu, 2012). When Voxellization is completed on an image, it is considered a whole and each Voxel on the image defines a point. In this way, the picture turns into a point cloud (Point Cloud) consisting of voxels. After this transformation is completed, it can be used in deep network architecture.

As it is known, what we define as data are concrete facts that contain information about an object and can be recorded.

While performing the transformations described above, it is necessary to keep the unique integrity of the data and to prevent any deterioration or change in the information it contains. Different methods have been developed to prevent these corruptions that may occur in the data.

In a study by Z. Wang et al. (Wang Z. , Liu, Qian, & Xu, 2012), a study was conducted in which segmentation and detection operations were performed on an aircraft from depth image data of aircraft. According to this study, two different methods called Voxel-based initial segmentation and Pixel-based accurate segmentation are recommended to use the segmentation of the data taken as images. Completion of these methods is conducted in 3 steps called data preprocessing, segmentation and detection. A different method can be developed for each problem for segmentation and detection, or existing methods can be adapted to the solution of the problem. The important thing here is the process of separating the data taken as a picture to be used in the implementation of both methods, first into 3D Point Cloud and then Point Cloud into Voxels. In this process, the 3D Point Cloud corresponding to the 2D depth image obtained from the Kinect is obtained. Then, a voxel Grid is created on the point cloud and the plane equation, plane normal and plane coefficients are obtained for each voxel using the least squares estimation technique. With these obtained values, a regular format of the data is created and the data becomes ready for use.

The process of creating regular formats of a data can be performed using different methods. In a study by E. Torlig et al. (Torlig, Alexiou, Foncesa, Querioz, & Ebrahimi, 2018), a new method is proposed to perform voxellation on the data received as Point Cloud. With this method, it is aimed to obtain a set of points with a regular geometric structure. In the method proposed here, each voxel is rearranged as 10-bit depth points and a new structure is created within the framework of certain rules. Using the newly created structure, a plane is found in real time in the three dimensions (3D) Point Cloud data and converted to a two dimensions (2D) picture.

2.1.2. Deep Learning

Deep learning, which is one of the machine learning method, is learning networks used to estimate a specified parameter after learning using a dataset. Although deep learning and machine learning are based on training a network, they are not the same thing. They are often referred to together with artificial intelligence, but these three fields of study are different from each other.

Artificial intelligence is a field of study that includes deep learning and machine learning. Over time, it has been divided into sub-branches with the diversification of artificial intelligence problems and solutions. Solution suggestions for problems in artificial intelligence and its sub-branches can be improved and developed day by day. In this respect, it is similar to human intelligence.

The concept of artificial intelligence, which emerged in the 1950s, can be defined as machines that can frequently improve themselves, since they can learn from error using open source datasets and learning strategies. In the 1980s, the concept of machine learning emerged to classify certain datasets and make class predictions. There are two types of learning strategies that are frequently mentioned in the literature, supervised and unsupervised learning.

Supervised learning is learning from labelled data. In this learning method, information about input and output data should be annotated. The accuracy of the parameter predicted by the model should be checked.

Unsupervised learning is the process of learning from unlabelled investigations. it doesn't need any training of the result data type. The algorithm itself draws conclusions from data. it is waited to discover itself.

Until the 1990s, machine learning and artificial intelligence studies could not develop. Deep learning studies started with the emergence of the concept of data mining in the early 1990s. Deep learning networks created by the use of machine learning and artificial intelligence techniques are still very popular today. It is possible to see Deep Learning applications in every field. Intelligent systems can be easily produced using deep learning algorithms, especially in tasks where the number of data is large and manual construction takes a lot of time.

Unlike artificial intelligence, deep learning applications are used in areas that require more complex and detailed learning networks. Segmentation and detection applications are among the best examples that can be given to these areas.

In deep learning, the features to solve the problem are used as input data. Deep learning models, which are known to be very successful in solving difficult problems, require working with large data sets.

The features of the data set pass through more than one layer, and as a result of each layer, more meaningful information is obtained from the data. There are three main types of deep models. These are: Multilayer perceptron, Convolutional Neural Networks, Recurrent Neural Networks (Beyaz.net, 2022).

2.1.2.1. Multilayer Perceptrons

Multilayer Perceptrons (MLP) has emerged because of the studies done to solve the XOR problem. This model developed by Rumelhart et al. is also called the 'Back Propagation Model' or the 'Error Propagation Model' because it spreads the error to the network.

This method uses a learning method called the 'Delta Learning Rule' (Rumelhart, Durbin, Golden, & Chauvin, 1995).

MLP works particularly well in classification and generalisation situations. The MLP algorithm is started with the input layer, which provides information flow to the hidden layer.

Information coming to the hidden layer is passed through all layers, respectively, towards the output layer. The number of hidden layers varies according to the problem, at least one, and is adjusted according to the need. The output of each layer becomes the input of the next layer. Thus, the output is reached. Each neuron is connected to all neurones in the next layer. The number of these neurons connected to each other is also determined according to the problem. In the output layer, the data from the previous layers are processed and the output layer is obtained. Activation functions are used to prevent the loss of information transferred between layers or to prevent unnecessary information transfer.

These functions can be linear, sigmoid or tanh, which are often mentioned in the literature. There are many alternatives to these functions.

Learning in a multilayer network is based on the Delta Learning Rule. Labeled data is required for the network to learn, and learning is done over the data. The learning process in Backpropagation Artificial Neural Networks is the process of finding the function that matches the input values in the sample set with the output values. The learning method of the system generally consists of two stages. The first stage is forward calculation while the second stage is back propagation. In the output oriented computation in other words forward propagation phase, the input given to the system reaches the output by passing through the intermediate layers. The net input is calculated by summing the inputs to each processing element. The calculated net input is passed through the activation function and the output of the current processing element is found and the output value is sent to the processing elements in the next layer. By repeating these processes, outputs are obtained from the last output layer. With the output from the network, the first stage of learning is completed.

The second stage will be the distribution of the error. If the expected output value and the result obtained are different from each other, it means that the model has made a calculation error. To eliminate this error, the error is distributed to the weight values in the backward calculation phase and the error value is reduced at each iteration. The weight values given to the system randomly at the beginning are updated in each iteration by distributing the errors to the weights.

Multi-Layer Perceptron (MLP) has rapidly increased the interest in Artificial Neural Networks. With MLP, a new perspective has come to ANN studies. Propagation and back propagation algorithms, which are used effectively in classification and identification studies especially in the medical field, are among the methods that often lag behind other methods in the field of neural networks.

In a study by Naraei et al. (Naraei, Abhari, & Sadeghian, 2016), classification study using 303 heart patient data was performed using Support Vector Machine (SVM) and Multi Layer Perceptron Neural Networks (MLPNN) methods and the results were compared.

According to this study (Naraei, Abhari, & Sadeghian, 2016), 80.5218% Acc and 0.2378 Rmse were obtained with MLP, while 84.4884% Acc and 0.2226 Rmse were obtained with SVM in the disease in which 5 different classifications were made.

In a study by Jusman et al. (Jusman, Firdiantika, Dharmawan, & Purwanto, 2021), skin cancer classification was performed using the HAM10000 dataset (Mader, 2021). For this classification, one of the deep learning methods, CNN, and one of the Neural Network methods, MLP, were compared. In the trained MLP network, 4 hidden Layers were processed and 60.0% - 71.9% test Acc was obtained.

In a study by Feghi et al. (Feghi, Tahar, & Ahmadi, 2011) , an Artificial Neural Network (ANN) network was trained using a back propagation algorithm. An identification rate of up to 100% was obtained in the network trained with 400 different fingerprint images of 10 different people. Features of fingerprint images are used to train the ANN network consisting of 10 layers, including input and output. Some image processing methods are included in the study for feature extraction.

When studies using Multi-Layer Perceptron (MLP) methods are examined (Feghi, Tahar, & Ahmadi, 2011; Jusman, Firdiantika, Dharmawan, & Purwanto, 2021; Naraei, Abhari, & Sadeghian, 2016), it can be concluded that this method could not show the performance it showed in identification and classification.

Deep learning (DL) methods are more appropriate to use in order to achieve high performance in classification processes, since the dataset is large and the classification labels are more.

2.1.2.2. Convolutional Neural Networks (CNN)

CNN is a deep learning method that can process 2D image data by taking an image sequence as input. This method is generally used for processing datasets where inter-pixel relations are important. As in all artificial intelligence models, this method has a layered structure. Features of an image are processed in these layers. Visual passing through these layers, called convolutional Layer, Pooling and Fully Connected, is used for training in the deep learning model by being subjected to different processes.

Since unstructural (irregular) data can be used while creating CNN models, it is a method that has been extensively studied in deep learning networks recently (Albawi, Mohammed, & Al-Zawi, 2017).

In a study by Y. Lyu et al. (Lyu & Huang, 2018), a CNN model capable of road segmentation was created for autonomous vehicles. In this model, a model that can be segmented using the road data obtained from the front camera of a vehicle is trained.

In this system, in which the CNN network is supported by Gated Recurrent Units (GRU), it is seen that compared to other methods, it is time-consuming, requires less memory and uses less calculations to reach the result. In this study (Lyu & Huang, 2018), a 4-layer CNN network trained with image data taken as a 60x150x5 matrix was created. A 60x256 matrix was obtained as output from the CNN network. The proposed CNN-GPU method with segmentation f1 score of 86.91% and average precision of 81.11% has proven its success.

In a study by Saez et al., road segmentation was performed with a CNN-based network to be used in autonomous vehicles. It is seen that the better segmentation is done on the road that is divided into 5 different categories by semantic segmentation, the more accurate the road that the autonomous vehicle defines itself. As it is known, the more data CNN networks are trained with, the better the results. In this study, 2.975 for train, 500 for validation, 1.525 for test data were used. In the study using an 8-layer CNN network, high performance was achieved with 97.1% IoU (Sáez, et al., 2019).

CNN studies have been the subject of not only the studies of researchers interested in computer science, but also studies in the field of medicine. In a study by Kayalıbay et al. in 2017, a CNN network with a 3D filter was trained using hand and brain MRI data. The greatest advantage of the trained CNN network to other studies in its field is that it can be detected in other anomalies affecting the central nervous system with the CNN network proposed in this study, while other studies focus on the main organs and soft tissue (Kayalibay, Jensen, & van der Smagt, 2017).

2.1.2.3. Recurrent Neural Networks (RNN)

RNNs are a kind of Deep Learning constructs that are often used to predict the next step. The biggest difference from other deep learning structures is that they remember. An RNN model is based on an iterative invocation structure. That is, the input of the result obtained in each iteration is the result obtained in the previous iteration. In this way, a memory structure was created. RNN can be compared to a traditional neural network that can be trained using a back propagation algorithm (Graves, Generating sequences with recurrent neural networks, 2013). Recurrent Neural Networks (RNNs) are one of the main methods used in classification and identification studies.

In a study by Habi et al. (Habi & Messer, 2020; Habi & Messer, 2018) developed a deep learning network capable of rain detection using gated Recurrent Units (GRUs). In the study using The Swedish Dataset (paperswithcode, 2020) and Dual Region Dataset (paperswithcode, 2020), the data were divided into 2 different groups, 80% training and 20% test data, by combining the two datasets.

The results obtained using 2 different activation functions (Sigmoid, Softmax) were compared. According to these results (Habi & Messer, 2020), 95.7% Acc was obtained with the sigmoid function in the validation data set and 95.4% Acc was obtained with the Softmax function. In both cases, the network created using RNN showed classification success over 90%.

In a study by Jayakumar et al. , an RNN network was constructed for disease classification using the picture of plant leaves. Pictures of watermelon leaves were used for the training of the created network. The images to be used for the training of the network were pre-processed and image acquisition, image processing and image segmentation operations were performed on the images. A high success rate was obtained from the trained network, with validation acc 98.7% and test acc 98.5% (Jayakumar, Elakkiya, Rajmohan, & Ramkumar, 2020).

In a study by Dong et al. (Dong, Zhang, & Shao, 2019), a hybrid model based on CNN-RNN was created. CNN has been frequently used in image transformation and image classification studies recently. In this study, in addition to the effective use of this success that can be achieved with CNN, the 2D wavelet transform layer (e 2D wavelet transform layer) is included in the model created.

With the inclusion of this layer, wave transform methods used in signal processing studies are based on RNN. For wave transformation, a 2-layer RNN network is created and its output is given as input to a 4-layer CNN network.

From the model trained using CIFAR 10 (Kaggle, 2020) and CIFAR 100 (Kaggle, 2020) datasets, 93.9% test acc was obtained in the CIFAR 10 dataset and 73.8% test acc in the CIFAR 100 dataset.

When studies of Recurrent Neural Networks (RNNs) are examined (Habi & Messer, 2020; Habi & Messer, 2018; Jayakumar, Elakkiya, Rajmohan, & Ramkumar, 2020; Dong, Zhang, & Shao, 2019), it is seen that it exhibits high performance in classification and definition-based studies.

2.1.3. Segmentation

Separation of an image or processable data into meaningful parts can be called segmentation. In segmenting a data, the features of that data and the features that best describe the data should be determined well (Gonzalez & Woods, 1993).

In other words; Segmentation is the division of the image into its constituent parts. The image segmentation process, which aims to group pixels according to their common characteristics, aims to minimise the diversity suitable for the selected criterion within the obtained group (Dunn, 1973; Bezdek, 1981; MacQueen, 1967). As the methods for segmenting different data types differ, even the methods for the same data types can differ. This indicates that there is no general segmentation method that will work for all data types. The features of the data to be used in segmentation, how well these features can be defined in the network and their usability affect the segmentation performance.

That is, as in image increasing and restoration problems, the methods designed for image segmentation and the performance of these methods vary depending on the image and the application. Developing a system capable of automatic segmentation for the solution of a problem is one of the most difficult issues especially for image processing researchers. (Huang, Schreiber, & Tretiak, 1971).

Algorithms for segmentation of monochrome images are based on the use of one of the two basic properties of grey level values. These basic features are discontinuity and similarity (Gonzalez & Woods, 1993).

2.1.3.1. Discontinuity-Based Approach for Segmentation

Discontinuity-based segmentation algorithms (DBA); It is based on being able to detect discontinuities such as isolated points, thin lines or picture edges (with gray level values suddenly changing) using similar masks as in low and high filtering (Gonzalez & Woods, 1993). The main purpose of such algorithms is to find the separation points, lines and edges on the image.

This method, which is generally used in the field of medicine, is based on the creation of an algorithm for the detection of many diseases, based on the changing color levels in the pictures.

According to Baykara et al. , a method was proposed for the detection of liver necrosis resulting from severe deterioration of liver functions. Because of this study, it is aimed to improve a Non manual (computer-aided) system that can define the necrosis area in the tissue and calculate the quantitative value of the necrotic area for the specialist who will make the determination. The system called CBT was tested on mice and the highest performance of necrosis in the liver was obtained with 64.29% (Baykara, Ertürkler, Gül, & Harputoğlu, 2004).

In a study by Kumar et al. (Kumar, Arthanari, & Sivakumar, 2012), methods based on discontinuity-based segmentation were examined and performance comparisons were made. The study (Kumar, Arthanari, & Sivakumar, 2012) where Point Detection, Edge detection methods were compared (Gonzalez & Woods, 1993; Canny, 1986), Canny Edge (Canny, 1986) Detection method, which is one of the edge detection methods, gave the best results.

2.1.3.2. Similarity-Based Approach for Segmentation

Similarity-based segmentation algorithms; based on thresholding or region splitting and merging. Image segmentation based on similarities in grey level values is also known as Region segmentation (Huang, Schreiber, & Tretiak, 1971).

This method can be applied to both static (static) and dynamic (time-varying) images. It can be used in conjunction with operations commonly heard in image processing such as Thresholding, Growing, and Split- and Merge (Gonzalez & Woods, 1993).

In a study by Huart et al. (Huart & Bertolino, 2005), 200 images with 300x300 pixel dimensions were segmented with 3 different labels (animal, ground and tree). Because of the studies, a success of 98.20% was achieved.

In a study by Cyr et al. (Cry & Kimia, 2004), a method was proposed to obtain 2D images from 3D images and then segment the object from different angles using images. In this method, while creating the dataset, a picture of the 3D object in every 15' rotation was taken and stored. The segmentation process was successfully performed on 12 different objects and 24 different angles of each object. As it is known, image processing and signal processing are two different fields of study where common methods are used. Similarity-based segmentation approach is also frequently used in signal processing studies.

2.1.4. Data Types

2.1.4.1. 2D images Data

GoogLeNet is a 22-layer deep convolutional neural network, a variant of the inception Network, a Deep Convolutional Neural Network developed by researchers at Google (Szegedy, et al., 2015). Presented at the ImageNet Large-Scale Image Recognition Competition 2014 (ILSVRC14), the GoogLeNet architecture was presented as a solution with a 93.3% success rate for computer vision tasks such as image classification and object detection.

The GoogLeNet architecture consists of 22 layers, and some of these layers consist of 9 initial modules. These modules are defined in 3 layers, namely Network in Network (NiN) layer, a Pooling operation, a large-sized Convolutional layer, and small-sized Convolutional layer. Parallel calculations are made in all of the specified modules. In this way, the number of parameters is reduced, and memory and time are saved (Szegedy, et al., 2015). ResNet is a network structure proposed by He et al. in 2015 and is presented as a method that outperforms its competitors with a 96.4% success rate in the ILSVRC-2015 classification task.

At the same time, the network depth consisting of 152 layers draws attention in the model, which is the first in ImageNet detection, ImageNet localization, COCO detection and COCO segmentation tasks. In this deep learning network, in each layer, both the learned output from the previous layer and newly added parameters are taken as input. In this way, a complex structure with high learning performance is obtained (He, Zhang, Ren, & Sun, 2016).

In a study by Graves et al., a learning method that can be defined as Sequential RNNs is presented as the Multi-Dimensional Recurrent Neural Network (MDRNN) architecture. In this architecture, the spatial transmission paths used in the RNN layers are defined as a repetitive spatial connection as a result of a series of operations (Graves, Fernández, & Schmidhuber, 2007). ReNet, created using this architecture, uses RNNs defined as an array, rather than multidimensional RNNs.In this way, the number of RNNs is equated to linearly scaled spatial transmission paths (d) at each respective layer. In this way, each convolution layer in the RNN can analyze an image both horizontally and vertically (Visin, et al., 2015).

2.1.4.2. 3D Point Clouds Data

Deep Learning (DL) has been used successfully in 2D images in many fields. In recent years, deep learning (DL) studies on 3D Point Cloud have become a wide research area in order to determine whether deep learning shows the same success in irregular data. Studies on 3D Point Cloud can be based on 4 different methods: Voxelization-based (Le & Duan, 2018; Wang P. S., Liu, Guo, Sun, & Tong, 2017), multi-view-based (Su, Maji, Kalogerakis, & Learned-Miller, 2015; Le, Bui, & Duan, 2017), graph-based (Bronstein, Bruna, LeCun, Szlam, & Vandergheynst, 2017; Yi, Su, Guo, & Gubias, 2017; Simonovsky & Komodakis, 2017) and set-based (Qi, Su, Mo, & Gubias, 2017; Qi, Yi, Su, & Guibas, 2017).

The method aiming to obtain voxel-based grid sections from point cloud data is known as voxelixation-based in the literature. Thus, 3D Convulation data could be processed in a similar way to 2D painting (Wu, et al., 2015; Maturana & Scherer, 2015). In this way, information loss of the data is reduced and cubic calculation techniques can be used on the data.

OctNet (Riegler, Ulusoy, & Geiger, 2017) and Kd-Net (Klokov & Lempitsky, 2017), which are created by using the advantages of the voxelization-based method, are two different methods that reduce the computational cost. In these methods, the Voxel, which is expressed as empty in the data allocated to the Voxels, is not included in the calculation, thus saving both time and memory.

The multi-view-based method defines the 3D point cloud as a series of images taken from different angles. The number of images taken from different angles, the image distribution, and how many degrees the angle will be are not fixed. Therefore, different parameters are required for each study. It is often described as an indefinite method (Su, Maji, Kalogerakis, & Learned-Miller, 2015; Le, Bui, & Duan, 2017; Kalogerakis, Averkiou, Maji, & Chaudhuri, 2017).

Graph-based method is a CNN-based method that processes the neighborhoods of each point in the point cloud in planar space and then creates the final planar space graph (Bronstein, Bruna, LeCun, Szlam, & Vandergheynst, 2017; Yi, Su, Guo, & Gubias, 2017).

In a study by Shen et al. (Shen, Feng, Yang, & Tian, 2018), 3D point clusters are defined as 3D data stacks whose correlation can be calculated, which can respond jointly to neighbouring points and can learn.

The 2 methods named ECC (Simonovsky & Komodakis, 2017) and SPG (Landrieu & Simonovsky, 2018) are methods based on the graph-based method that proposes to create Convulation filters using graph weights. Since these methods can only operate on predefined weights, they have been effective only on certain data structures. Therefore, it is not a recommended method in the literature.

According to Wang et al., the Set-based method is a method that can be used directly with point cloud data. However, it is a method that is not preferred in semantic segmentation studies since it ignores the neighbourhood relations that contain structural information between the points (Wang, Huang, Zhang, & Shan, 2019).

2.1.5. Segmentation and Classification Applications of Cultural Building

Mesh and solid model generation studies from 3D point data for the analysis of historical buildings; It has been a field of study that has attracted the attention of researchers, archaeologists and curators. There are studies aiming to record the changes of a building over the years, to use it in modernisation processes and to make it an integrated system.

Saygı et al. base this perspective, which is called Information Management Strategy in the field of architecture, on 3 basic elements: segmentation, structuring hierarchical identification and meaning enrichment (Saygı & Remondino, 2013).

Grilli et al. presented a study in 2018 that can be used in architectural and archaeological studies and can automatically bring semantic explanations to structures of historical value. In this study, a research was conducted to document the information about the historical building, to distinguish different architectural techniques, and to collect and analyse previous restoration works. This study defines the characteristic of a structure in the best way with a limited number of input data. In the results obtained using seven different machine learning methods, the classification performance of the study was 69% (Grilli, Dininno, Petrucci, & Remondino, 2018).

Sithole proposes a system that can semi-automatically detect the bricks that make up the wall in a building. In this study, triangulation, reflexion and RGB triplets to effect a proximity-based segmentation was performed on the data obtained from the laser scanner as a 3D point cloud. In the method based on the weighted proximity segmentation method, considering the wall thickness and depth, 2000 points were processed for each brick. This study was completed with 750 test data (Sithole, 2008).

Riveiro et al. present a segmentation approach for both fixed and mobile devices that enables automatic segmentation of wall blocks from a 3D point cloud obtained with LiDAR technology. In the study, in which automatic segmentation was performed using image processing methods over the point cloud, the morphological analysis used to assemble the valuable parts for the structural analysis and inference of masonry structures was successfully performed (Riveiro, Lourenço, Oliveira, González-Jorge, & Arias, 2016).

Oses et al. performed classification on a masonry wall using machine learning classifiers, Support Vector Machines and Classification trees. In the study where K-Nearest neighbour (K-NN), Support Vector Machines (SVM), Naive Bayes (NB) and Classification Trees methods were used, 19 Features belonging to 3 different classes and the highest 80.2% acc were obtained using K-NN, 28 Features SVM achieved the best results with 75.6% acc (Oses, Dornaika, & Moujahid, 2014).

Tang et al. (Tang, Huber, Akinci, Lipman, & Lytle, 2010)conducted research on automated Building Information Models (BIM) generation processes used in Civil engineers and computer science. According to this study, there are 3 basic steps to be followed while creating a BIM model of a structure. These are the geometric modelling, Object identification and object relations are modelling. In the study, the most efficient methods that should be used in the BIM creation process, data preparation, data labelling, working with 2D and 3D data, storing each data in different folders according to a certain standard and listing it when desired, were discussed in the study (Campbell & Flynn, 2001; Mian, Bennamoun, & Owens, 2005; Golovinskiy, Kim, & Funkhouser, 2009; Matei, et al., 2006), which have gained a wide place in the literature. It has been decided that the most important factors in BIM systems are the correct analysis of the data, its correct classification, its presentation to the user in the correct hierarchical order, and the correct management of the space needed in the storage of the created system.

2.1.6. Overview Studies

PointNet (Qi, Su, Mo, & Guibas, 2017), which is an end-to-end deep neural network architecture that allows working immediately on the Point dataset and can be used for classification, part segmentation and semantic segmentation using Point Cloud, is one of the pioneering studies in this field. In the PointNet architecture, the input layer consists of a set of MLPs that use the properties of point clouds.

In the layer known as the Max Pooling layer, the symmetric properties of the input data are used, the input permutation calculations are made and the global values of the data are calculated. Fully Connected Layers, on the other hand, performs label prediction and classification.

However, the authors, who stated that PointNet could not capture local geometries over time, presented the PointNet+ + (Qi, Yi, Su, & Guibas, 2017) architecture as a new study. In this research, a hierarchical grouping was made to identify local features. More details on the point cloud can be captured using point-to-point metric calculations. According to the results obtained, PointNet++ performs better than PointNet.

In the part segmentation study by Yi et al., A method for object segmentation was proposed over Point cloud data belonging to 16 different categories containing different numbers of data. According to this method, different regions of the object were determined in each object category and the system was trained in this direction. Deep learning methods using a total of 95,000 data were supported by different Framework methods and a structure called Scalable Active Framework was created. With this part segmentation method, an F1 score varying between 85% and 95% was obtained in 16 different categories (Yi, et al., 2016).

In a study by Wang et al. , an object-based segmentation method was proposed. This method uses image as input data. Data belonging to 5 different animal categories (horse, sheep, cow, cat, dog) were used in the study, where part segmentation was aimed through the image. A two-stream fully convolutional network (FCN) was formed, which was supported and trained with the method known as semantic compositional parts (SCP) for the segmentation of the head, body, foot and tail regions of these animals. The proposed new method is called Efficient Fully Connected conditional random field (FCRF). With this method, the average IoU value of the 5 categories was obtained as 78.64% in the segmentations made on the picture (Wang, et al., 2015). In the study of the relationship between architectural features by using Covariance features of architectural structures , a correlation equation was created between the size of the building and the radius of the building elements using the Random Forest algorithm (Grilli, Farella, Torresani, & Remondino, 2019).

In another study created by further deepening this study, a similar correlation equation was created and a classification based on similarity was made between similar structural elements in different structures (Grilli & Remondino, 2020).

In addition to both studies, Murtiyoso et al. argued that existing data should be labeled and processed on labeled data in order to enhancement the accuracy in machine learning methods (Murtiyoso & Grussenmeyer, 2020).

In recent years, the segmentation process has yielded remarkable results in studies with 3D datasets (Xie, Tian, & Zhu, 2020; Zhang, Zhao, Chen, & Lu, 2019). One of the biggest problems in segmentation studies using Deep Learning (DL) methods with 3D datasets is the need to work with large datasets. The lack of sufficient data in the area to be studied prevents the Deep Learning model to be created from being successful enough in the prediction process most of the time. To eliminate this problem, studies have been conducted using synthetic data. In these studies, it was especially emphasised that the generated data should have the features closest to the original data, otherwise it was clearly stated that the model would yield incorrect results (Pierdicca, Mameli, Malinverni, Paolanti, & Frontoni, 2019; Griffiths & Boehm, 2019).

2.2. MESH MODEL GENERATION

Along with the use of the Point Cloud concept in computer science, the concept of mesh is frequently used in 3D model design, product prototyping, and reverse engineering (RE) applications. The Mesh model, in its most general terms, is the process of creating a connection (Surface) between each point forming the point cloud and its neighbourhoods by using a certain function. When existing methods are examined, the use of triangle (Triangular) mesh and quadrilateral (Quadrilateral) mesh are the most popular polygonal mesh creation methods during the 3D solid model creation phase (Williams, et al., 2019).

Many studies have shown that studies with a quadrilateral meshing method give better results than Triangular mesh in texturing, simulation and imaging techniques consisting of a finite number of elements (Berger, Levine, Nonato, Taubin, & Siva, 2013). However, studies using historical buildings in the areas where the method is more useful on the point cloud have not gained much space in the literature.

The method called Surface Reconstruction may cause failures in the process of creating a surface normal in cases where some parts of the data cannot be accessed because regular points have different normals in different coordinates and too much noise at scattered points.

To overcome this situation, the Poisson Surface Reconstruction method using Poisson equations has been developed. This method, called Poisson Surface Reconstruction, is similar to the triangular meshing method and has been frequently used in mesh generation studies from point cloud data. The most important feature that distinguishes this method from others is that it does not take the data piecemeal and combine it, but evaluates it as a whole and gives results according to the whole (Berger, Levine, Nonato, Taubin, & Siva, 2013). With this aspect, it is a method that can give more accurate results even for noisy data. With the frequent use of deep learning networks in computer science studies, the analysis of Poisson's equations using artificial intelligence models has accelerated.

One of these studies is a CNN-based volumetric shape completion algorithm developed by Dai et al. This algorithm is used to combine low resolution volumetric data and aims to obtain a whole volume instead of obtaining high resolution outputs (Dai, Ruizhongtai Qi, & Nießner, 2017).

Han et al. the input data taken as 3d is converted into a 3d mesh model using a two-unit network called global structure inference and local geometry refinement (Han, Li, Huang, Kalogerakis, & Yu, 2017).

Ohtake et al. A solution is proposed based on the local quadric error minimisation strategy, calculating the neighbourhoods of a single point and merging the set of adjacent points over time (Othake, Belyaev, & Seidal, 2005).

All the studies reviewed show that (Pietroni, Tarini, Sorkine, & Zorin, 2011; Xiong, Zhang, Zheng, Chai, & Liu, 2014; Ohtake, Belyaev,, & Seidel, 2005); In all deep learning-based or poisson-based solutions, it is necessary to establish the connection of other points clustered around a point in the best way. The proximity of these points to each other, noise removal methods in noise-containing data, and the equation according to which the data will be turned into a surface are the main problems to be solved in mesh model creation studies (Kazdan, 2005).

CHAPTER 3

MATERIALS & METHODS

3.1. SEGMENTATION & CLASSIFICATION

3.1.1. Deep Learning

The concept of Deep Learning was first introduced by Hinton in 2006, when it was suggested that multilayer artificial neural networks could be trained more efficiently (Yılmaz & Kaya, 2020).

Deep learning is a multi-layer neural network that is associated with artificial intelligence and machine learning methods and is often confused with these two fields of study. A deep learning model can be created for all areas where multi-layer artificial neural networks are used. These areas can be listed as Speech Recognition (SR), Natural Language Processing (NLP), Deep Neural Network (DNN), Image Segmentation and Recognition.

Deep Learning is different from machine learning methods that do rule-based learning. Unlike traditional methods, it can learn automatically through the features of a data and can be continuously improved. (Yılmaz & Kaya, 2020).

Deep learning networks can also learn from untreated image or txt data kept in a csv file. The accuracy value of the deep learning model, which is created according to the variety of data used and the number of data, also changes. The most important feature that distinguishes deep learning from rule-based systems is that the deep learning model is designed to learn from the data, instead of learning only the specified parameters with a set of rules. With this aspect, effective results can be obtained by using different parameters in solving similar problems.

Model selection plays an active role in problem solving in deep learning networks. The fact that the model is oriented towards the problem increases the performance to be achieved.

The learning process of machines is designed by imitating the human brain. Here, sine cells called neurons, which determine the brain's capacity to process information, are also modelled in artificial learning networks so that machines can learn like humans. This modelling is called artificial neuron (Yılmaz & Kaya, 2020).

Neuron networks, which form the basis of Deep Learning, are to determine the function approach of an unknown function.

Special functions are used in different learning layers to specify this approach. The learning network consists of 3 layers; input layer, output layer and hidden layers. While the input layer and output layer must be present in each network, the number of inputs and outputs may vary depending on the problem.

The hidden layer, is the layer in which activation functions are used, which can be increased or decreased according to the problem.

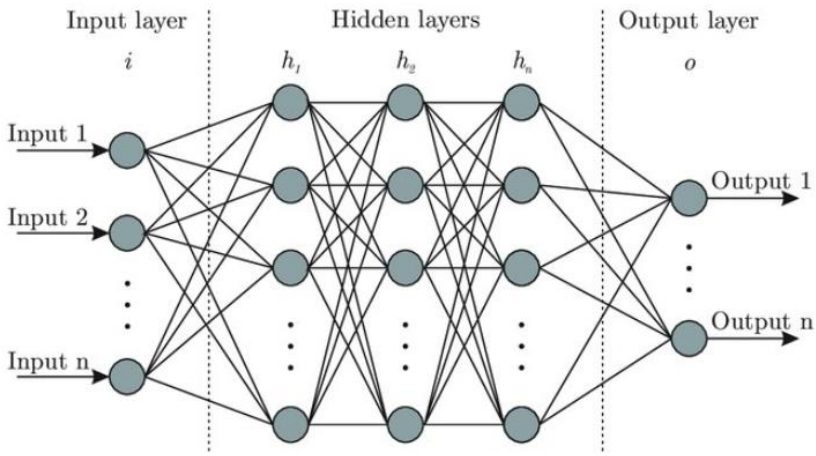


Figure 3.1. Network Model

Multilayer perceptrons, the feed-forward neural networks shown in Figure 3.1 are the basis of deep learning models.

3.1.1.1. Deep Learning Models

3.1.1.1.1. Restricted Boltzmann Machines (RBM)

It is a scotastic neural network that was designed by Hinton et al. in 1986 with a different version of the Boltzmann Machine and modelled using stochastic units with a certain distribution method such as the Gaussian distribution. Since these networks are probabilistic-based, they can be used to model the probabilistic relationship between variables.

Restricted Boltzmann Machines have a two-piece structure, visible and concealed, with symmetrical connections between them (Hinton & Salakhutdinov, 2006).

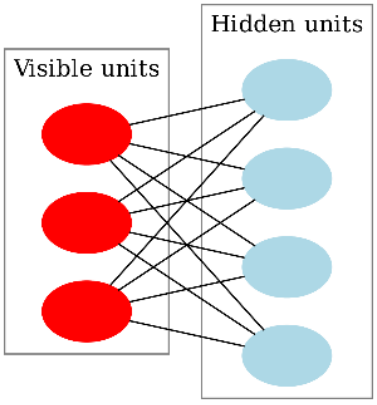


Figure 3.2. RBM Model

Each circle in the graph in Figure 3.2 represents a neuron-like unit called a node, and nodes are simply where computations take place. Nodes are connected to each other in layers, but no two nodes in the same layer are connected to each other. Therefore, there is no intra-layer communication.

This is the limitation of RBMs. Each node is a computational focus that processes input and makes random decisions about whether to forward that input.

3.1.1.1.2. Deep Belief Network (DBN)

A deep belief network (DBN) in machine learning is a generative graphical model. It is a class of deep neural network that consists of multiple layers of hidden nodes and there is a connection between the layers, and there is no connection between the nodes.

By training on an unsupervised set of examples, the DBN can learn to reconstruct its entries probabilistically. The layers then act as feature detectors. After this learning phase, a DBN can be trained with more controls to perform classification.

DBNs can be viewed as a composite of simple, unsupervised networks such as constrained Boltzmann machines (RBM) or autoencoders, with each subnet acting as the hidden layer, the visible layer of the next.

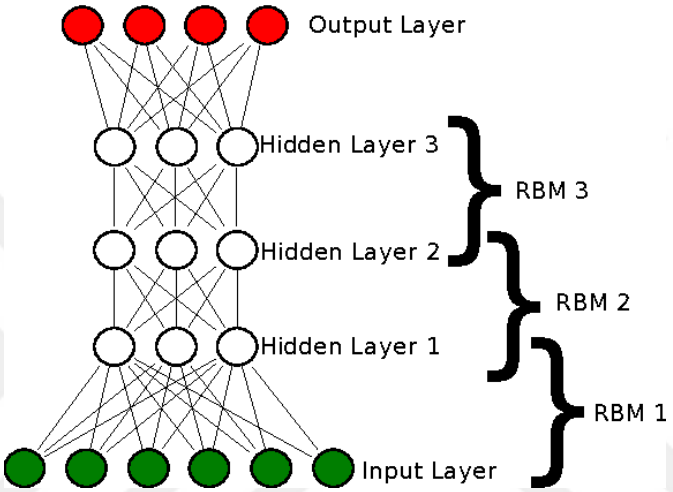


Figure 3.3. DBN Model

(semanticscholar.org, n.d.).

A DBN with k hidden layers has k weight matrices, w(1),….,w(k). However, the DBN has k+1 bias vectors, b (0)…b(k), with b(0) providing the bias values for the visible layer (Yılmaz & Kaya, 2020).

The probability distribution of the DBN is expressed as:

$$P(h^k, h^{k-1}) \propto exp\left(b^{k^t}\right)h(k) + b^{k-1^T}h^{k-1} + h^{k-1^T}w^k h^k \quad (3.1)$$

$$P(h_i^m = 1|h^{m+1}) = \sigma\left(b_i^m + w_{i,j}^{(m+1)T}h^{m+1}\right) \forall_i \forall_m \in 1, \ldots . k-2 \quad (3.2)$$

$$P\left(v_i = 1|h^{(1)}\right) = \sigma(b_i^0 + w_{i,j}^{(1)T}h^{(1)})\forall_i \quad (3.3)$$

With diagonal β value for traceability in case of real value units;

$$v \sim N(v, b^0) + w^{(1)T} h^{(1)}, \beta^{(-1)} \quad (3.4)$$

3.1.1.1.3. Denoising Autoencoders (DAE*)*

Auto-encoders (AE) called Diabolo network is a special artificial neural network used for unsupervised learning . AE is a neural network that copies values from the input layer to the output layer. That is, it recreates the data given as input to the neural network in the output layer. AE is a feedforward neural network that learns the best features from the compressed representation of the input data (Yılmaz & Kaya, 2020) As shown in Figure 3.4, DAE changes the weights of the input data until it still provides the same input as output after encryption-decryption. in this way, it tries providing the input and output by changing the weights until it reaches the target.

Deep autoencoders (DAEs), on the other hand, are neural networks consisting of multiple layers of AEs, in which the outputs of each layer are connected to the inputs of the successive layer.

It is a deep learning application in which back propagation learning is applied from unsupervised learning algorithms used in face recognition, signal noise removal, speech recognition.

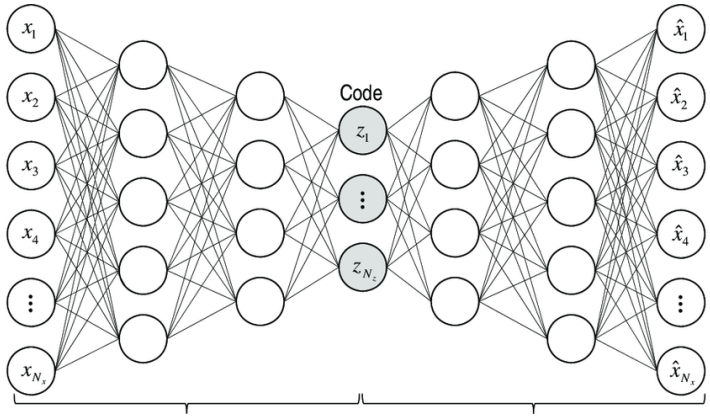


Figure 3.4. DAE Model

In Figure 3.4, the input sequence is obtained using the encrypted sequence in the first hidden layer. The input to the next hidden layer is obtained because of encryption in the 2nd hidden layer.

Because of these stages, the coding process has been completed. The continuation of this structure is the decryption phase. This structure, which is thought to be the opposite of the said operations, ends because of decrypting the password passing through the hidden layers and giving it as output (Yılmaz & Kaya, 2020).

3.1.1.1.4. Recurrent Neural Network (RNN)

Convolutional networks are types of neural networks that use a mathematical operation called convolution, based on processing data in a grid-like topology.The data can consist of time series data expressing a one-dimensional grid or picture images expressing a two-dimensional grid.In order for a convolution operation to occur, a convolution operation must be applied in at least a layer.The convolution layer contains the set of filters that can learn.Each filter occupies a small space in width and length and extends to the full depth of the inlet volume (Yılmaz & Kaya, 2020; Graves, 2013).

$$h_t = f(h_{t-1}, x_t) \tag{3.5}$$

$$h_t = tanh(w_{hh} h_{t-1} + w_{xh} x_t) \tag{3.6}$$

$$y_t = w_{hy} h_t \tag{3.7}$$

A recurrent neural network (RNN) is a network that is trained to process sequential data. This network is used for problems where information transfer between the previous output and the next input is important. It uses a definition of memory called memory between entries. (Yılmaz & Kaya, 2020).

There are many types of RNNs that update and maintain memory in different ways, such as Elman RNN, DTR (Discrete-time repeater), MLP (Multi Layer Perceptrons) and Long Short-term Memory (LSTM).

RNNs are a kind of Deep Learning constructs that are generally used to predict the next step. The biggest difference from other deep learning structures is that they remember. To create this memory structure, they use a loop-like structure within themselves in the hidden layers.

3.1.1.1.5. Region Based CNN (R-CNN)

Known CNN algorithms are used in many areas such as face recognition, image classification and object identification. These perfectly developed CNN algorithms have a major weakness in detecting only one object in an image. For this reason, it is not an effective method to use CNN alone in images that contain objects in more than one class. Especially, researchers working on image processing have developed new hybrid network models to solve this problem.

One of these network models, R-CNN, is a method that allows us to specify the classes of objects in an image and the bounding lines of the objects. R-CNN is used as an alternative solution when we cannot get results with CNN alone in images of multiple objects.

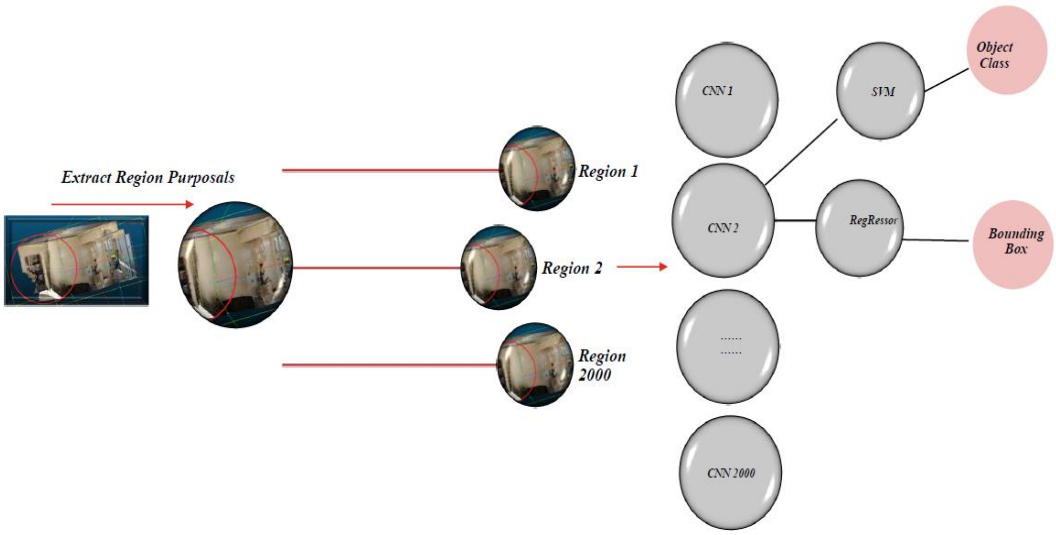


Figure 3.5. R-CNN Network

It consists of 2 basic steps, R-CNN Selective Search and Non-Max Suppression . Selective Search is a method used to find regions that need to be captured in images. Finding the smallest regions in a hierarchical fashion then relies on finding the main regions by combining small similar regions. In the region extraction process with selective search, 2000 different regions of an image are obtained.

The second step, Non-Max suppression, performs zone reduction. It is impossible to use all 2000 regions obtained in Step 1. It selects the ones to be used from these regions by specifying a certain IoU (Intersection Over Union) value. This threshold value is 0.5. A basic R-CNN Network shown in Figure 3.5.

3.1.1.1.6. Fast Region Based CNN (Fast R-CNN)

The 2000 different regions used in the Region Based CNN model and the use of 2000 CNN networks in the processing of these regions cause great costs in the creation and training of the model. To reduce this cost, the Fast R-CNN network, which uses a single CNN network instead of 2000, has been developed.

Fast R-CNN; has combined CNN, SVM and Regressor operations into a single operation to reduce time and learning costs.

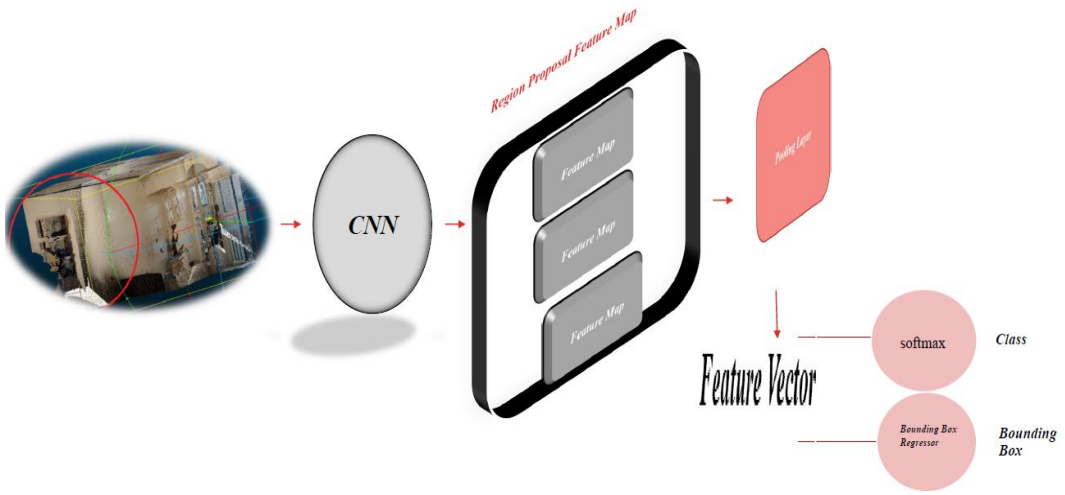


Figure 3.6. Fast R-CNN Model

As shown in Figure 3.6, in the Fast R-CNN model, the entire image is processed with a single CNN network and Feature maps are generated.

In this process, called Region Proposal Feature Map, the required attributes for each region proposal are collected.Obtained features are reduced to certain dimensions in the max-pooling layer and the reduced features are converted to one-dimensional vectors.The transformed vector is inserted into a neural network and the region classification and bounding boxes are obtained.

3.1.1.1.7. Faster R-CNN

Instead of Faster R-CNN Selective Search, developed by researchers who think that the Selective Search step used in Fast R-CNN does not reduce the cost of the model enough, it uses the more useful Region Proposal Network (RPN).

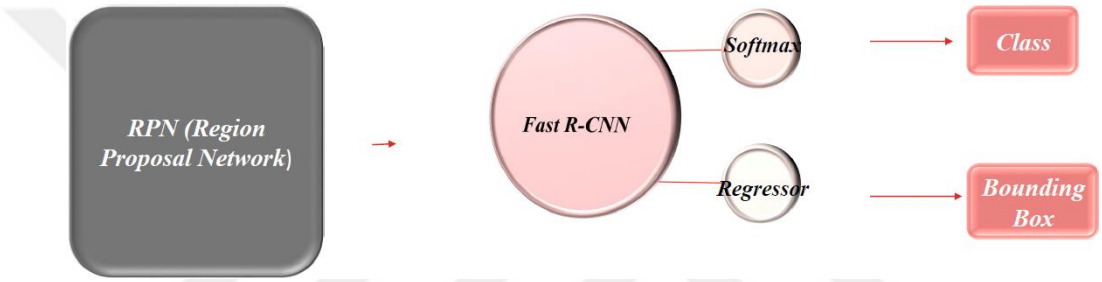


Figure 3.7. Faster R-CNN

In a system that combines Fast R-CNN as a package, the structure called RPN (Region Proposal Network) is used before Fast R-CNN is started. RPN is a region propositional convolutional neural network that has been frequently heard in computer science in recent years. Taking an array of any size as input, RPN outputs a rectangular proposal showing regions that may belong to an array of objects. In creating this proposal, it uses a small mesh wrapping method over the attribute map generated by the convolutional mesh.

3.1.2. Deep Learning Library

Some of the machine learning libraries used on the Python platform are customized for deep learning applications. TensorFlow(Python), Caffe(Python), Theano(Python), Keras(Python), Torch(C++), Deeplearning4j(Java), Covnetjs(Java), Mxnet(Python), PyLearn2(Python), Deep Learn Toolbox-Matlab, Accord.NET(C#), Sci-Kit Learn(Python).

MachineLearning, which has the widest working area among these libraries, Keras and TensorFlow libraries form the basis of deep learning applications with a python. In this study, the tensorflow library was used to create and train the deep learning network (Yılmaz & Kaya, 2020).

3.1.2.1. Keras

Keras can be defined as the core of deep learning applications. It has high level API which can be used in many deep learning (DL) applications. (Yılmaz & Kaya, 2020).

3.1.2.2. PyTorch

It is a python-based deep learning package that can perform Tensor calculations using libraries such as Numpy, supported by Python for mathematical operations and has a powerful graphics processing unit (GPU). This package allows the reuse of python packages such as Numpy, Scipy and Cython when needed. The deep neural networks in this structure are based on a band-based and autograd system (Yılmaz & Kaya, 2020).

3.1.3. Optimizers of Squential

Machine learning methods are used in the processes of creating an inference using existing information. The models developed in the inference creation process initiate the learning process by repetitively using the data. The feature that distinguishes deep learning models from other machine learning methods is to obtain the best generalization instead of finding a global minimum. Getting a good generalization means finding the closest result to the solution in a reasonable time (Seyyarer, Ayata, Uçkan, & Karcı, 2020).

Some optimization algorithms are used to minimize the error in the process of finding the closest result to the solution. Optimization methods are a step-by-step process. The number of steps operated in this process is called the learning coefficient. While determining the learning coefficient, the most appropriate value should be selected.

If the learning coefficient is selected small, the solution process is prolonged, if it is selected large, it causes the minimum point to be overlooked (Bosch, Zisserman, & Munoz, 2007; Frome, Singer, & Malik, 2006; Seyyarer & Aydın, 2017).

Stochastic Gradient Descent (SGD), Momentum, Adadelta and RMSprop derived from Adagrad algorithms are the most popular optimization methods (Cebeci , 2019; Fortuner, 2019). Figure 3.8 shows the evolutionary map of these optimization methods.

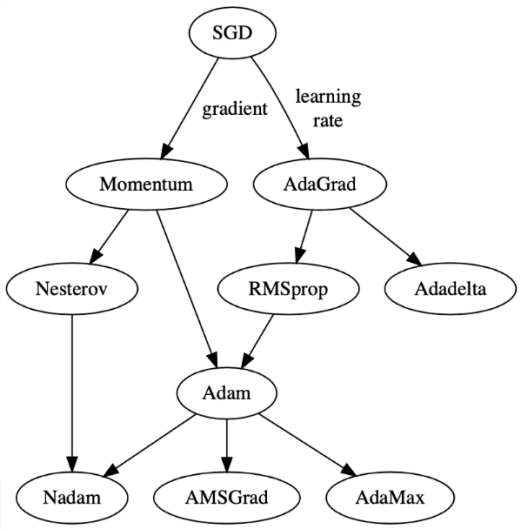


Figure 3.8. Evolutionary Map of Optimization Methods **(KDnuggets, 2022)**.

3.1.3.1. SGD

In many studies in the literature, it is used as a gradient descent. SGD doesn't work on all gradients. Instead of working on all gradients, it operates on random gradients. It updates the current weight (wt) by multiplying the current gradient (∂L / ∂wt) by the learning coefficient (a) (Bosch, Zisserman, & Munoz, 2007; Frome, Singer, & Malik, 2006; Seyyarer & Aydın, 2017).

$$w_{t+1} = w_t - \mathrm{a}\,\frac{\partial \mathrm{L}}{\partial \mathrm{wt}} \tag{3.8}$$

3.1.3.2. Momentum

While searching for the optimum point in SGD, there is a lot of oscillation. The momentum method aims to reduce oscillation that are seen as a barrier to reaching the goal. In this method, where the momentum gradient value is sought, the current gradient is not used. This is the aspect that distinguishes this method from the others.This method may also be called controlled stroke (Bosch, Zisserman, & Munoz, 2007; Frome, Singer, & Malik, 2006; Seyyarer & Aydın, 2017).

$$w_{t+1} = w_t - aV_t \quad (3.9)$$

$$V_t = \beta V_{t-1} + (1 - \beta)\frac{\partial L}{\partial wt} \quad ,Vt=0 \ , \quad 0< \beta < 1 \quad (3.10)$$

- For the β is commonly used value of 0.9.

3.1.3.3. AdaGrad

The problem known as constant learning coefficient is frequently seen in the SGD and Momentum methods. AdaGrad was designed and used to overcome this problem.

To operate with a different learning coefficient at each step, it divides the learning coefficient in Equation (3.12) by the square root of the cumulative sum of the squares of the past gradients (Bosch, Zisserman, & Munoz, 2007; Frome, Singer, & Malik, 2006; Seyyarer, Ayata, Uçkan, & Karcı, 2020).

$$W_{t+1} = w_t - \frac{a}{\sqrt{S_{t+e}}} \cdot \frac{\partial L}{\partial W_t} \quad (3.11)$$

$$S_t = S_{t-1} + \left[\frac{\partial L}{\partial W_t}\right]^2 \quad (3.12)$$

Here, S is initially 0, and ϵ is usually taken as an insignificant number (10-7) to avoid the error of dividing by zero.

3.1.3.4. RMSprop

It is proposed to solve the constant learning coefficient problem as in Adagrad. The difference between them is that instead of squaring the gradients in the adagrad method, the gradients with momentum are squared (Bosch, Zisserman, & Munoz, 2007; Frome, Singer, & Malik, 2006; Seyyarer, Ayata, Uçkan, & Karcı, 2020).

$$w_{t+1} = w_t - \frac{a}{S_{t+e}} \cdot \frac{\partial L}{\partial w_t} \quad (3.13)$$

$$S_t = \beta S_{t-1} + (1 - \beta)\left[\frac{\partial L}{\partial w_t}\right]^2 \quad (3.14)$$

3.1.3.5. AdaDelta

The Adadelta and RMSProp approaches were suggested by different people in the same year. Both methods are adagrad versions that focus on solving the fixed learning problem.

In the In Adagrad and RMSProp methods, there is an obligation to choose a learning coefficient. The Adadelta method does not require determining a learning coefficient. With this aspect, it provides flexibility in use and definition.

Instead of learning coefficient, momentum sums of squares of delta values expressing the difference between current weights and updated weights are used (Bosch, Zisserman, & Munoz, 2007; Frome, Singer, & Malik, 2006; Seyyarer, Ayata, Uçkan, & Karcı, 2020).

$$w_{t+1} = w_t - \frac{\sqrt{D_{t-1}}-1}{\sqrt{\mathrm{St}+\epsilon}} \cdot \frac{\partial \mathrm{L}}{\partial w_t} \tag{3.15}$$

$$D_t = \beta D_{t-1} + (1 - \beta)\,[\Delta w_t]^2 \tag{3.16}$$

$$S_t = \beta S_{t-1} + (1 - \beta)\,[\frac{\partial \mathrm{L}}{\partial \mathrm{w}_t}]^2 \tag{3.17}$$

$$\Delta w_t = w_t - w_{t-1} \tag{3.18}$$

3.1.4. Activation Functions

Functions and events in real life are not linear. The working logic of Neural Networks requires linearity. In order to apply artificial solutions to real life problems, some functions that will affect linearity are needed.

Basically, in a simple artificial neural network, x is defined as inputs, w as weights, and f(x) ie activation process is applied to the value transferred to the output of the network. Then, this will be the final output or the input of another layer.

Linear functions are only polynomials of odd degree.A neural network without activation function will behave like a linear regression with limited learning power.The neural network is expected to learn nonlinear situations as well.

Because neural networks are given complex unique information such as features obtained from images, video, text, and audio to learn. Multilayer deep neural networks can thus learn meaningful features from the data (Yılmaz & Kaya, 2020; Sharma, 2022).

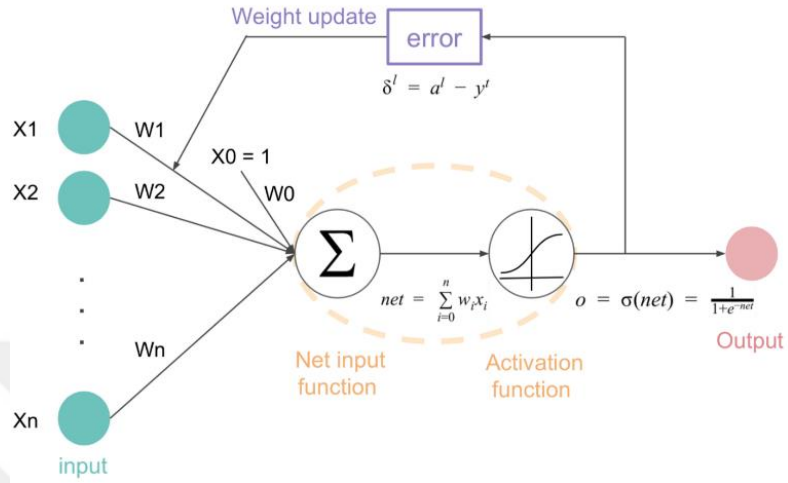


Figure 3.9. Usage of Activation Functions

(Sharma, 2022).

Artificial neural networks are designed as universal function convergent and are desired to work toward this goal. This means that they must be able to calculate and learn any function. Thanks to the non-linear activation functions, stronger learning of networks can be achieved.Usage of activation function shown in Figure 3.9.

3.1.4.1. Step

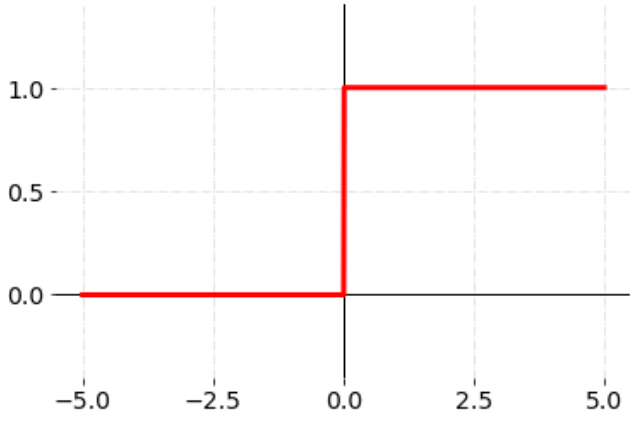


Figure 3.10. Step Activation Function

(Samson, 2022).

A function that takes two values, 0 and 1, is called a step function. It is a good classifier because it can only take 2 values and is usually used in the last layer of classification operations (Sharma, 2022; Samson, 2022).

The function defining the step function is shown in Equation 3.19 and the graph is shown in Figure 3.10.

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \tag{3.19}$$

3.1.4.2. Linear

Linear activation functions produce an ordered set of points. The values produced can be differentiated due to the equation f(x)=x, but its derivative is constant. In this respect, it is different from the step function, which produces values as 0 and 1 on the y axis (Yılmaz & Kaya, 2020; Shama, 2022). In back propagation algorithms, the learning process is done through neurons, and the back propagation algorithm is a derivative-based algorithm.

$$f(x) = x \tag{3.20}$$

The negative side of the linear activation function is that the learning process does not occur because the derivative is constant. linear functions increase or decrease with a certain ratio in the x and y axis. This situation can lead to rote learning while learning in hidden layers that are connected with each other, and therefore it is not preferred in model trainings.

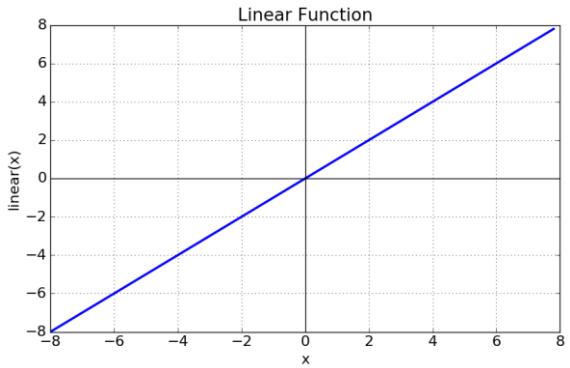


Figure 3.11. Linear Activation Function

(Shama, 2022).

3.1.4.3. Sigmoid

Many functions in nature are not linear.The sigmoid function is also inspired by functions in nature and consists of nonlinear combinations.It has a differentiable structure that forms the basis of learning continuity.This means that it consists of non-constant derivatives.When the graph in Figure 3.12 is examined, it is seen that even a small change in the x-axis will cause a big difference in the y-axis.

When Figure 3.12 is examined, it is seen that it is a function that can take an infinite number of positive and negative values. When sigmoid encounters all these values, it always generates values between (0,1). This shows that this function can be used as a good classifier. (Shama, 2022; Yılmaz & Kaya, 2020).

This situation prevents the activation values to be compatible with each other and to avoid extreme values.

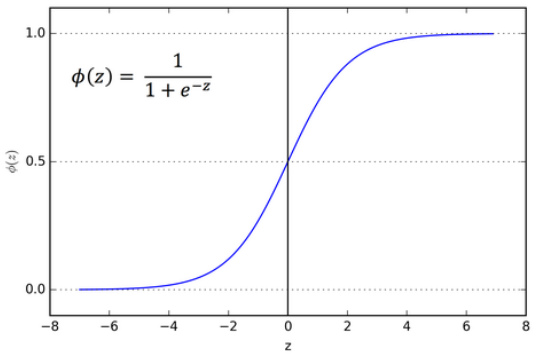


Figure 3.12. Sigmoid Activation Function

(Shama, 2022).

Examining the figure give in above , toward the ends of the function, the y-values respond little to changes in x. Here, the derivative values in these regions become insignificant and converge to 0.This is called vanishing gradients and learning occurs at a minimum level. When a slow learning event occurs, the optimization algorithm that minimizes the error can be stuck to local (local) minimum values and the maximum performance that can be obtained from the artificial neural network model cannot be obtained.

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^{x}}{e^{x}+1} = 1 - S(-x) \tag{3.21}$$

3.1.4.4. Hiperbolic Tanjant (Tanh)

It has a structure very similar to the sigmoid function. However, the range of the function is defined as (-1, +1). Its advantage over the sigmoid function is that its derivative is steeper, that is, it can take more values. This means that it will be more efficient because it has a wider range for faster learning and classification. In this function, the problem of the gradients dying at function ends continues.

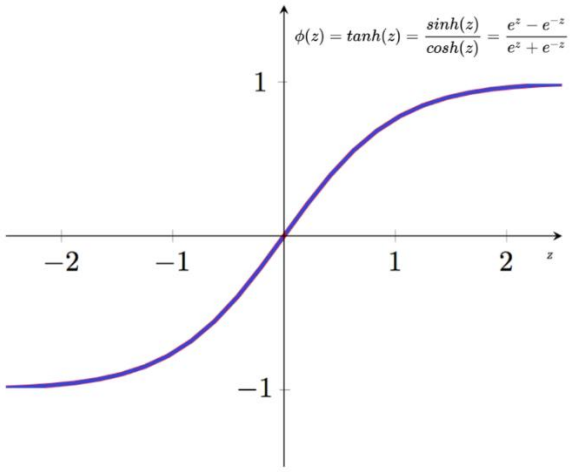


Figure 3.13. Tanh Activation Function

(Shama, 2022).

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.22}$$

3.1.4.5. ReLu (Rectified Linear Unit)

The most popular activation function recently is the relu function. It was used for the first time in 2012. It gives excellent results in imaginal dataset competitions every year .Although at first glance it seems to have the same properties as the linear function on the positive axis, ReLU is non-linear in nature.

$$f(x) = x^+ = \max(0, x) \tag{3.23}$$

if the x value is negative, the results is zero if the positive the results is x.

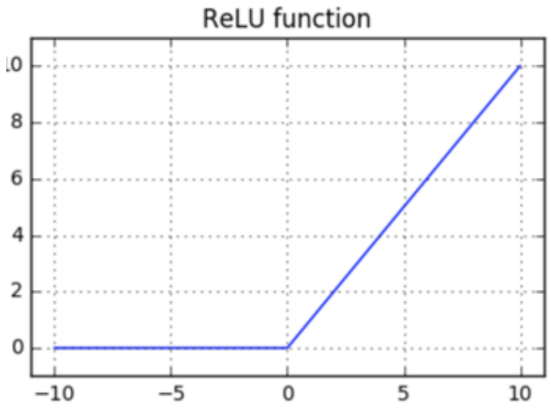


Figure 3.14. ReLu Activation Function

(Shama, 2022).

The ReLu function is a good estimator. It is possible to converge to any other function with combinations of ReLU.

Considering a large neural network with many neurons, Sigmoid and Hyperbolic Tangent cause almost all neurons to be activated in the same way.This means that the activation takes a lot of action.It is desirable that some neurons in the network are active and that the activation is an efficient computational load.This is achieved with the ReLU function.Getting 0 values on the negative axis also means that the network will run faster.The fact that the computational load is less than the sigmoid and hyperbolic tangent functions has caused it to be preferred more in multilayer networks.

3.1.5. Evaluation Methods of Classification Results

Regression and classification methods from supervised learning models use labelled training datasets to make predictions. Thus, new inputs fed from the output are created and learning can be supported. There are many different metrics to evaluate the results to be obtained.

3.1.5.1. Confusion Matrix

A confusion matrix is a table used to describe the performance of a classification model on a set of test data for which the actual values are known. The complexity matrix is an NxN matrix that shows the number of correct and incorrect predictions made by the classification model compared to the actual results on the data. (N: number of classes) Confusion matrix is the most used metric in the performance evaluation of models .

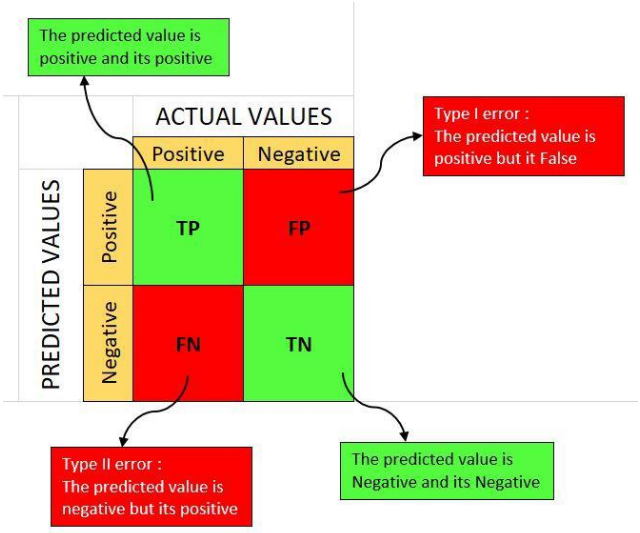


Figure 3.15. Confusion Matrix

(Suresh, 2022).

The definitions of TP, TN, FP and FN shown in Figure 3.15 are as follows (Öğündür, 2022):

True Positives (TP): These are instances where the true value is 1 and the predicted value is 1.

True Negatives (TN): These are instances where the true value is 0 and our predicted value is 0.

False Positives (FP): These are instances where the true value is 0 but our predicted value is 1.

False Negatives (FN): These are instances where the true value is 1 but our predicted value is 0.

By using the confusion matrix given in Figure 3.15, evaluation metrics such as accuracy, precision and recall shown in Figure 3.16 can be obtained.

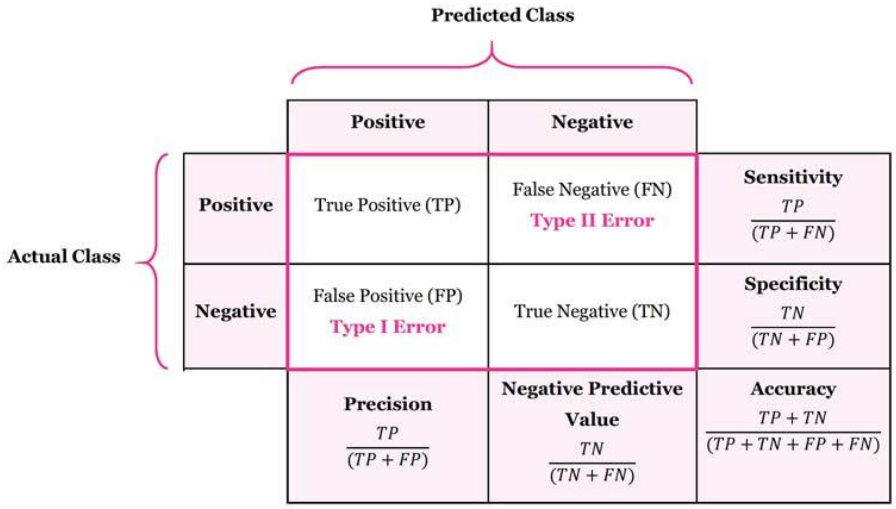


Figure 3.16. Evaluation Metrics of Artificial Network

(Lee R. , 2022).

3.1.5.1.1. Precision

It is an evaluation metric in which model performance is calculated based on positivepredictions. if it is costly to calculate the damages that a model may cause in wrong estimation, that is to calculate the FN and TN values, the precision value can be checked and the higher this value, the better the performance (Öğündür, 2022).

$$Precision = \frac{TP}{TP+FP} \tag{3.24}$$

3.1.5.1.2. Recall

Recall is a metric that shows how much of the transactions we should have predicted as Positive, we predicted as Positive. The precision value is a metric used when the cost of estimating as False Negative is high. It should be as high as possible (Öğündür, 2022).

$$Recall = \frac{TN}{FN+TN} \tag{3.25}$$

3.1.5.1.3. F1 Score

F1 Score value shows the harmonic average of Precision and Recall values. The reason why it is a harmonic mean instead of a simple mean is that extreme cases are not ignored.

If it were a simple average calculation, a model with a Precision value of 1 and a Recall value of 0 would have an F1 Score of 0.5 and an incorrect result would be obtained (Öğündür, 2022).

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3.25)$$

3.2. MESH MODEL GENERATION

3.2.1. CGAL (The Computational Geometry Algorithms Library) Studies

CGAL is an open source software library built in C++ that is used in various fields that need geometric computation such as Computer-aided design, molecular biology, medical imaging, Computer graphics and robotics (cgal.com, 2022). This library is frequently used in solid modelling studies from 3d point cloud data regardless of the platform. With the reconstruction method offered by this library as an open source, mesh model creation studies can be conducted using different value ranges. These methods are detailed in the following sections.

3.2.1.1. Surface Reconstruction

Surface reconstruction is a process of modification (change) of the surface layer of a crystal; causes the atomic structure of the layer to differs from the corresponding atomic planes in the crystal stack. The term 'reconstruction' is also used to describe the reconstructed surface itself (Williams, et al., 2019).

Surface reconstruction has been a large area of research since 1990. Berger et al. (Berger, Levine, Nonato, Taubin, & Siva, 2013) compared 10 different methods in a study and divided all of these methods into 4 main categories technically.

Indicator function techniques describe a scalar function that determines whether a given point is within a specified area (Kazdan, 2005; Kazdan, Bolitho, & Hoppe, 2006). More than one method can be used to define this function. The main feature that distinguishes these methods from each other is how they use the indicator function. Poisson Surface Reconstruction inverts the gradient operator by solving a Poisson equation to define the indicator function (Burrus, 1997).

While Fourier surface reconstruction represents the Fourier based on function display function, Wavelet surface reconstruction (Manson, Petrova, & Schaefer, 2008) uses Haar or Daubechies waves as the display function. The Haar wavelet (Burrus, 1997) is a discontinuous wavelet type that can be conceptually defined as fast, simple, and memory. Daubechies waves (Mallat, 1989) are also discontinuous waves that can be transformed quickly just like the Haar wavelet, and the only difference from Haar waves is that the filter length is more than two.

3.2.1.2. Poisson Surface Reconstruction

The Poisson distribution (Kazdan, Bolitho, & Hoppe, 2006) is a discrete-time probability distribution used to estimate the number of times an event will occur in a given time interval.

The Poisson distribution is expressed in Formula 3.26 using the probability mass function lambda parameter (Wikipedia, 2022).

$$f(x;\lambda) = p(X = x) = \lambda^x \epsilon^{-\lambda} \frac{1}{x!} \tag{3.27}$$

- "k" is number of events (k=0.1.2…)
- "e" stands for Euler number (e=2.71828...).

Given a set of 3D points, the indicator function that best matches the normals of the input data is calculated using Poisson Surface Reconstruction.Then, the region covering the function in the point set whose boundaries are determined, is filled with cubes and solid model creation is performed (CGAL, 2022). The steps to be followed to obtain a 3D model from the point cloud data using the Poisson method are shown in Figure 3.17.

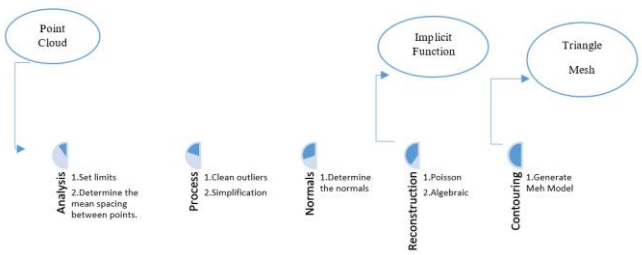


Figure 3.17. 3D Model Generation Steps

CHAPTER 4

THE PROPOSED HBIM METHOD

4.1. SEGMENTATION & CLASSIFICATION

4.1.1. Point Cloud Segmentation Methods

4.1.1.1. HBIM Dataset

In this study, the Riegl VZ400 Laser scanning (LIDAR) device was used to create the HBIM dataset.

The data obtained from the device were processed in the RiscanPro program that came with this device. This machining process involves assembling structure data scanned piecemeal in the field. Because of Merging, a 3D point cloud is obtained.

The HBIM dataset consists of historical buildings in Gaziantep province, which was taken as a 3D point cloud with a laser scanner. These data were obtained from the relevant institutions and organizations working on these structures in Gaziantep.

Due to insufficient scanned data on the buildings in Gaziantep, each room in the building was included in the system separately, instead of using the existing buildings as a whole. In this way, 140 rooms were obtained from 19 historical buildings.

The number of data for the deep learning network to be created with this method has been increased.

However, due to insufficient data over time,11 synthetic structures (converted from 3D BIM model to point cloud) were created and included in the system.

For each room and building, 5 labels were determined as door, window, wall, floor, and ceiling, which are defined as unique building elements.

The labeling process of 140 rooms and 11 synthetic data used in the study is shown in Figure 4.1.

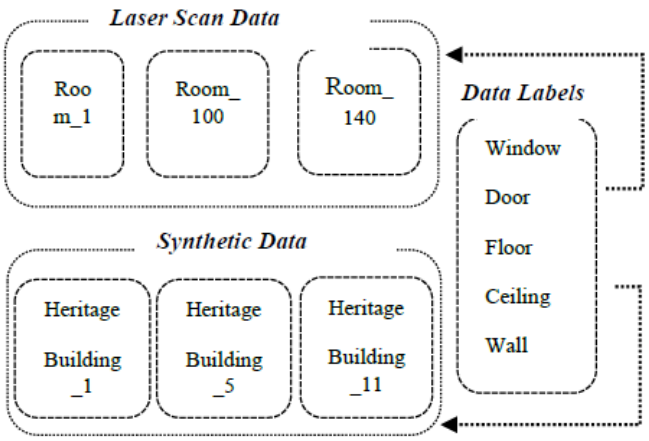


Figure 4.1. HBIM Data Structure

During labelling, the unique architectural features of these historical buildings were taken into account. In the labelling process, point set data are labelled with Bentley Pointools, and point cloud processing software.

One of the main reasons for using this software is that it is a free and easy-to-use software system that allows the segmentation of the obtained point cloud. Additionally, Bentley Pointools v8.1 version was preferred because the extensions (. pod, rsp) of the scan data (LIDAR) used in the study can be opened and processed in this software.

First, by giving coordinates to the corners of the labelled building elements, a model was tried to be produced as in Figure 4.1. However, it has been determined that in some building elements, the model cannot be created by only giving coordinates to the corner points. As a result, the second method has been developed to segmentation-building elements.

The second method is the process of location-based separation of the structural element to be segmented from the entire structure that has been laser-scanned. This process is performed by leaving the individual building elements in isolation from the whole building data and saving the isolated element as a separate file without changing or distorting its location and coordinates. The recorded building elements were recorded by naming them according to the room and type. In this way, a more accurate model was obtained by giving coordinates to each point on the defined elements. Point cloud data were transferred to the Bentley Pointools programme and classified.

4.1.2. Synthetic Data Generation Methods

Two different scenarios were tried within the scope of generating synthetic data using restitution data. The application steps of these scenarios are detailed in Figure 4.2.

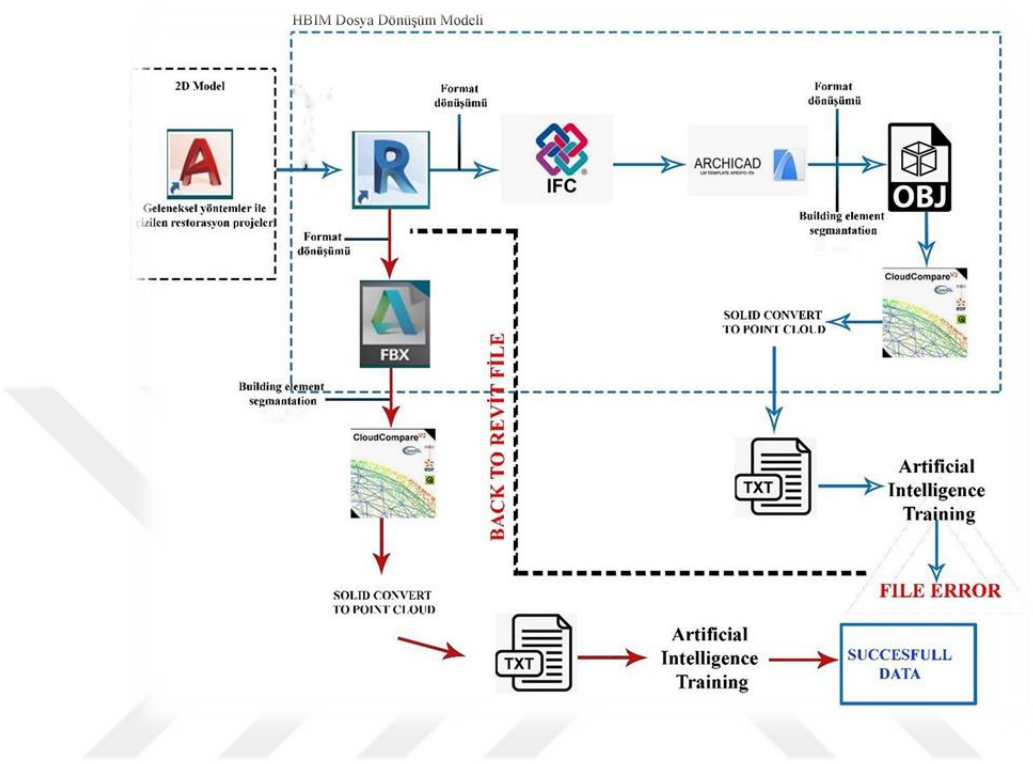


Figure 4.2. Synthetic Data Generation Methods

4.1.2.1. IFC Format Conversion

The fact that the HBIM dataset, which is created from laser scanning data, consists of historical buildings that have been deformed over time, reduces the training and testing performance of the deep learning network. In order to increase the network performance, 3D point cloud data were produced using 2D restitution data using reverse engineering methods.

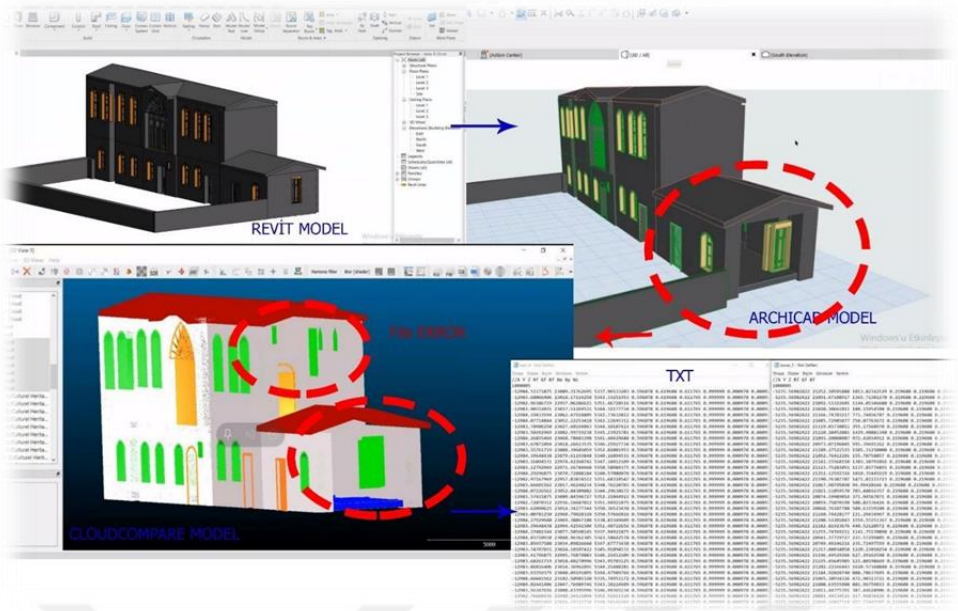


Figure 4.3. IFC Format Conversion.

The following steps were followed while performing the reverse engineering studies shown in Figure 4.3:

- In order for the data converted from 2D environment to 3D environment with Autodes
- k Revit, a BIM software, consists of parametric objects and this information is not lost, files were transferred between programs with IFC format.
- In order to transfer data to CloudCompare using this method, IFC files were converted to OBJ format.
- In this process, ArchiCAD program was used for the conversion of the data converted to IFC format and the data was converted to .obj format.
- Finally, the data in .obj format was manually labeled and converted to point cloud data.

4.1.2.2. FBX Format Conversion

By converting from the IFC format, incompatibilities arising from the coordinate systems between the programs were observed during the data production phase. As an alternative way to eliminate these coordinate inconsistencies, studies have been carried out to transfer HBIM models developed in the Revit environment directly to the CloudCompare platform in FBX format.

4.1.3. HBIM (Heritage Building Information Modelling) Architecture

In the PointNet architecture given in Figure 4.4, the input layer consists of a set of MLPs that use the properties of point clouds. In the layer known as the Max Pooling layer, the symmetric properties of the input data are used, the input permutation calculations are made, and the global values of the data are calculated. Fully Connected Layers, known as the last layer, perform label prediction and classification.

In the PointNet network (Qi, Su, Mo, & Guibas, 2017), 3D data consisting of n points is taken as input. To transform the input data, Input Transform and feature transform operations are performed, which enable the independent transformation of each point. The schema showing this transformation is given in Figure 4.5. In the most general terms, PointNet takes a series of *(x, y, and z)* coordinate values, and each point in this coordinate array is in the form of labelled data. It is an integrated system that can classify and segment by calculations on coordinate values and determination of the surface normal values. Three basic modules make up this integrated system (Qi, Su, Mo, & Guibas, 2017). *Symmetry Function for Unordered Input*, it is described as order a set of irregular data in an orderly and understandable order, train the ordered data using the RNN network, and generate a new set of vectors using a symmetric function.

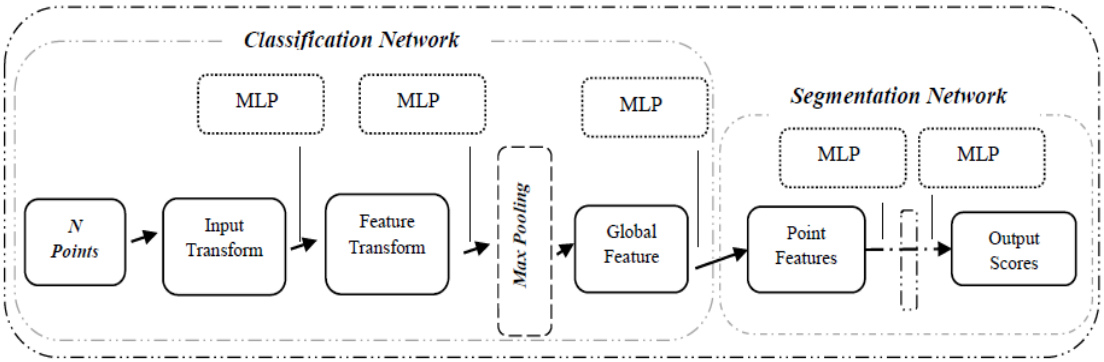


Figure 4.4. PointNet Architecture

(Qi, Su, Mo, & Guibas, 2017).

PointNet processes the n input data in an artificial neural network known as MLP (Multi-Layer Perceptrons) to obtain regular data. After the input is transformed (64,64), it is passed through the MLP network again for the feature transformation (64,128,1024) and the input data is converted into regular information in nx1024 dimension.

It has been proven in the literature that high performance is achieved with the use of RNN networks on 3D point cloud data. To create a suitable RNN network in the PointNet network, our input data must be based on a universal function. This function is shown in Equation (4.1).

$$f(\{x_1..x_n\}) \approx g\big(h(x_1),..,h(x_n)\big) \tag{4.1}$$

Where, $f: 2^{R^N} \to R$, $h: R^N \to R^k$, $g: R^k x..xR^k \to R$

An input dataset consisting of [f_1,..,f_k] can be used for training using an SVM (Support Vector Machine) or another classifier. However, a combination of local and global information must be used to perform point cloud segmentation (Qi, Su, Mo, & Guibas, 2017; Qi, Yi, Su, & Guibas, 2017).

PointNet has defined the module where it performs this operation as *Local and Global Information Aggregation.* Point features are extracted from the point inputs and a new operation is defined by using the global properties of each point in the network given in Figure 4.4.

In this way, combined properties consisting of new local and global information are defined for each point. Although the number of data does not change during segmentation, the input data containing more information are included in the model.

Therefore, our chance of more accurate segmentation will be increased. The module called *Joint Alignment Network (JAN)* is included in the PointNet architecture so that the labels of the segmented point clouds are not lost after 3D grid or solid model transformations, as well as to protect the segmentation (Qi, Su, Mo, & Guibas, 2017). In this module, a transformation matrix is defined in a mini-network called T-Net for data transformation. This matrix is shown in Figure 4.5.

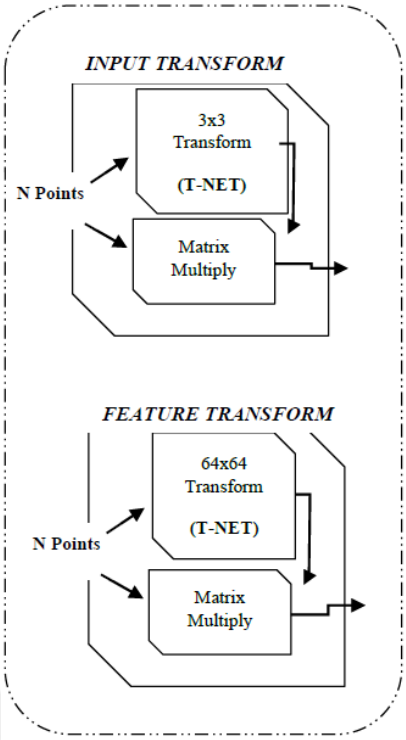


Figure 4.5. Data Transformation

(Qi, Su, Mo, & Guibas, 2017).

The size increases in the feature matrix due to this matrix transformation caused model optimization to take much more time. Therefore, the problem was solved using the Softmax training function in the model. The feature transformation matrix is limited by the formula given in Equation (4.2) . In this way, a more stable and efficient network is obtained.

$$L_{reg} = \|I - AA^T\|2F, \tag{4.2}$$

where A is the feature alignment matrix predicted by a mini network. Studies using PointNet architecture have inspired many studies in different fields. In this study, the segmentation study of historical-cultural structures in Gaziantep was made with our original data by making improvements on the PointNet network.

4.1.4. Research Methodology: Case Study

The research employs the case study methodology. Heritage buildings in Turkey at risk in Gaziantep are selected as case studies. Images of the case studies are shown in Figure 4.6.

These buildings are provided by the Heritage Conservation Department of the Gaziantep Metropolitan Municipality, called KUDEB (KUDEB, 2022) that is an active partner in the project as the end user. Thus, experts from KUDEB also validate the research outcomes and the related test results.

These mansions from 16th century are the listed historic buildings in Gaziantep, reflecting local character and identity of Gaziantep, and their restoration has been recently completed by the Gaziantep municipality and relevant documentations about their historic background, restitution records and restoration experience and challenges are all recorded and available in KUDEB.

Figure 4.6. Historic Buildings in Gaziantep (a) Kozanlı Konak in Gaziantep, (b) Eyüpoğlu Konak in Gaziantep

These case study buildings are currently used as museums and selected from many other examples provided by KUDEB, based on the scale, accessibility, available rich semantic information with moderately complex nature to be able to manage and manipulate within the research capacity. KUDEB provides the registered and post-processed point cloud data for these case study buildings.

Point cloud data of the heritage buildings captured via terrestrial 3D laser scanner will be used since it is more appropriate than airborne LIDAR in capturing the characteristic details of heritage buildings. These point cloud data will form the datasets, which will be provided by KUDEB for the research and development. In this study, the segmentation study of historical-cultural structures in Gaziantep was made with our original data by making improvements on the PointNet network (Qi, Su, Mo, & Guibas, 2017). Figure 4.7 shows the Deep Learning (DL) based research process flow.

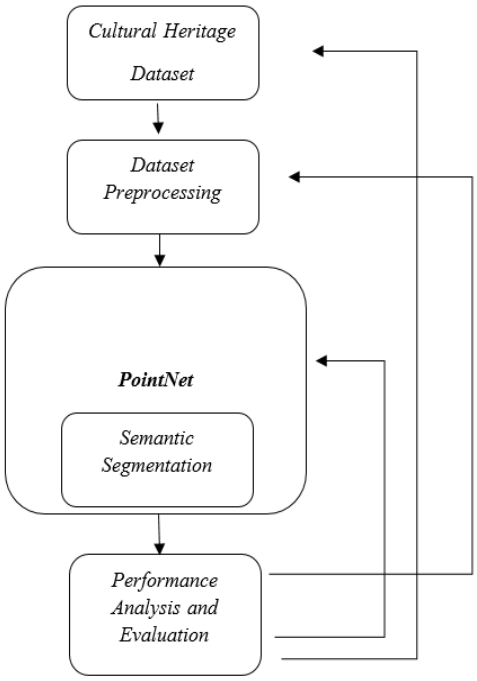


Figure 4.7. PointNet Based Heritage Data Segmentation Research Process Flow

The main challenge is the accurate classification and segmentation of the point cloud model.The main strategy for this is the surface-based segmentation because the intention is to categorize the mesh model as surfaces of walls, windows, etc.

Once this segmentation is achieved, the segmented entities will be dealt with separately in the 3D environment for 3D BIM extraction. Mathematical modelling for BIM extraction will use object recognition and constructive solid geometry methods. As a result, an accurate 3D BIM model of the heritage building will be generated. This BIM model will form the 3D geometric dataset for the HBIM system, which will be developed via an incremental prototyping approach.

4.1.5. Classification and Segmentation

It is important that the data to be used in the training and testing of our deep learning network is obtained from lidar or radar scanning data, not 2D pixel arrays (images) or 3D voxel arrays.

The *Point Interactions* operation states that if we want to obtain meaningful data from each point, the points should be evaluated together with their neighbourhoods (Qi, Su, Mo, & Gubias, 2017; Qi, Yi, Su, & Guibas, 2017).

In the step called *Transformation invariance* shown in Figure 4.8, MLP was used to increase the (x, y, z) coordinates of each point from 3 dimensions to 64 dimensions and then from 64 dimensions to 1024 dimensions. These processes, detailed as Input transforms and Feature transforms in the previous sections, constitute the first stage of the classification process.

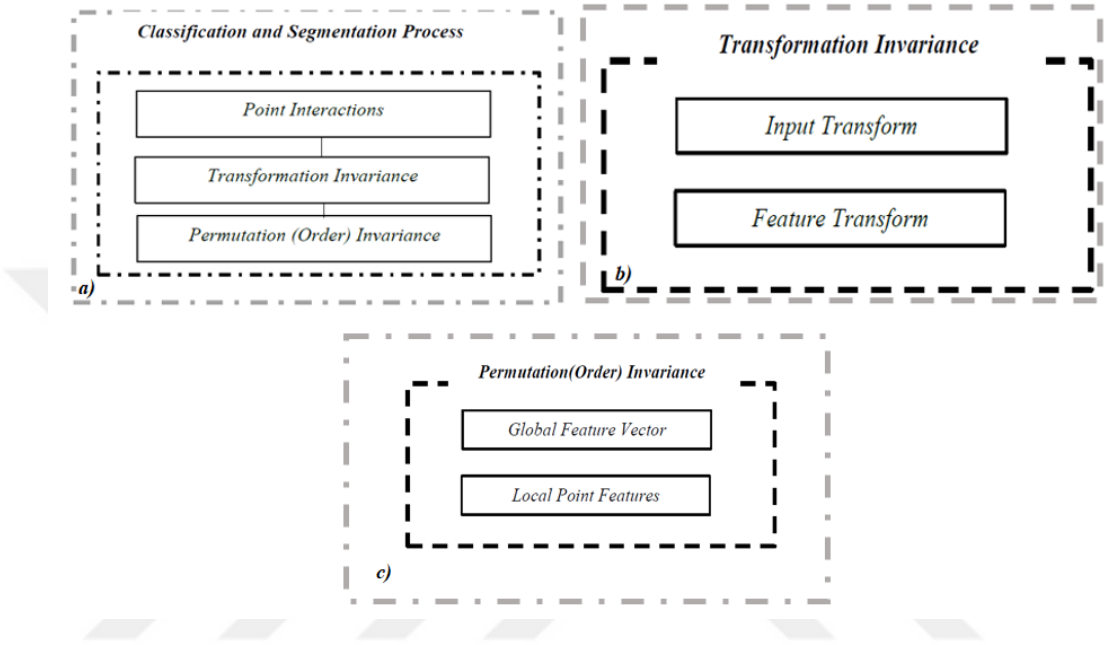


Figure 4.8. Research Process Plan (a) Classification and Segmentation, (b) Transform Invariance, (c) Permutation (Order) Invariance

(Qi, Su, Mo, & Guibas, 2017).

To be able to segmentation after classification, we will use deep learning architectures that directly consume point clouds and well respect the permutation invariance of points in the input capable of reasoning about 3D geometric data such as point clouds or meshes. In the step called Permutation invariance given in Figure 10, an MLP network was used to obtain global features and Local Point Features.

For an array containing N points, N! situation arises. N! cases must mean the same thing for a single point. Therefore, all probabilities must be based and fixed on a single function.

Global and local features are obtained as the output of the MLP network after fixation using a symmetric function. While global feature vectors are used in the classification, segmentation can be performed when used with local point features.

The vector defined as R^{1088} for each point in the MLP network used in the segmentation process is converted into an array of nxm dimensions. Here, n is the number of points and m is the number of classes (Qi, Su, Mo, & Guibas, 2017).

4.1.5.1. Point Cloud Segmentation

Point cloud data can be cluttered and occluded. The main challenge is the accurate classification and segmentation of this data (i.e. walls, windows, floor, doors, etc). To achieve this goal, the main strategy is the surface-based segmentation, indeed the intention is to categorise the point cloud mesh model as surfaces of walls, windows, etc.

In the literature, image processing techniques incorporating voxelization (Gonzalez & Woods, 1993), Region growing, Brute force plane sweeps, Hough Transforms (Dube & Zell, 2011) , Expectation Maximization techniques (Zhu, Meng, Cai, & Lu, 2016) are tested and implemented for the surface-based segmentation but due to large amounts of data and extracting information from enormous datasets, these techniques were not sufficient for point cloud segmentation. Thus, deep learning will be the main approach in point cloud segmentation.

A point cloud dataset collected with 3D laser scanners was created. The objects of the dataset were labelled as doors, windows, walls, etc. This process is labour-intensive and manual. The dataset was divided into 3 groups for training, validation, and testing. This separation was done at 70%, 10% and 20%, respectively. Weights were created by training the training and validation dataset with the PointNet model. The test dataset with 20% rate was used to measure the test success of the trained model.

4.1.6. The Proposed Deep Learning Model

4.1.6.1. Convolutional Neural Network (CNN)

CNN; It is also known as a Customized Transformational Neural Network that processes data, known as a grid topology.CNN is the most popular applied to computer vision and the most common sharing of parameters that occur within the network.The CNN model treats the picture as a 2D grid of pixels while working on the picture .

It generalizes the connection of these pixels with each other by using relational parameters, and in this way, it can perform identification and classification operations on an image.

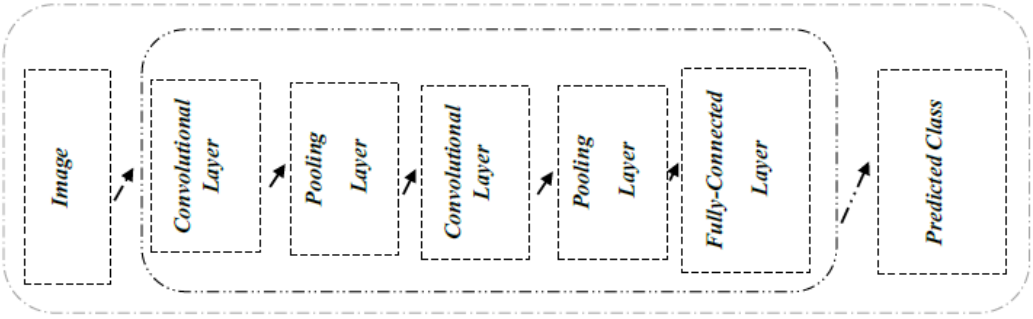


Figure 4.9. CNN Network

The CNN algorithm, primarly based on a standard neural network, is used for the solution of classification problems. It uses the unique properties of each class when performing the classification process, also it works with a multi-layered structure to determine and identify distinctive properties (Yılmaz & Kaya, 2020; Wikipedia, 2020). The definition of the layers that make up this classifier are given below.

Convolutional Layer: Used to extract features from an image. Applies some filters to extract high and low-level features in an image.

Non-Linearity Layer: It is the stage of introducing non-linearity to the system. It is the layer where activation functions are used to ensure that the outputs do not have a linear combination.

Pooling (Downsampling) Layer: It is used to reduce the processing load on the network. Thanks to this layer, smaller outputs are used for the neural network to make the right decision.

Flattening Layer: Prepares the data for the Classical Neural Network. Converts the data in the matrix form from the pooling layer into a one-dimensional array.

***Fully-Connected Layer*:** It is the Standard Neural Network used in the Classification. It performs the learning process using the Neural Network.

4.1.7. Deep Learning Library

4.1.7.1. TensorFlow

Tensorflow, which was opened as an open source software library by Google in 2015, is a deep learning supported artificial intelligence library. By using Tensor Flow libraries, a method that accelerate and facilitate the learning process of artificial intelligence systems, artificial intelligence technologies that perform different functions are better trained and brought to a more useful position (Yılmaz & Kaya, 2020; Tensorflow.org, 2022).

Although the library, which allows centralized processing using more than one CPU or GPU, was widely used on Python when it first emerged, it is also used in object-oriented programming languages such as C#, C++, Java, JavaScript today (Yılmaz & Kaya, 2020; Tensorflow.org, 2022).

It is an advantageous library with its feature of being a high-level neural networks API (Application Programming Interface), which can work on Keras, TensorFlow, Theano and Microsoft Cognitive Toolkit (CNTK) .

TensorFlow is a technology with various uses. One of the most important factors in its widespread use is that there are TensorFlow libraries prepared for different platforms. In other words, TensorFlow provides a library suitable for the platform for projects that can be developed on mobile applications, web applications or IoT devices (Yılmaz & Kaya, 2020).

4.1.8. Optimizers of Squential

4.1.8.1. Adam

It is the gradient descent algorithm proposed by combining the advantageous aspects of rmsprop and momentum methods. It uses the V in the momentum method and the S in rmsprop (Bosch, Zisserman, & Munoz, 2007; Frome, Singer, & Malik, 2006; Seyyarer, Ayata, Uçkan, & Karcı, 2020).

$$w_{t+1} = w_t - \frac{\mathrm{a}}{\sqrt{S_{t-e}}} \cdot \hat{\mathrm{V}}_t \tag{4.3}$$

$$\hat{\mathrm{V}}_t = \frac{V_t}{1-\beta_1^t} \tag{4.4}$$

$$\hat{S}_t = \frac{S_t}{1-\beta_2^t} \tag{4.5}$$

$$V_t = \beta_1 + V_{t-1} + (1 - \beta_1) \frac{\partial L}{\partial w_t} \tag{4.6}$$

$$S_t = \beta_2 S_{t-1} + (1 - \beta_2) \left[\frac{\partial L}{\partial w_t}\right]^2 \tag{4.7}$$

4.1.9. Activation Function

4.1.9.1. SoftMax

It has a structure very similar to the sigmoid function. It performs very well when used as a classifier, just like Sigmoid.

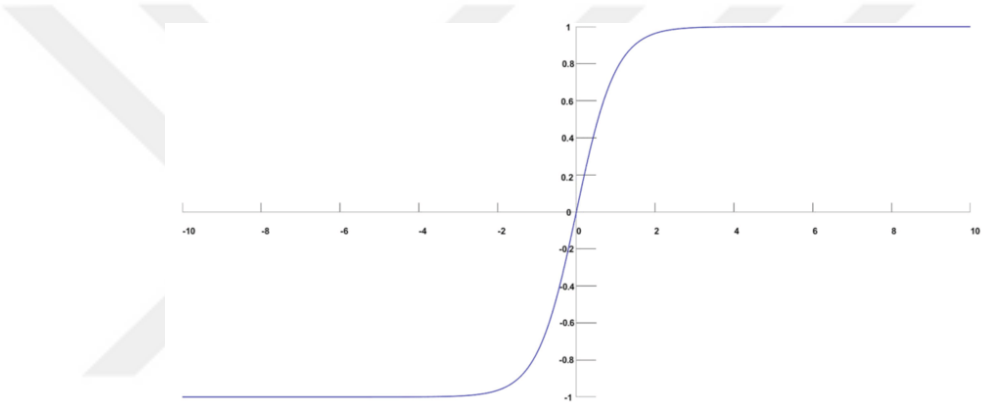


Figure 4.10: SoftMax Activation Function

(Leixian, 2022).

The most important difference is that it is preferred in the output layer of deep learning models, especially in cases where it is necessary to classify more than two, such as the sigmoid function. It determines the probability of the input belonging to a certain class by generating values in the range of 0-1. That is, it performs a probabilistic interpretation (Wikipedia, 2022; Yılmaz & Kaya, 2020).

$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$ for i=1,,,,,K and $z = (z_1,,,,,z_k) \in R^K$. (4.8)

(Wikipedia, 2022)

4.1.10. Hyper Parameters of Deep Learning Models

Before deep learning studies, methods called feature extraction was used to learn from a data set, and the best representative of the data was selected from the obtained features.

With the development of deep learning methods, these methods have been replaced by studies on how to design a multilayer artificial neural network, how many layers it will consist of, how many neurons it will contain, which optimization algorithm or activation function will be used.

Many hyper-parameters, such as the number of layers to be used in the design of a CNN network, the number of neurons in the hidden layers, the selection of the activation function to be used between the layers, the dataset width of the created network, the number of repetitions of the datasets are determined by the designer.

In a deep learning network, there are different sets of generally accepted hyperparameters where the best results for solving the problem are obtained. These groups can be changed to obtain better results according to each problem, and a new best hyperparameter group can be obtained for the solution of the problem studied. The choice of hyperparameters is generally based on the intuition of the designer, the experience gained from previous problems, the reflection of applications in different fields on our own problem, current trends, design dependence within the model, etc. varies accordingly (Çarkacı, 2022; Yılmaz & Kaya, 2020).

4.1.10.1. Size of Dataset

Preparation and selection of data sets in artificial intelligence models is an important field of study. In deep learning applications, the size of the data set is an important concept as well. Because deep learning models require working with large datasets. In learning models that work with large data sets, the system requirements for training are high. Large data sets require high performance for processing.

However, as the dataset increases, the performance does not increase proportionally to infinity, after a certain point the performance starts to increase little by little; The representativeness of the dataset also has a limit point. In this sense, while training the model, it should be investigated at what point the dataset we have is broken.

In other words, if we can achieve the same performance using a smaller data set and training and storage space is important to us, the dataset can be waived to a certain extent (Çarkacı, 2022; Yılmaz & Kaya, 2020).

The breaking point of the dataset used in the deep learning method is shown in the figure below.

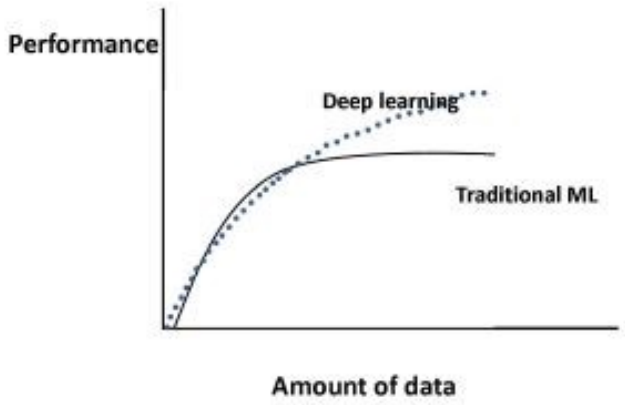


Figure 4.11. Comparasion of the Performance and Data Size **(towardsdatascience, 2022)**.

Deep Learning applications should work with a large data set. In cases where there is not enough data, it will be more accurate to use machine learning methods. If there is not enough data in a study that requires deep learning methods, alternative methods should be used. The methods we used in this study can be summarized as follows.

- The data set should be increased with synthetic data or feature transfer should be done with "transfer learning" methods.
- If the passivity between the classes in the data set is high; that is, if the classes are too similar to each other, or if there is too much noisy data, the performance graph can show a lot of bumps. Here, if the data cannot be changed, the problem can be solved by changing the "batch" value.

4.1.10.2. Batch Size

Processing the entire dataset used to train a model at the same time is a time- and memory-consuming process. As the number of data in the dataset increases, processing all the data at once may adversely affect the accuracy of the model. As a solution to this problem; The dataset is divided into small groups and the learning process is done on these selected small groups.Processing more than one input in parts in this way is called “mini-batch”.

The value determined as the mini-batch parameter while designing the model; means how much data the model will process at the same time. In this studies carried out, it has been determined that the loss value increases but time is gained in the case of processing the data in groups (mini-batch) (Çarkacı, 2022; Yılmaz & Kaya, 2020).

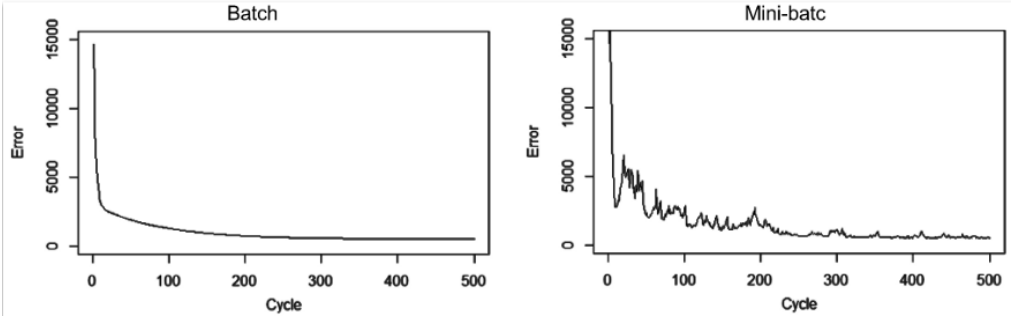


Figure 4.12. Performance Comparasion of Batch and Mini-batch **(Çarkacı, 2022)**.

As can be seen in the Figure above, fluctuations may occur in the division of the dataset into parts. The reason for this is that the determined hyperparameters are suitable for one group of data and not suitable for another group of data.

In case the mini-batch value is equal to the number of all elements in the training set; Since all the data in the training set will enter the training, the process will be the same as "batch gradient descent".

So it will look like the left graph in Figure 4.12. In such a case; The model will learn less noise. However, learning will take a very long time as it will process all the data at the same time. Fluctuations were observed in the dataset used in this study.

With the effect of synthetic data added to the dataset, the use of the mini-batch method is inappropriate. For this reason, the batch parameter will be used.

4.1.10.3. Learning Rate

Updating the parameters in deep learning is done by backpropagation. In the backpropagation process, this update is done by finding the difference by using backward derivative called "chain rule" and multiplying the difference value with the "learning rate" parameter, subtracting the result from the weight values and calculating the new weight value (Çarkacı, 2022; Yılmaz & Kaya, 2020).

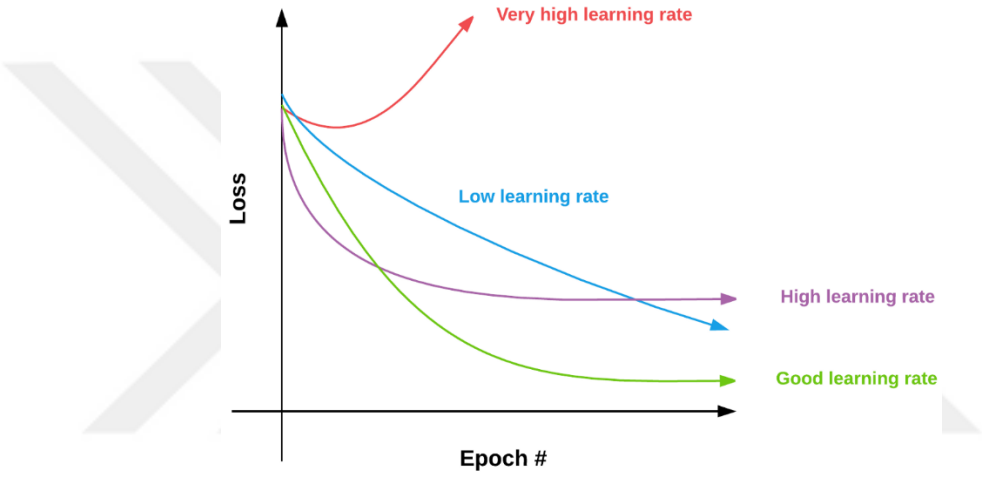


Figure 4.13. Selection of Learning Rate

(Kaur, 2022).

The "learning rate" parameter used during this process can be determined as a fixed value, or it can be determined as a step-by-step value, it can be determined depending on the momentum value, or it can be learned during learning by adaptive algorithms.

When Figure 4.13 is examined, the following can be said about the learning rate;

- The higher the learning rate, the more the network is affected by the data.
- A high learning rate will cause oscillation. However, being small will cause the learning to take a very long time as it will progress in small steps.
- If the learning rate is too small, it may get stuck on the local optimum value and cause the global optimum value not to be reached at all.

- The learning rate value is usually 0.01 as the default value, and it is reduced to 0.001 after a certain epoch.
- The generally used momentum beta coefficient is 0.9. The appropriate parameter range is 0.8 to 0.99.

4.1.10.4. Number of Epoch

While training a model, the entire dataset is not included in the training at the same time. It is divided into a certain number of pieces and each piece is included in the network sequentially. The weights are updated retrospectively according to the results obtained from each dataset included in the training .This process is repeated at each training step to try calculating the most appropriate weight values for the model.Each of these training steps is called an "epoch".

Since the most suitable weight values to solve the problem in deep learning are calculated step by step, the performance will be low in the first epochs, and the performance will increase as the number of epochs increases. However, there will also be a breaking point in learning. The size of the epoch number also varies according to the problem type. For example, the number of epochs should be kept larger in pattern-learned RNNs (Recurrent Neural Networks) compared to other models. As the number of epochs increases, the performance of the model increases significantly. Training can be terminated at these points, as performance will increase in insignificant units after a certain epoch (Çarkacı, 2022; Yılmaz & Kaya, 2020).

4.1.10.5. Dropout Value

Dilution of the nodes below the specified threshold in Fully connected layers increases the performance. That is, learning to forget weak information is directly related to the performance of the network. 0.5 is generally used as the dropout value. Situations where it is used differently are also common. It varies according to the problem and dataset. Also, when the Dropout value is used as the threshold value, it is defined as a value in the range of [0,1]. It is not mandatory to use the same dropout value across all tiers; Different dilution (Dropout) values can also be used (Çarkacı, 2022; Yılmaz & Kaya, 2020).

4.1.10.6. Weight Value (w)

The weight values (w) of the model can be defined as 0 when starting the model, as a distribution with a standard deviation of 0.5, with a uniform distribution between 0.5 and 0.9, or by using the weight values of an earlier model. There are other weight determination methods. How the weights are determined the learning and speed of the model.

If the W weights are initialized to 0 at the beginning of the model, the input values appear as the output, as the matrix product is a sum.

$$y = f(x, w) \rightarrow w.x \tag{4.9}$$

Since the values of the w matrix are 0, the addition of x and the matrix containing zero will still give x itself. Therefore, w weights should not be initialized to 0 initially. If the W weights are started from randomly small numbers at the beginning of the model, the model will work in small networks, but will result in a heterogeneous distribution of activation between network layers (Yılmaz & Kaya, 2020; Çarkacı, 2022).

4.1.10.7. Number of Layers and Neurons

The most important feature that distinguishes the deep learning method from other artificial neural networks, especially in complex problems, is the number of layers. However, there is no general theory about how many layers should be. The reason for this is that increasing the number of layers according to the complexity of the problem sometimes gives the opposite result.

For this reason, when starting a unique work, the number of layers should be kept low and the number of layers should be increased in a controlled manner according to the rate of affecting the problem solution. If a similar study has been done before, these studies can be taken into account in determining the number of layers (Yılmaz & Kaya, 2020).

In this study, a structure suitable for the PointNet (Qi, Su, Mo, & Guibas, 2017) architecture has been put forward and the number of layers is compatible with this architecture.

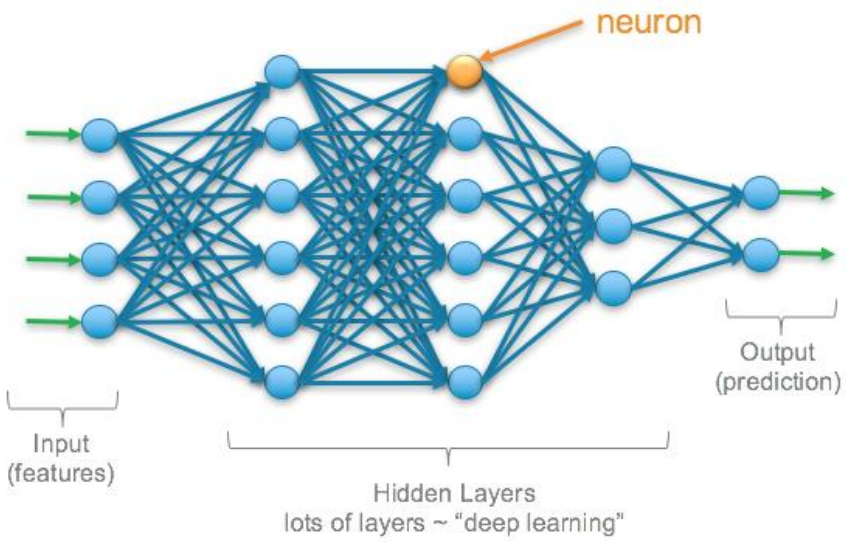


Figure 4.14. Neurons and Layers

(Ronaghan, 2022).

The neurons seen in the figure above can be thought of as a processing unit. Each of these processing units holds different information and processes the information to be sent to the next neuron. The higher the number of neurons in a network, the higher the amount of information to be processed. In this case, the system requirement to train the model is greater. However, information loss may occur due to insufficient use of neurons in the network. Therefore, the number of neurons to be used in the network, including the hidden layers, and their equal distribution to each layer are important.

4.1.10.8. Convolutional Neural Network (CNN) "Pooling" method: max, min, mean

In convolutional neural networks, kernels that operate on the matrix are used in each layer. These are like simple Gabor filters. The size of these kernels is highly effective for learning. Because the kernel size and how wide the data will affect each other are decided. Generally, 3x3, 5x5, 7x7 kernels are used. Using a large kernel will cause the image to be smaller after convolution is applied. Here, small sized kernels such as 3x3 is generally used because it causes a loss of information. In the edge detection process, to be able to look to the right, left, top and bottom of the pixel being processed, filters that can be centrally composed of odd numbers are used (Çarkacı, 2022).

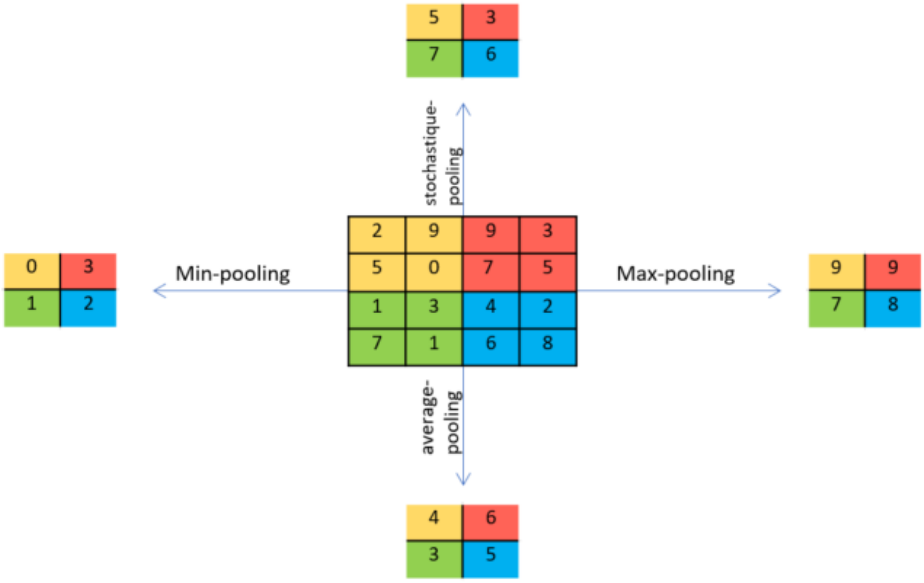


Figure 4.15. Pooling Methods

The "pooling" operation is performed on the output of the kernels used in convolutional neural networks. The "pooling" process is a filtering technique for the data in the kernel. Frequently used pooling methods; Taking the largest one in the matrix in the kernel is “max pooling”, taking the smallest one is “min pooling” and taking the average is “mean pooling”. Among these, the most preferred and claimed to give the most successful results is “max pooling”. In this study, the max pooling method was used.

4.1.11. Evaluation Methods of Classification Results

4.1.11.1. Accuracy (%)

It is an evaluation criterion that indicates how many percent accuracy the labels predict during the training and testing phases of the deep learning model. While calculating accuracy in the training dataset, the dataset called validation dataset is used to test the network as if it were the test dataset. In this study, 80% training data set and 20% validation set were used to test the training model.

Accuracy is a statistical evaluation metric. It shows how close a parameter is to reality. In other words, accuracy is the ratio of true positive and true negative data to the whole dataset. (Metz, 1978). The accuracy formula is shown in Equation 4.10.

$$Accuracy = \frac{TP+TN}{Tp+TN+FP+FN} \quad (4.10)$$

4.1.11.2. IoU (Intersection Over Union)

The AI-based model trained using point cloud data was used in the segmentation of other point cloud data. The intersection over union (IoU) value calculation method, known as the jaccard index (Jaccard, 1908), was used to measure and verify the success of the segmentation process. The IoU value is a frequently used verification and measurement method of Object detection (Xu, Ma, He, & Zhu, 2019), object segmentation (Hou, et al., 2021) workspaces.

This value measures the similarity between ground truth and model prediction. The IoU calculation method is the intersection of the ground truth and the predicted area divided by the combination of these two areas, as shown in Equation (4.11).

$$Score(IoU) = \frac{Area\_of\_Overlap}{Area\_of\_Union} \quad (4.11)$$

4.2. MESH MODEL GENERATION

4.2.1. Materials

Building Information Modelling (BIM) studies, which support automation systems, have gained momentum due to the limitation of manual methods and labour-intensive work in the restoration process of historical buildings (Volk, Stengel, & Schultmann, 2014) . A point cloud of a detailed 3D model of a region or a building can be created using 3D laser scanners. 3D point clouds consist of all points whose positions are determined in 3D space. Points with defined XYZ coordinates in space combine to form an image of an object and/or a building (Tang, Huber, Akıncı, & Lipman, 2010). The image consisting of dots gives information about the object figuratively. According to the resolution value of the laser scanner that creates the point cloud, there are gaps between these points. The structure consisting of spaces and points is not a solid model. To convert 3D points to solid models, points that are close to each other must be joined by completing the gaps between them and a surface must be created between these points.

In this study, the method of creating a 3D mesh model of cultural-historical heritages scanned and segmented with 3D laser scanners was examined. Point cloud data, which is formed as a result of modelling cultural-historical buildings with 3D laser scanners, can be used for such components as doors, windows, roofing, etc. The detection and segmentation of point cloud objects by AI is a scientific field of study (Qi, Su, Mo, & Guibas, 2017).

PointNet (Qi, Su, Mo, & Guibas, 2017), an architecture based on segmentation and classification, is a CNN-based segmentation algorithm in which point cloud data is used on 13 different labels. In this deep learning network, the input layer consists of a set of MLPs that use the properties of point clouds. In the layer known as the Max Pooling layer, the symmetric properties of the input data are used, the input permutation calculations are made, and the global values of the data are calculated. Fully Connected Layers, known as the last layer, perform label prediction and classification.

Deep learning-based algorithms are used to segment point clouds of objects within point clouds (Qi, Su, Mo, & Guibas, 2017). The purpose of segmentation is to distinguish objects from each other by identifying objects in an entire point cloud. Object identification is used for object recognition (Zhou & Tuzel, 2018) in robotic systems, autonomous driving (Zhou & Tuzel, 2018) in autonomous vehicles, and restoration (Andriasyan, Moyano, Nieto-Julián, & Antón, 2020) of historical buildings for modelling purposes. The method that this study focuses on is to improve the restoration and restitution process by remodelling the 3D scan point cloud results of segmented historical buildings. The methods available in the literature for converting 3D point cloud data to 3D model files are examined in the process.

4.2.2. Research Methodology: Case Study

The study consists of a four-step process as shown in Figure 4.16. These stages are grouped under two main parts. The first part, shown with the dotted line, shows the segmentation process of point cloud data, which is the current study area in the literature (Qi, Su, Mo, & Guibas, 2017). The part indicated by the dashed line is the subject of this study.

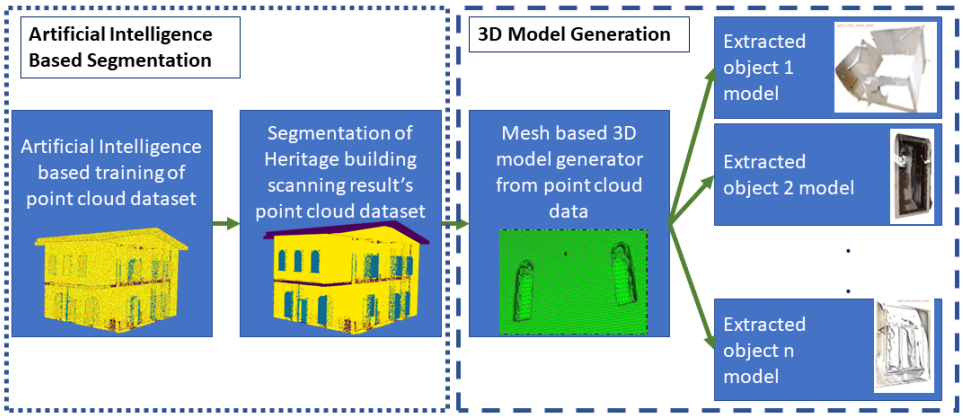


Figure 4.16. Stages of extraction of 3D mesh models of building objects from point cloud data

The first part is the creation of the point cloud of cultural-historical buildings scanned using 3D laser scanners and the AI-based segmentation operations on the data. In the second part, after the segmentation process is completed, the point cloud data is turned into a solid mesh model and the building's sub-elements such as door, window, roof, etc. are separated. The point cloud data, which is created by setting the XYZ coordinates of the defined points of the identified elements, can be converted into 3D solid models by completing the gaps between the points. In this study, a 3D mesh model was created from the point cloud using the Poisson Surface Reconstruction (PSR) method (Kazdan, 2005; Kazdan, Bolitho, & Hoppe, 2006).

The Poisson distribution (Kazdan, Bolitho, & Hoppe, 2006)is a discrete-time probability distribution used to estimate the number of times an event will occur in each time interval. The distribution formula Equation (4.12) is also shown.

$$f\left(k;\lambda\right)=\Pr\left(X=k\right)=\frac{\lambda^{k}e^{-\lambda}}{k!} \tag{4.12}$$

- k is the number of occurrences
- e is Euler's number *(e = 2.71828...)*
- $!$ is the factorial function.
- λ is the expected value of X .

The AI-based model trained using point cloud data was used in the segmentation of other point cloud data.

CHAPTER 5

RESULTS & DISCUSSION

5.1. EXPERIMENTAL DATASETS

5.1.1. Laser Scanner Data

The HBIM dataset consists of historical buildings in Gaziantep province, which was taken as a 3D point cloud with a laser scanner. This data was obtained from the relevant institutions and organizations working on these structures in Gaziantep.

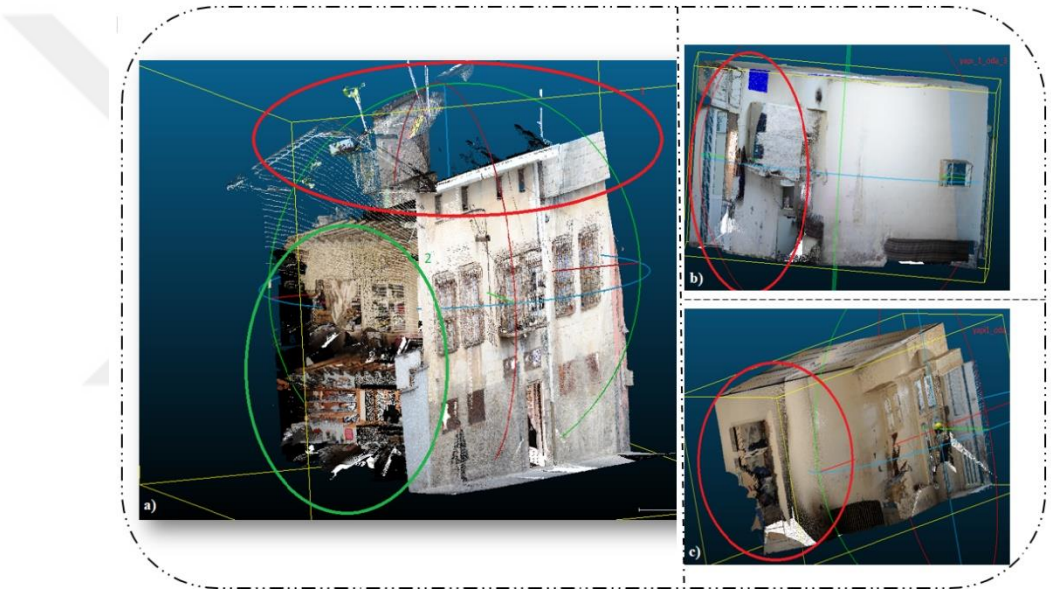


Figure 5.1. Deformed Laser Scanner Data (a) 'Structure_1_room_1', (b) 'Structure_1_room_2', (c) 'Structure_1_room_3'

Gaziantep historical buildings contain damage and deformations. There are coordinate losses caused by the damages of various building elements in the data sets numbered in the range 1-18, which we define as survey-based building data. In Figure 5.1, the deformations in the survey data set of structure 1 are shown. The damage and deformations on the whole of the building 2 whose rooms are shown in Figure 5.2.

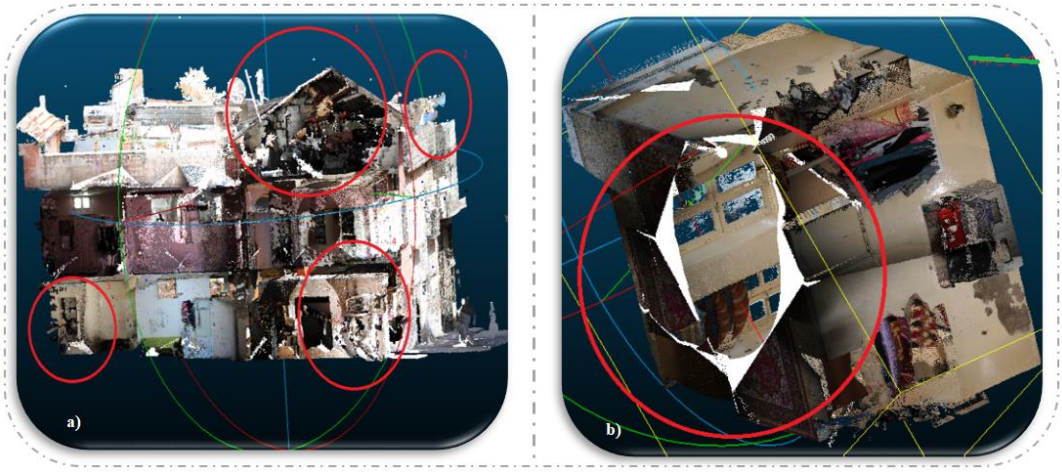


Figure 5.2. HBIM Laser Scanner Data (a) 'Structure_2_room_1', (b) 'Structure_2_room_2'

Due to the insufficient scanned data on the buildings in Gaziantep, each room in the building was included in the system separately, instead of using the existing buildings as a whole. In this way, 140 rooms were obtained from 19 historical buildings. In addition, 11 synthetic structures (converted from 3D BIM model to point cloud) were created and included in the system.

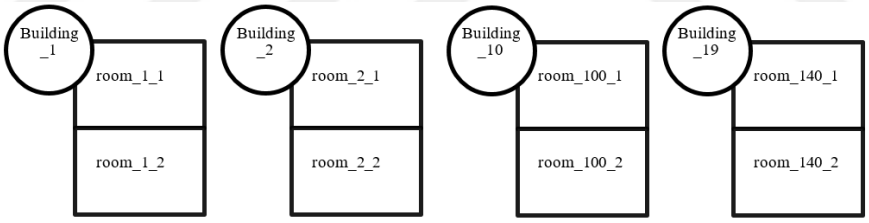


Figure 5.3. HBIM Dataset Collection

Five different labels were made for each room: door, window, wall, floor and ceiling, which are the unique elements of these structures. The labeling process of 140 rooms used within the scope of the study is shown in Figure 5.3 and 5.4 .

Room 1	*Room 2*	*Room 3 ...*	*Room 138*	*Room 139*	*Room 140*
Door	Door	Door	Door	Door	Door
Ceiling	Ceiling	Ceiling	Ceiling	Ceiling	Ceiling
Floor	Floor	Floor	Floor	Floor	Floor
Window	Window	Window	Window	Window	Window
Wall	Wall	Wall	Wall	Wall	Wall

Figure 5.4. HBIM dataset labelling

5.1.2. Synthetic Data

Obtaining point cloud data from HBIM restitution data was included in the synthetic data generation section. Restitution data converted from the FBX format is given in the Table 5.1. These transformations were made by the architects involved in the HBIM creation process.

Table 5.1. Synthetic Data Collection

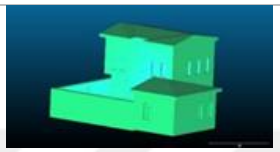
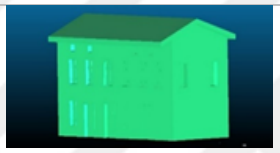
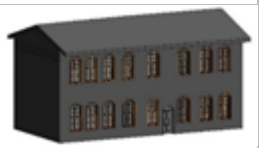
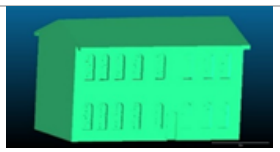
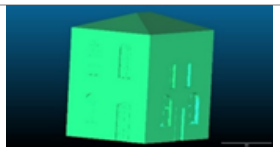
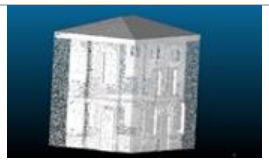
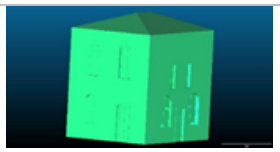
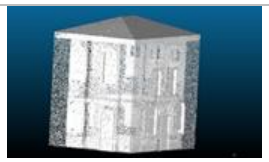
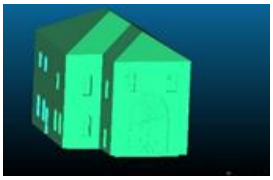
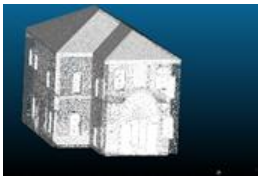
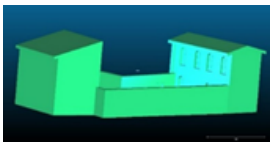
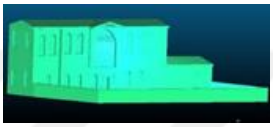
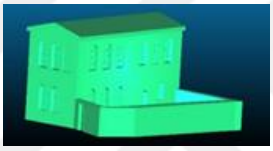
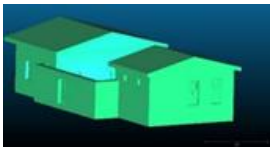
B-N	*REVIT MODEL (.RVT)*	*SOLID MODEL (.FBX)*	*POINT CLOUD (.TXT)*
19			
20			
21			
22			
23			
24			

Table 5.2. Synthetic Data Collection (continued)

B-NU	*REVIT MODEL(.RVT)*	*SOLID MODEL (.FBX)*	*POINT CLOUD (.TXT)*
25			
26			
27			
28			
29			
30			

5.1.3. Preparing the Dataset for the Training Network

HBIM data was created by separating the labels (door, window, wall, floor, ceiling) according to x, y, z coordinates in 'txt' format. In order to include the obtained labeled HBIM data in the system in accordance with the PointNet infrastructure, data in 'ply' and 'h5' formats were created using the Python programming language. With this process, the classification process was carried out with the tags expressed as “0, 1, 2, 3, 4” from the data processed according to the tag name. The outputs of these files are shown in the following sections.

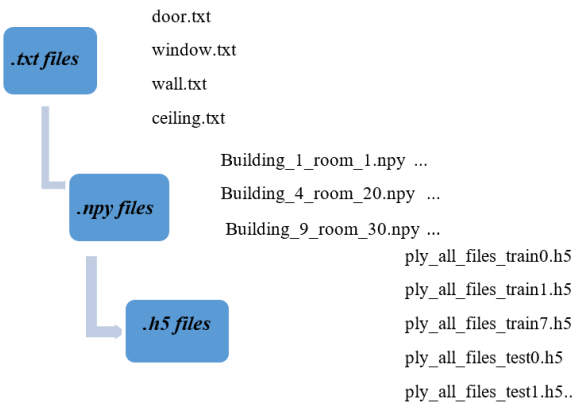


Figure 5.5. Prepocess of HBIM dataset

With the completion of the pre-processing stages shown in Figure 5.5 , the data converted into a matrix is included in the deep learning network. With this transformation, it is easier to make inferences from the data as a whole, and a great deal of time is saved in the learning process.

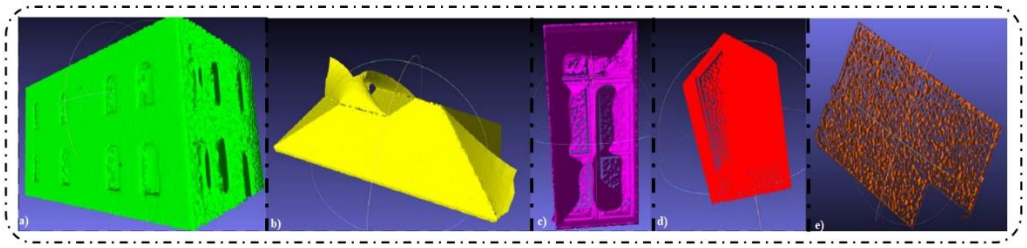


Figure 5.6. Point Clouds of Synthetic Data labels

As shown in Figure 5.6 (Separate the rooms), dividing each structure into 5 different labels is a common operation in both laser scanning and synthetic data.

Point clouds obtained by dividing the point cloud data of one of the synthetic data shown in Table 5.1 into 5 different labels are given in Figure 5.6.

5.2. EXPERIMENTAL RESULTS

Artificial intelligence training is planned to be performed in two stages. In the first stage, labelling for data preparation will be made, and in the second stage, training and testing will be carried out using labelled data. At this stage, a significant part of the data set will be used for artificial intelligence training.

In the literature, 70% to 80% of the datasets were used for training and the remaining 20% to 30% were used for testing. Considering these rates, test and training sets will be used at the same intervals in this study.

The data allocated for training was used to train the deep learning model using PointNet (Qi, Su, Mo, & Guibas, 2017), and the data allocated for testing was used to measure and verify the success of the trained deep learning model.

The test data given to the trained model is obtained as a segmented point set model as the output. The diagram showing this process is given in Figure 5.7.

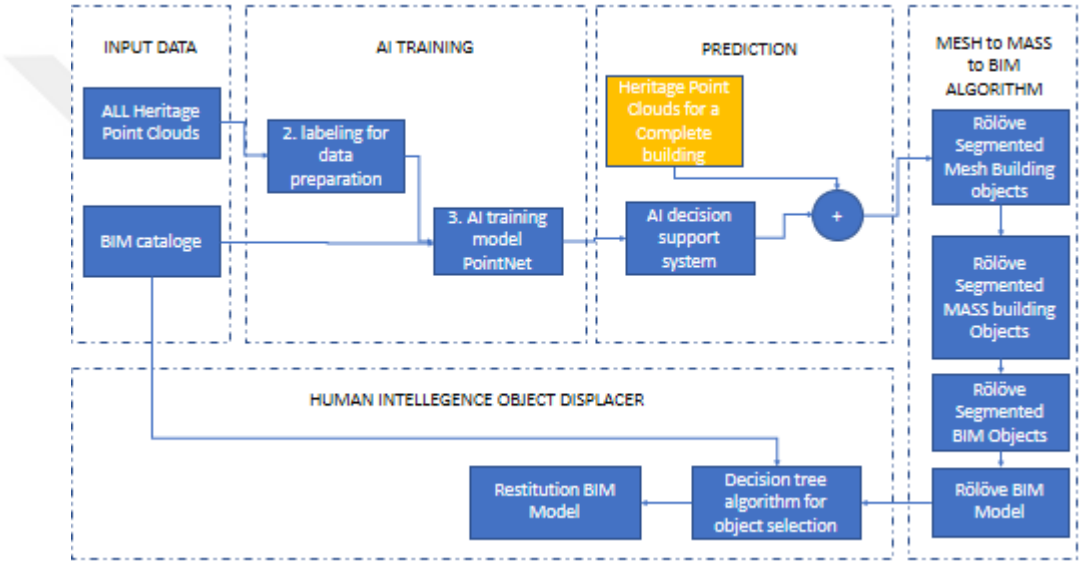


Figure 5.7. HBIM Process Diagram

According to the given diagram, first, point cloud data were prepared for the training of the network and a dataset was created, then point cloud segmentation was done using the prepared data. In the last step, mesh model generation studies were conducted using the segmented point cloud data. Detailed results and visuals of these studies are presented under different headings in the following sections.

5.2.1. SEGMENTATION & CLASSIFICATION

5.2.1.1.Training the Deep Learning Network Using PointNet data with HBIM Laser Scanner Data

In the process of creating the segmented data obtained from the HBIM dataset, 5 MLP (Multi-Layer Perceptron) networks, 3 in the classification network and 2 in the segmentation network were used. In the model, in which the five-layer CNN (Convolutional Neural Network) is used for segmentation. As it is known, different functions and optimizers can be used in deep learning networks.

In this study, Adam optimizer and ReLu activation function were used in the training of the model. Also, all hyper- parameters of the network used for this segmentation are given in Table 5.2.

Table 5.2. Hyper Parameters of Proposed HBIM Network

Parameter	***Learning Rate***	***Max Epoch***	***Momentum***	***Batch Size***	***Optimizer***	***Point Number***
Value	0.001	50	0.9	24	Adam	4096

The results of the HBIM, which we created using a PointNet-based system, were recorded separately using both the PointNet data set and the HBIM data set. In this way, is compares the outputs and makes inferences about the target system. PointNet (Qi, Su, Mo, & Guibas, 2017) data consists of open source point clouds with 13 different labels. The high number of data in this dataset prevented it from reducing the performance despite the large number of labels.

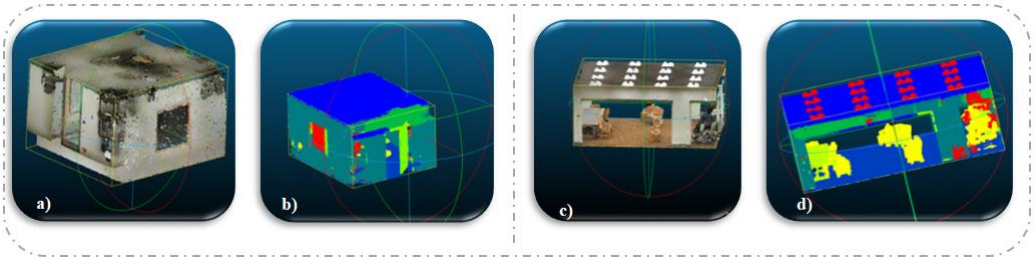


Figure 5.8. PointNet Dataset Results (a) 'RGB Laser Scanner data', (b) 'Segmented Laser Scanner data', (c) 'RGB S3DIS data', (d) 'Segmented S3DIS data'

Within the scope of this study, segmentation was made for 5 different labels, namely, door, window, wall, ceiling and floor. To show these labels on the Point Cloud, the window was expressed in red, the wall in dark green, the door in light green, and the floor and ceiling in blue. The .obj output with RGB color code consisting of laser scan data of the PointNet dataset and the .obj output obtained after segmentation is given in Figure 5.8. The .txt files showing the classification of the outputs in the.obj format given in Figure 5.8 are given in Figure 5.9. These results are the classification results from the deep learning network.

Accordingly, Figure 5.9 (a) holds the coordinate and class labels of the point cloud data used in the classification. Figure 5.9 (b) is the folder where the actual class values required to calculate the classification performance are kept.

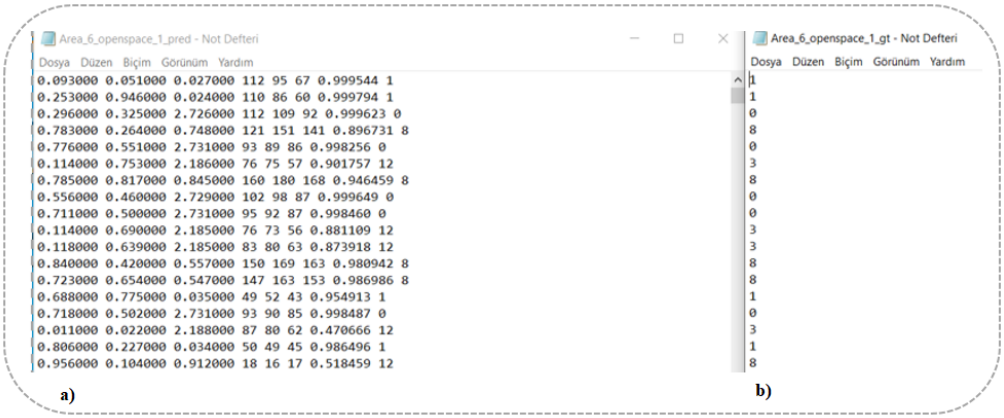


Figure 5.9. Classification results (a) Predicted values, (b) Actual values

Considering that the deep learning network used cannot always give accurate results, the boundaries of the network were tested with data close to each other at the coordinate level.

As seen in Figure 5.10, the deep learning network could not obtain effective results in these two labels due to the location of the doors and windows both in size and in the coordinate plane.

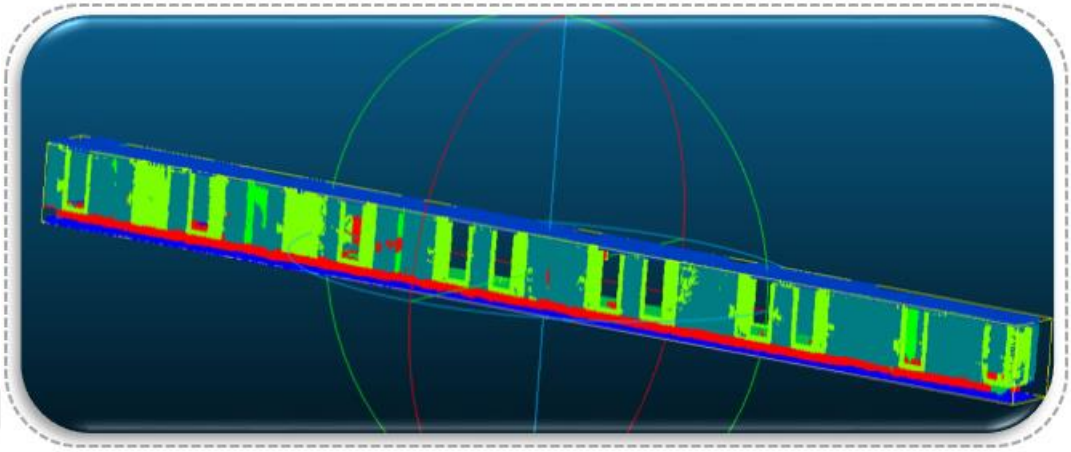


Figure 5.10. Segmented PointNet data

It can be seen that the room shown in Figure 5.10 is incorrectly segmented despite the high data number of PointNet and its high performance. While the places shown in green in this room should be segmented as windows (red) instead of doors, it is seen that they are misjudged due to the similarities between the door and the windows.

HBIM data were included in the system with 19 survey building data sets and 5 different labels (wall, door, window, ceiling, floor) created from 140 different rooms and 11 restitution building data sets.

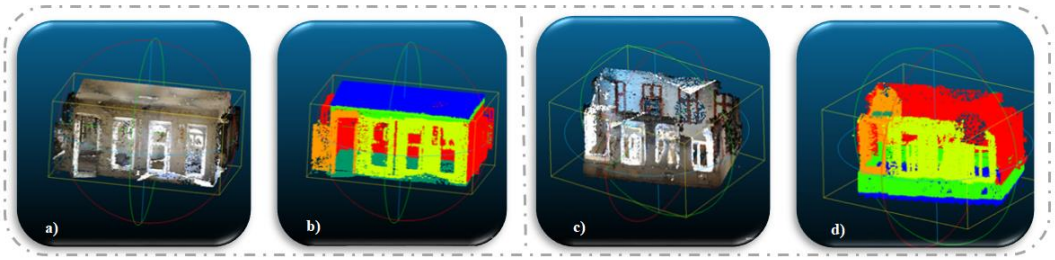


Figure 5.11. HBIM dataset (a) 'Structure_1_room_1' RGB data, (b) 'Structure_1_room_1' Segmented data,(c) 'Structure_1_room_2' RGB data, (d)' Structure_1_room_2' Segmented data

3D objects belonging to HBIM data and their images after segmentation are shown in Figure 5.11 and program outputs of classified HBIM data are shown in Figure 5.12.

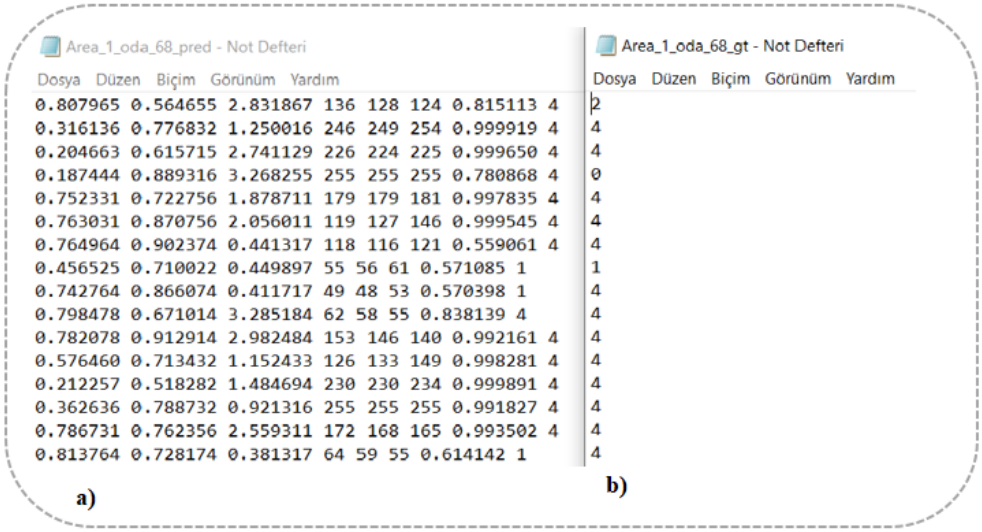


Figure 5.12. Classified data results (a) predicted values, (b) Actual values

In R & D (Research and Development) study, it is common to encounter times when all the results do not meet the expectations. In Figure 5.10, data belonging to a Standford university was shown to be incorrectly segmented. Similar to this result, showing incorrectly segmented data in the HBIM dataset will support the realism of the study. The studies on why the mis-segmented structure was mis-segmented and how improvements were conducted are detailed in the following sections. Figure 5.13 shows an incorrectly segmented room of the HBIM dataset. As can be seen, the windows (yellow) on the rear facade are segmented as the wall (red).

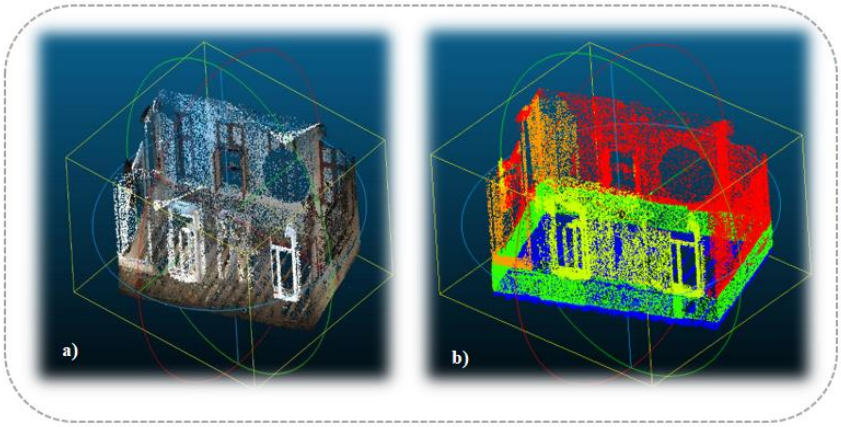


Figure 5.13. Incorrctly segmented data (a) 'HBIM RGB' data, (b) 'HBIM Segmented' data

Additionally, the accuracy and loss values obtained from the simulation results using the data of historical buildings in Gaziantep and PointNet data obtained from Stanford University within the scope of the Project are shown in detail in Table 5.3.

Table 5.3. Loss and Accuracy results of HBIM Network

Dataset	***Structure/ Room***	***Data Split***		***Accuracy***		***Loss***	
		Train	***Test***	***Train***	***Test***	***Train***	***Test***
HBIM (Laser Scanner data)	100 rooms in 18 HBIM structures	Area_1 Area_2 Area_3 Area_4 Area_5	Area_6	0.7147	0.5783	1.1179	1.8003
PointNet-Standford	271 rooms in 6 PointNet structures	Area_1 Area_2 Area_3 Area_4 Area_5	Area_6	0.8102	0.8200	0.7072	0.7526
HBIM + PointNet	100 rooms in 18 HBIM structures 40 rooms in 1 PointNet structures	Area_2 Area_3 Area_4 Area_5 Area_6	Area_1	0.9230	0.8793	0.1818	0.4410

When the table obtained is examined, the training accuracy of the HBIM laser scanning data is 71%, and the test accuracy is 57%. To determine the reason for the low test performance, it was observed how the network reacted to other datasets. Using the open source dataset created by Stanford University, 271 different rooms in 6 areas, the training accuracy of the network was 81% and the test accuracy was 82%.

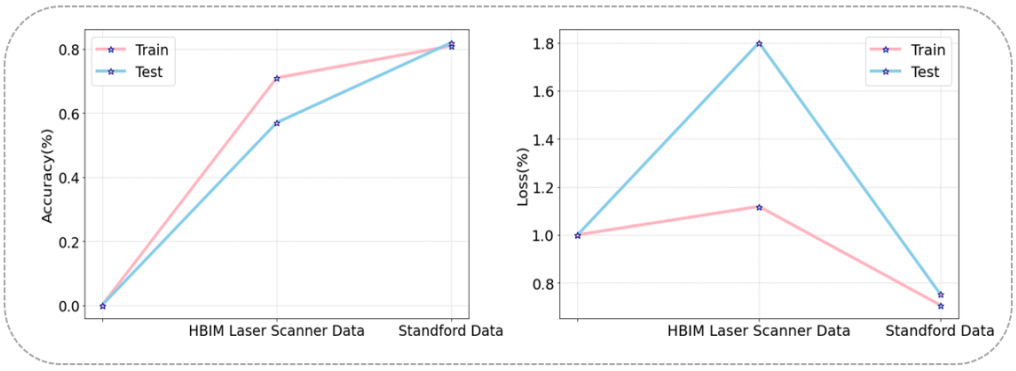


Figure 5.14. Results of different datasets (a) 'Accuracy' values, (b) 'Loss'values

Accuracy and loss values obtained when each data set is used separately are shown in Figure 5.14. The evaluation metrics obtained show the following:

If there is such a difference between the evaluation metrics obtained using the same network with different datasets, the problem that arises is not in the training network, but in the dataset used. To support this theory, 40 rooms of pointnet data was added to the HBIM dataset. Obtaining 97% training and 87% test success with the new dataset created shows that the deformations in the HBIM dataset reduces the evaluation quality of the network. For this reason, the synthetic data generation process detailed in the previous section has been started added to our original dataset. The results obtained at the end of this process are explained in the following sections.

5.2.1.2. Training the Deep Learning Network Using Synthetic data with HBIM Laser Scanner Data

Within the scope of efforts to increase the performance, restitution data of historical buildings converted from 3D model data to point cloud, suitable for the structure of historical buildings in Gaziantep, were created, their segmentation and classification performances were calculated by including them in the model.

The raw point cloud image of structure 25 from the synthetic data shown in Table 5.2 is shown in Figure 5.15. The other synthetic point cloud data included in the system are in the same format.

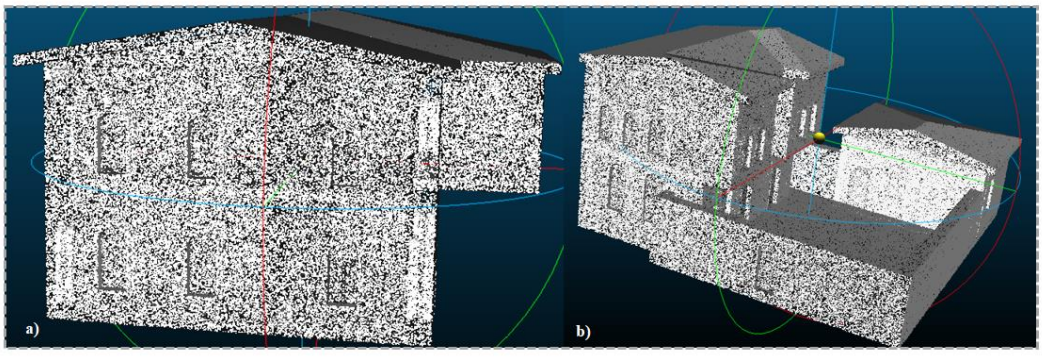


Figure 5.15. Synthetic data (a) ‘Structure 25’ (b) ’Structure 25’ side view

The .txt file showing the x, y, z, Nx, Ny, Nz coordinate values of the structure whose screenshot is given is shown in Figure 5.16.

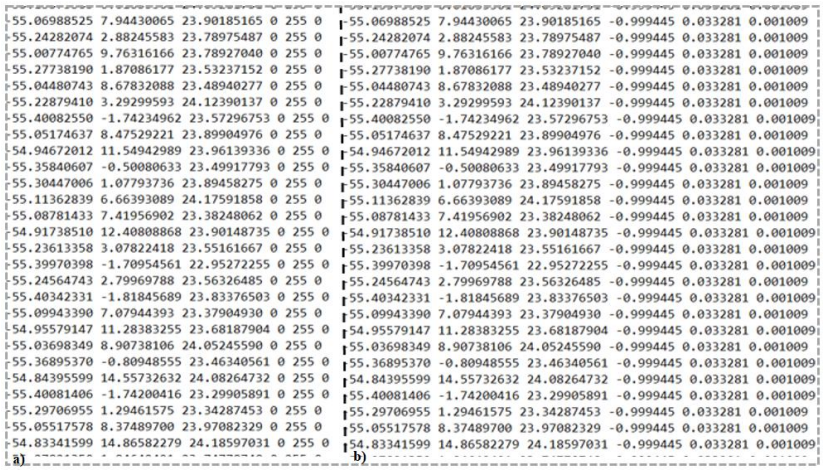


Figure 5.16. Coordinate files of structure 25 (a) Coordinate values, (b) Coordinate values with RGB colour codes

The aim of incorporating the above-mentioned synthetic structures into the HBIM network is to prevent the deformed HBIM data from degrading the performance of the network. 11 Synthetic data from the data shown in the Table 5.2 were added to the laser scanning data created as 100 rooms in 18 structures from the HBIM data. The dataset created by mixing the data is divided into 80% training and 20% test data. The results obtained using the newly created data set are detailed in Table 5.4.

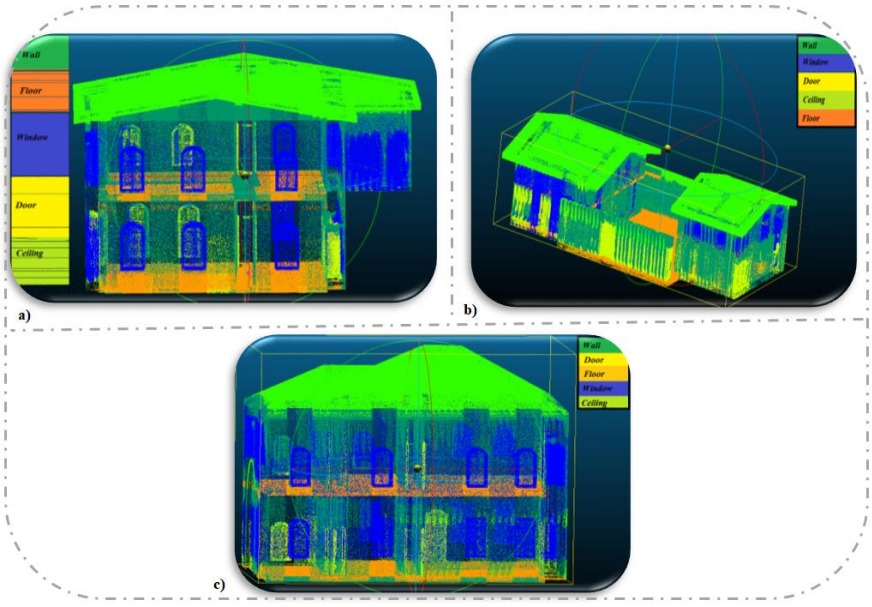


Figure 5.17. Segmented syntehetic data (a) 'Structure _25', (b)'Structure_29', (c) 'Structure_30'

Structure 25, structure 29 and structure 30, which are used as training and test data in Figure 5.17 , are used as random test data. 91%, 81%, and 86% success rates were

obtained from these data, respectively. In addition to the restitution building data in Table 5.2 , 13 survey building data of Gaziantep historical buildings (structure 6–18) were used as training data in the model, and 95% educational success was achieved. The purpose of creating restitution-based training data; to increase model training and accuracy performance. During this study, it was observed that increasing the quality of the training data also increased the segmentation accuracy. Sample test data from studies on segmentation of a set of restitution-based points belonging to the historical building are shown in Figure 4.6 (see section 4).

Table 5.4. HBIM Network Performance

Dataset	**Structure/ Room**	**Data Split**		**Accuracy(%)**		**Loss**	
		Train	**Test**	**Train**	**Test**	**Train**	**Test**
HBIM (Laser Scanner data)	100 rooms in 18 HBIM structures	Area_1 Area_2 Area_3 Area_4 Area_5	Area_6	0.7147	0.5783	1.1179	1.8003
HBIM + Standford	100 rooms in 18 HBIM structures + 40 rooms in 1 PointNet Data	Area_2 Area_3 Area_4 Area_5 Area_6	Area_1	0.9230	0.8793	0.1818	0.4410
HBIM	100 rooms in 18 HBIM structures + 11 synthetic data	structure_6 structure _7 … structure _21 structure _22	structure _30	0.9514	0.8652	0.3320	0.4150
			structure _29	0.9514	0.8159	0.3320	0.7734
			structure _25	0.9514	0.9120	0.3320	0.3250

When the Table 5.4 is examined, it is revealed once again that the model performance mentioned in the previous section is highly dependent on the dataset. While the test

performance of the network was 57%, the performance increased to 87% with the inclusion of some of the pointnet data in the dataset. Subsequently, synthetic data was included in the system in we use the original data, and test success was achieved between 81 and 91% was achieved.

The training accuracy obtained from training the HBIM network with different datasets was recorded as 71% when only deformed data were used, 91% when pointnet and deformed data were used, and 95% when deformed and synthetic data were used.

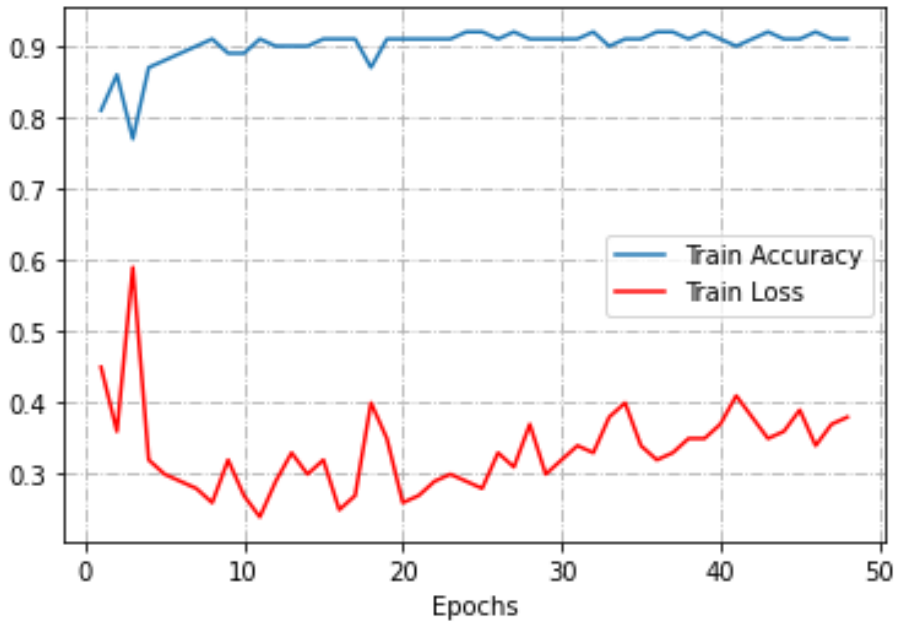


Figure 5.18. Train Performance of HBIM

When both training and test results of the deep learning network are examined, it is seen that the best performance is achieved when synthetic data is used together with deformed data. This shows that the creation of synthetic data and its inclusion in the network is the right approach. Figure 5.18 shows the training performance of the HBIM learning network using completely original data.

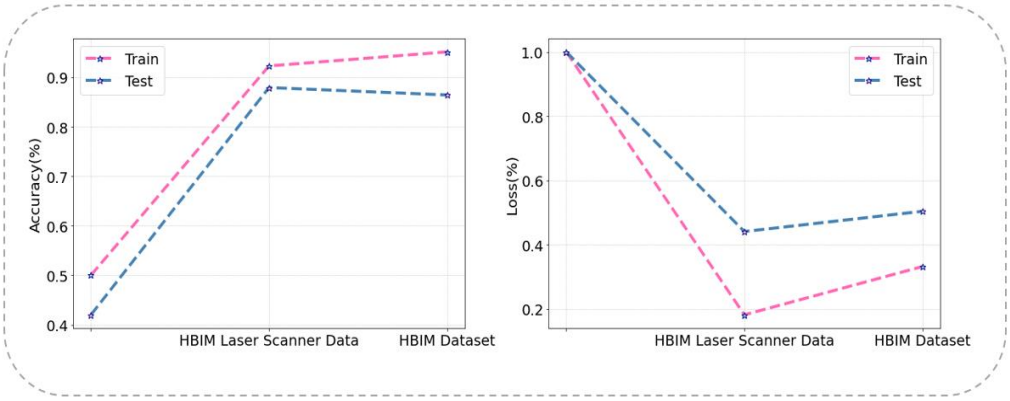


Figure 5.19. Performance comparasion od datasets

The performance comparison of the sandford data included in the system to support the deformed and the synthetic data created by us is shown in Figure 5.19.

When the given tables and graphs are examined, a situation causes a sudden decrease and increase in the loss values of the training and test data. The reason for these irregular changes is the presence of data that reduces the quality of both the training and the test data. To eliminate such fluctuations, manual methods called data selection have been included in the literature. These fluctuations, which stand out especially in studies using point cloud data, are caused by the variable number of points (density), noise and data quality contained in each data. To solve this problem; some results were obtained using different test and training groups. These results are especially important for testing each structure alone and determining its effect on the model alone. Additionally, it is important to choose the data to be used in the network so that many of which data should be used, whether the model memorized it or not, so that the performance does not decrease after a certain iteration.

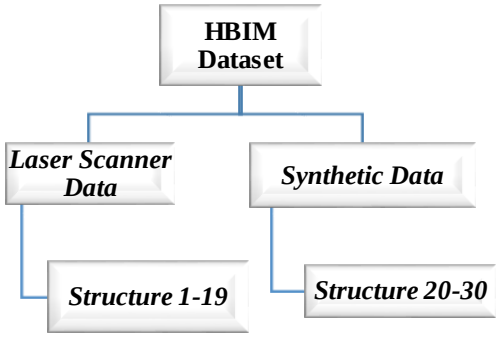


Figure 5.20. Number of HBIM dataset

In the Figure 5.20, a diagram of the laser scanning and synthetic data used for the training of the network is detailed. The results of the training and test data selected from these data are shown in Table 5.5.

Table 5.5. Selected HBIM Data Results

Train Data	***Data Type***	***Test Data***	***Number of Room***	***Accuracy (%)***	***Avarage Accuracy***
Structure 6 Structure 7 Structure 8 Structure 9 Structure 10 Structure 11 Structure 12 Structure 13 Structure 14 Structure 15 Structure 16 Structure 17 Structure 18	*Laser Scanner Data*	Structure *1*	*Room 1*	*0.911393*	*%81.28*
			Room 2	*0.839315*	
			Room 3	*0.858211*	
			Room 4	*0.822293*	
			Room 5	*0.858130*	
		Structure *2*	*Room 6*	*0.862083*	
			Room 7	*0.907064*	
		Structure *3*	*Room 8*	*0.838172*	
			Room 9	*0.811369*	
			Room 10	*0.847816*	
		Structure *4*	*Room 11*	*0.893998*	
			Room 12	*0.779090*	
			Room 13	*0.801532*	
			Room 20	*0.838041*	
		Structure *5*	*Room 22*	*0.380605*	
Structure 19 Structure 20	*Synthetic Data*	Structure *25*	*Not devided*	*0.772353*	*%82.98*
		Structure *29*		*0.842225*	
		Structure *30*		*0.858911*	

5.2.3. MESH MODEL GENERATION

The work involving mesh model creation was completed using the point cloud data obtained from two methods: laser scanning of 19 buildings incorporating 140 rooms and converting restitution BIM models of 11 buildings into the point cloud data synthetically. The purpose of creating the latter data type, which is called the synthetic data in this study, was to contribute to the training dataset of the system that can perform automatic segmentation based on deep learning methods. All buildings in the data set represent historical vernacular architecture in the city of Gaziantep.

As can be seen in the graph, the accuracy of the model trained using HBIM data reached over 0.92, the training loss went down as low as 0.30. As seen, the synthetic data added to the model training improved the model's test success. Simultaneously, the HBIM-integrated system built on the PointNet architecture in this study gave the closest results to the training performance obtained in the original PointNet study (Qi, Su, Mo, & Guibas, 2017).

It is an important issue to obtain the closest results to this architecture, which has a place in the literature. The synthetic data produced is at least as good as the results obtained from the studies conducted with the original data, it shows that the production is made in the closest way to the restitution data and is used in the right ratio in the network.

In this study, mesh objects were obtained from segmented HBIM data using deep learning methods. The methods and parameters used for meshes are created using the Poisson Reconstruction Method as introduced above. Poisson mesh representation is based on Poisson equations (Kazdan, 2005; Kazdan, Bolitho, & Hoppe, 2006). The tree depth to be used in mesh creation is 8; the octree value is 0; the ratio between the cubes that will form the mesh is 1.1, and the result obtained without using linear interpolation in the reconstruction process is shown in Figure 3.18 (see section 3). In mesh creation, the octree value and the ratio between cubes are usually kept constant and do not affect the result. The parameter that expresses the object depth (depth) is defined with a value between 1 and 9 in large point cloud data.

The greater the surface depth to be created on the point cloud is, the more detailed the reconstruction process occurs (Kazdan, Bolitho, & Hoppe, 2006).

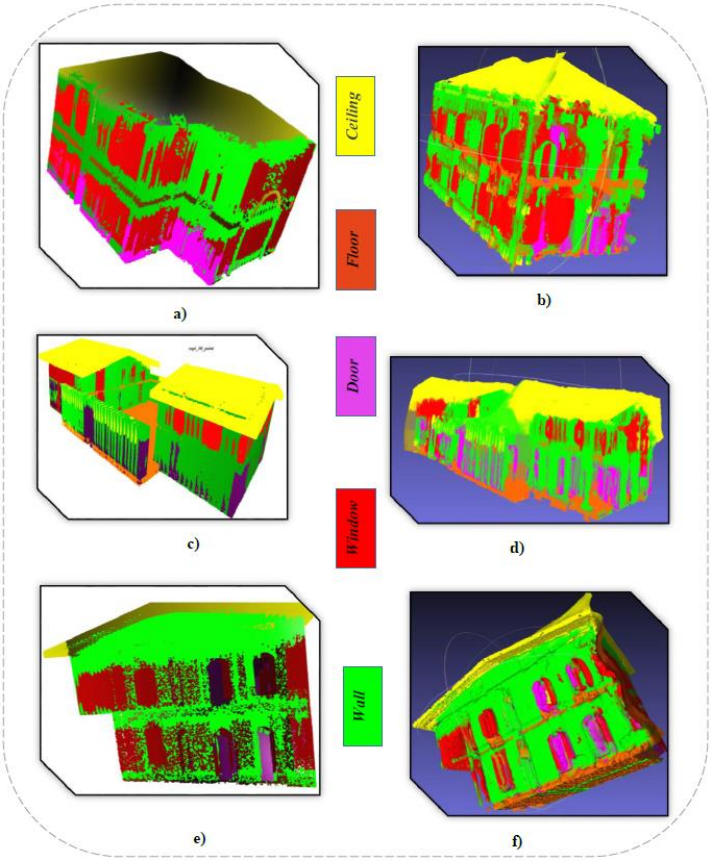


Figure 5.21. Poisson Mesh Reconstruction using synthetic data
(a) 'Building_25' Point Cloud, (b) 'Building_25' Poisson Mesh, (c) 'Building_30' Point Cloud, (d) 'Building_30' Poisson mesh, (e) 'Building_29' Point Cloud, (f) 'Building_29' Poisson mesh

High-density datasets often contain noise, and the optimum value for the depth constant used to create the mesh from these datasets must be found. If the value used is too large, noises that do not belong to the original data are included in the mesh, or if it is too low, the points belonging to the original data can be ignored (Peng, et al., 2021; Vyas, Jiang, Liu, & Ostadabbas, 2021; Li, Wan, Cheng, & Cui, 2010; open3d, 2021).

In Figure 5.21, the results of the surface creation process from the segmented point cloud data obtained from Building_25, Building _29 and Building _30 of the HBIM dataset were visualised. Separate colour codes are defined for 5 different labels for door, window, wall, floor and ceiling.

According to the results, Building_29 and Building_30 produced from the synthetic data that rely on the restitution data have test success over 80 percent and Building_25 has test success 77 percent.

The success rate of the synthetic data can be attributed to the data width and the number of points in the point cloud produced.

The fact that the success obtained from the synthetic data does not fall below 70 percent indicates that the data suitable for the trained network is produced. This study increased the training and test performance of the network in which laser scanning data was used together with the synthetic data. In the following sections, the results obtained by combining the data are detailed.

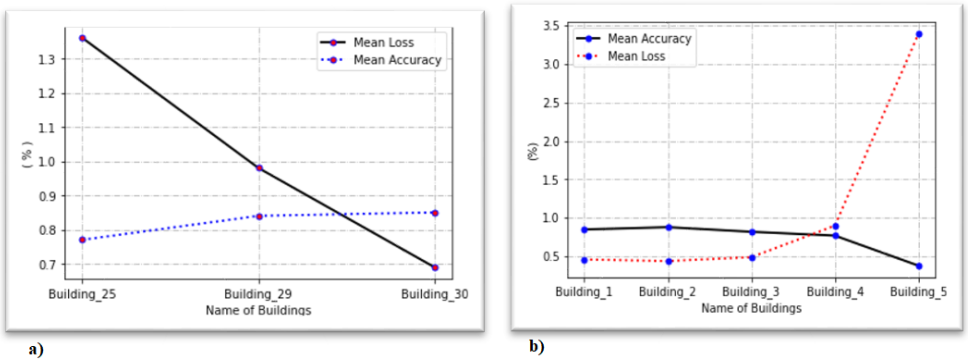


Figure 5.22. HBIM Data Test Results
(a) 'Synthetic Data' Test results, (b) 'Laser Scanner Data' Test results

Figure 5.22 also shows the test results of the HBIM laser scan data. According to these results, the highest performance was obtained from Building_2, and the lowest performance was from Building_5. The poor performance of Building_5 is due to high point loss and the amount of noise in the data. Results from laser scanning data and synthetic data are shown in Table 5.6.

Table 5.6. Test Result from HBIM test data

Data	Types of Data	Average Accuracy (%)	Average Mean Loss
Structure 25	Synthetic data	0.772353	1.367686
Structure 29		0.842225	0.984102
Structure 30		0.858911	0.698297
Structure 1	3D Laser Scanner data	0.85	0.46
Structure 2		0.88	0.44
Structure 3		0.82	0.49
Structure _4		0.77	0.90
Structure 5		0.38	3.39

In Figure 5.23, the surface reconstruction processes of some of the 3D laser scanning data of the HBIM dataset are shown. There are point deficiencies in the data obtained from the 3D laser scanning scanner due to the deformations that occur over time. Considering that each point connects with the 2 closest points in the mesh creation process, it is seen that the coordinate losses in the data affect the mesh quality .Common parameters are used in the reconstructed synthetic and 3D laser scanning data in Figure 5.21 and 5.23.

The difference in quality between the mesh obtained from synthetic data and the mesh obtained from laser scanning data is clear.

It has been demonstrated that this difference arises from the deterioration of point neighbourhoods and the inability to calculate the surface equations depending on the coordinate losses in the fractured areas in the historical buildings over time. From HBIM data of 30 buildings, 80% training and 20% test data sets were used in the deep learning network that completed the segmentation module.

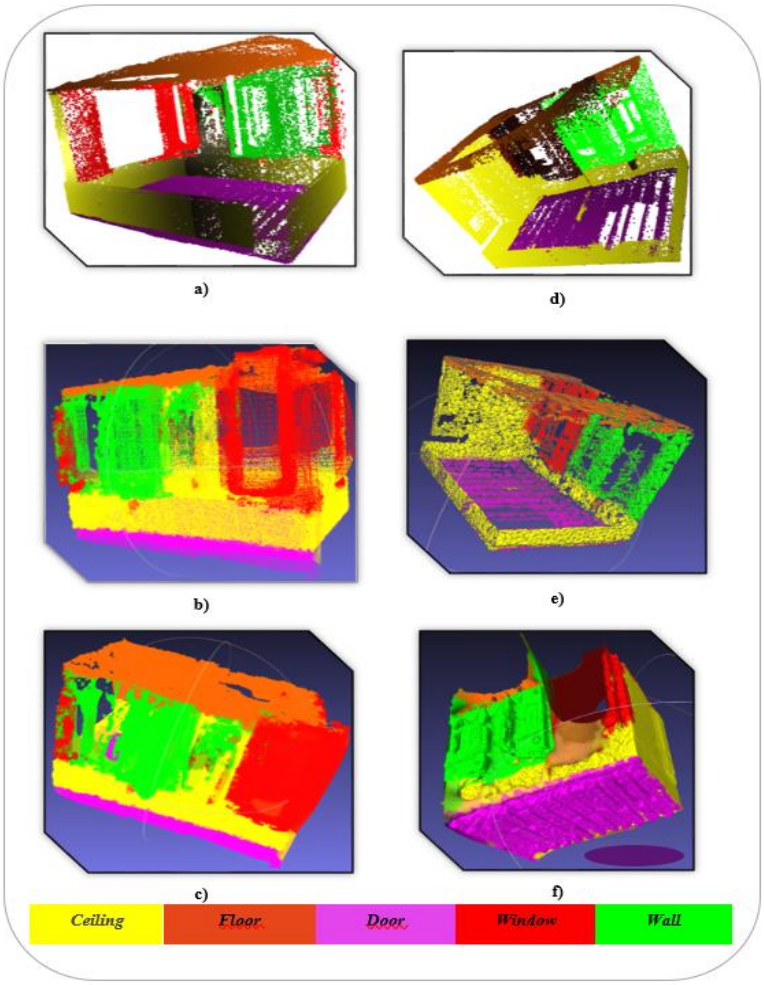


Figure 5.23. Poisson Reconstruction using 3D laser scanner data (a)'Building_1_room_1'point cloud, (b)'Building_1_room_1' vertex renders, (c)'Building_1_room_1' Poisson mesh, (d)'Building_3_room_8' point cloud, (e)'Building_3_room_8' vertex renders, (f) 'Building_3_room_8' Poisson mesh

The IoU (Intersection Over Union) values and classification performance obtained using these test data in different combinations are listed in Table 5.7 in detail.

As the table presents the segmentation performances of synthetic data and 3D laser scanning data differ according to the labels. Better segmentation performance is achieved due to the less point losses in the 'floor' object, which has the highest IoU value in 3D laser scanner data. The segmentation performance decreased due to the similarity of the object specified as 'window' with the door in both size and visually.

Table 5.7. Test Result from HBIM test data

Data	*Data Type*	*Window (IoU)*	*Wall (IoU)*	*Ceiling (IoU)*	*Door (IoU)*	*Floor (IoU)*	*Avg.*	*Acc (%)*
Structure 25 *Structure 29* *Structure 30*	*Synthetic data*	0.506	0.458	0.887	0.294	0.821	0.494	82.9
Structure 1 *Structure 2* *Structure 3* *Structure 4* *Structure 5*	*3D Laser Scanner data*	0.251	0.537	0.6414	0.856	0.838	0.521	81.2
Structure 1 *Structure 2* *Structure 3* *Structure 4* *Structure 5* *Structure 25* *Structure 29* *Structure 30*	*Synthetic Data* +	0.483	0.464	0.855	0.594	0.821	0.536	82.8

The graph of the IoU results of the test data is shown in Figure 5.24.

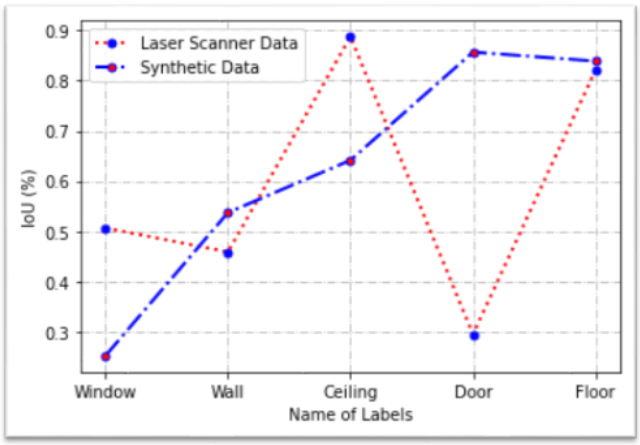


Figure 5.24. The IoU results for five labels

The aim of this study was to accurately segment historical structures and to convert segmented point clouds into mesh models with novel methods. For this reason, the fact that the average IoU value of the synthetic data included in the system is lower than the average IoU value obtained from the 3D laser scanning data indicates that the synthetic data is served for its intended purpose. In addition, the overall system accuracies for the data types are close to each other and over 80% indicates that the synthetic data are produced close to the originals and support the system performance.

CHAPTER 6

CONCLUSION

In this thesis, it is aimed to automatically produce building information models (BIM) from the point cloud of historical buildings to record losses that may arise from human and natural disasters, to provide smart management of historical buildings, and to support correct maintenance-repair and restoration works. In the study, the was selected among the historical heritage buildings in Gaziantep as case studies. Algorithm and software development activities are carried out to produce automatic BIM models by using the dataset in the form of points of selected case studies. In this process, object recognition and pattern matching techniques that will provide the best results are examined and their potential to transform into an automatic recognition system is evaluated. To place the study in a sustainable context, the maintenance and management processes and scenarios of these intelligent and parametric models within the cultural heritage building information modelling and management (HBIM) systematics are also examined.

To create the HBIM system, the studies in which the segmentation of historical structures were examined and it was seen that the studies in this area were insufficient. In the studies examined and detailed in the previous sections, the S3DIS dataset, which is frequently mentioned in the literature, was used. The S3DIS dataset consists of historical but undeformed structures. The data used in this thesis consist of completely unique and deformed historical buildings in Gaziantep.

In the training of the created deep learning network, in addition to the original laser scanning data, point cloud data obtained from the restitution data of historical buildings were used.

These data were named as synthetic data within the scope of the thesis.In the training and testing of the deep learning network, 19 laser scan data and 10 synthetic data were used. The aim of this thesis is to segment the deformed laser scan data and then obtain the mesh model.

To this result, the number of synthetic data used in the network has been increased or decreased in a way that does not affect the success of the target system. In this thesis, the scanned data of the historical buildings that were deteriorated and deformed were used, 83,30 % test and 95,14 % training success was achieved even though it did not contain sufficient information about the structure due to the deformations in the buildings. Segmentation for point cloud data of historic buildings can be challenging and AI based algorithms can be insufficient due to their unique, out of standards and deformed conditions.

However, preparing training data set from the restitution information of the historic building that is called restitution data in this research helps significantly for high accurate segmentation. This restitution data and laser scanning data were used together for segmentation of five components (windows, doors, wall, ceiling and floor). The reason for 5 components is because the case study heritage buildings are deteriorated and deformed from which the segmented results were still satisfactory.

The results show that the combined use of restitution data and existing conditions data together would be way forward for the point cloud segmentation with AI for heritage structures belonging to the same period. Therefore, the research will expand further by identifying other minor components in the case study buildings by preparing a training dataset for the algorithm towards enhanced and detailed segmentation with higher accuracy. In addition, PointNet ++ (Qi, Yi, Su, & Guibas, 2017), an improved system of PointNet (Qi, Su, Mo, & Guibas, 2017), can provide better segmentation performance with the approach proposed in the paper.

Therefore, as an expansion from the current research, PointNet++ will be also considered to improve the segmentation as part of the research plan that will be conducted on R-CNN and Fast R-CNN networks to incorporate the unlabelled data into the HBIM network.

Another method presented this thesis and detailed in the previous sections is to create a mesh model from a segmented point cloud data. Poisson Mesh method, which is a method of the CGAL library offered as an open source in python, has been used in mesh model creation studies unlike the deep learning model. During the segmentation, the labels that the deep learning model confuses the most with each other are the door and window.

During the mesh model creation stage, the IoU result for the door was 0.59, and that for the window is 0.48. According to the results obtained from the mesh model creation studies; When the connections established by each point with its neighbourhoods are expressed as vectors, the complexity and resolution of the resulting vector have a direct effect on the mesh model accuracy. When the structural elements are evaluated separately and evaluated as a whole, the number of points on an axis changes.

Therefore, the normal vector can also change.Here, when a unique equation of each building element is used to express the building element, the result is smooth and understandable as seen in the previous sections.

Creating a general mesh for a structure is the determination of a general shape and surface normals of that structure.The process of creating a common surface for labels with 5 different formats did not achieve the expected success, especially on missing data.

A reason why mesh generation fails with missing data is that points that are not in the structures are automatically filled according to the normals determined using the meshing process. If an object is thought to be placed in a cube, the missing parts required for the object to take the cube form are automatically completed.During this automatic completion, the problem of where the added region belongs (to which label/which building element) should be well resolved.

Within the scope of this thesis, segmentation and mesh model studies, which are the basic building blocks to creating the HBIM system, have been completed with the methods and success rates described above.The methods used in this thesis are aimed to be a precedent for future studies.

REFERENCES

Alanyurt, U. (2009). Türkiye'de Koruma ve Onarım Üzerine Analiz. *Masrop.*

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. *In 2017 international conference on engineering and technology (ICET)*, 1-6.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., & Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches.

Andriasyan, M., Moyano, J., Nieto-Julián, J. E., & Antón, D. (2020). From point cloud data to building information modelling: An automatic parametric workflow for heritage. *Remote Sensing*, 1094.

Arabacıoğlu, P., & Aydemir, I. (2007). The Concept Of Revalorization in Historical Environments. *YTÜ Arch. Fac.*, 1-9.

Arayici, Y. (2007). An approach for real world data modelling with the 3D terrestrial laser scanner for built environment. *Automation in construction, 16(6)*, 816-829.

Arayici, Y. (2008). Towards building information modelling for existing structures. *Structural Survey*.

Arayici, Y., Counsell, J., Mahdjoubi, L., Nagy, G., Ḥawwās, S. Z., & Dweidar, K. (2017). Heritage building information modelling. *Abingdon: Routledge*.

Arayici, Y., Hamilton, A., & Gamito, P. (2005). Built environment reverse and forward prototyping.

Arayici, Y., Hamilton, A., & Gamito, P. (2006). Modelling 3D scanned data to visualise and analyse the built environment for regeneration. *Surveying and Built Environment, 17(2)*, 7-28.

Arayici, Y., Hamilton, A., Gamito, P., & Albergaria, G. (2005). Using the 3D laser scanner data capture and modelling to visualise the built environment: Data capture and modelling. London.

Baykara, M., Ertürkler, M., Gül, M., & Harputoğlu, M. (2004). Karaciğer Dokusundaki Nekroz Alanının Doku Tabanlı Bölütleme Kullanılarak Belirlenmesi ve İncelenmesi. Asyu-inista.

Berger, M., Levine, J. A., Nonato, L. G., Taubin, G., & Siva, C. T. (2013). A benchmark for surface reconstruction. ACM Transactions on Graphics (TOG). *32(2)*, 1-17.

Beyaz.net. (2022, May 5). Retrieved from Beyaz.net makaleler: https://www.beyaz.net/tr/yazilim/makaleler/derin_ogrenme_deep_learning_nedir.html

Bezdek, J. C. (1981). Objective function clustering. In Pattern recognition with fuzzy objective function algorithms. *Springer*, 43-93.

Bosch, A., Zisserman, A., & Munoz, X. (2007). Image classification using random forests and ferns. *In 2007 IEEE 11th international conference on computer vision* (pp. 1-8). IEEE.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine, 34(4)*, 18-42.

Brumana, R., Oreni, D., Raimondi, A., Georgopoulos, A., & Bregianni, A. (2013, October). From survey to HBIM for documentation, dissemination and management of built heritage: The case study of St. Maria in Scaria d'Intelvi. *In 2013 Digital Heritage International Congress (DigitalHeritage), Vol.1*, 497-504.

Burrus, C. S. (1997). Introduction to wavelets and wavelet transforms: a primer. *Englewood Cliffs*.

Campbell, R. J., & Flynn, P. J. (2001). A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding, 81(2)*, 166-210.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence, (6)*, 679-698.

Çarkacı, N. (2022, May 21). *Medium*. Retrieved from Medium: https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4

Cebeci , H. (2019, 11 17). *Yapay Zeka ve Derin Öğrenme A-Z™: Tensorflow*. Retrieved from Yapay Zeka ve Derin Öğrenme A-Z™: Tensorflow.

CGAL. (2022, June 02). Retrieved from CGAL 5.4 - Poisson Surface Reconstruction: https://doc.cgal.org/latest/Poisson_surface_reconstruction_3/

cgal.com. (2022, june 02). Retrieved from cgal.com: https://www.cgal.org/

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., & Yu, F. (2015). Shapenet: An information-rich 3d model repository.

Cry, C. M., & Kimia, B. B. (2004). A similarity-based aspect-graph approach to 3D object recognition. *International Journal of Computer Vision, 57(1)*, 5-22.

Dai, A., Ruizhongtai Qi, C., & Nießner, M. (2017). Shape completion using 3d-encoder-predictor cnns and shape synthesis. *In Proceedings of the IEEE conference on computer vision and pattern recognition* , 5868-5877.

Dong, Z., Zhang, R., & Shao, X. (2019, November). A CNN-RNN Hybrid Model with 2D Wavelet Transform Layer for Image Classification. *In 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)* , 1050-1056.

Dore, C., & Murphy, M. (2013). Semi-automatic modelling of building facades with shape grammars using historic building information modelling. International

Archives of the Photogrammetry. *Remote Sensing and Spatial Information Sciences, 5*.

Dube, D., & Zell, A. (2011). Real-time plane extraction from depth images with the randomized hough transform. *In 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 1084-1091. https://ieeexplore.ieee.org/abstract/document/6130371/?casa_token=12PczqKImt4AAAAA:O3Kl-2T8bvGt58u1LJcHkuzHrlUgRXJmquxX-PbXGajj6uYjOD9NbW7xTXlB98oEEW6TnStKHg adresinden alındı

Dunn, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 32-57.

Fai, S., & Sydor, M. (2013, October). Building Information Modelling and the documentation of architectural heritage: Between the 'typical'and the 'specific'. *In 2013 Digital Heritage International Congress (DigitalHeritage), 1*, 731-734.

Feghi, E., Tahar, A., & Ahmadi, M. (2011, June). Efficient Features Extraction for Fingerprint Classification with Multi Layer Perceptron Neural Network. *In ISSCS 2011-International Symposium on Signals, Circuits and Systems, 978-1-61284-943-0*, 1-4.

Fortuner, B. (2019, 05 07). *Github.* Retrieved from Github: https://github.com/bfortuner/ml-glossary/blob/master/docs/loss_functions.rst.

Frome, A., Singer, Y., & Malik, J. (2006). Image retrieval and classification using local distance functions. *Advances in neural information processing systems*, 19.

Garagnani, S., & Manferdini, A. M. (2013). Parametric accuracy: building information modeling process applied to the cultural heritage preservation. International Archives of the Photogrammetry. *Remote Sensing and Spatial Information Sciences, 5(1)*, 87-92.

Golovinskiy, A., Kim, V. G., & Funkhouser, T. (2009, September). Shape-based recognition of 3D point clouds in urban environments. *n 2009 IEEE 12th International Conference on Computer Vision*, 2154-2161.

Gonzalez, R., & Woods, R. (1993). *Digital Image Processing.* Addison Wesley Publishing Company.

Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.

Graves, A., Fernández, S., & Schmidhuber, J. (2007, September). Multi-dimensional recurrent neural networks. *In International conference on artificial neural networks* , 549-558.

Griffiths, D., & Boehm, J. (2019). SynthCity: A large scale synthetic point cloud.

Grilli, E., & Remondino, F. (2020). Machine learning generalisation across different 3D architectural heritage. *ISPRS International Journal of Geo-Information, 9(6)*, 379.

Grilli, E., Dininno, D., Petrucci, G., & Remondino, F. (2018). From 2D to 3D supervised segmentation and classification for cultural heritage applications. *In ISPRS TC II Mid-term Symposium "Towards Photogrammetry 2020", 42*(42), 399-406.

Grilli, E., Farella, E. M., Torresani, A., & Remondino, F. (2019). Geometric feature analysis for the classification of cultural heritage point clouds. *In 27th CIPA International Symposium "Documenting the past for a better future", 42*, 541-548.

Habi, H. V., & Messer, H. (2018, July). Wet-dry classification using lstm and commercial microwave links. *In 2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 149-153.

Habi, H. V., & Messer, H. (2020). Recurrent neural network for rain estimation using commercial microwave links. *IEEE Transactions on Geoscience and Remote Sensing, 59(5)*, 3672-3681.

Han, X., Li, Z., Huang, H., Kalogerakis, E., & Yu, Y. (2017). High-resolution shape completion using deep neural networks for global structure and local geometry inference. *In Proceedings of the IEEE international conference on computer vision*, 85-93.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition* , 770-778.

Hichri, N., Stefani, C., De Luca, L., Veron, P., & Hamon, G. (2013). From point cloud to BIM: a survey of existing approaches. *In XXIV International CIPA Symposium (p. na). Proceedings of the XXIV International CIPA Symposium, XL5/W2*(5), 343-348.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science, 313(5786)*, 504-507.

Hou, F., Lei, W., Li, S., Xi, J., Xu, M., & Luo, J. (2021). Improved Mask R-CNN with distance guided intersection over union for GPR signature detection and segmentation. *Automation in Construction, 121*.

Huang, T. S., Schreiber, W. F., & Tretiak, O. J. (1971). Image processing. *Proceedings of the IEEE, 59(11)*, 1586-1609.

Huart, J., & Bertolino, P. (2005, September). Similarity-based and perception-based image segmentation. *In IEEE International Conference on Image Processing 2005 , 3*, III-1148.

Jaccard, P. (1908). Nouvelles recherches sur la distribution florale. *Bull. Soc. Vaud. Sci. Nat, 44*, 223-270.

Jayakumar, D., Elakkiya, A., Rajmohan, R., & Ramkumar, M. O. (2020, July). Automatic prediction and classification of diseases in melons using stacked RNN based deep learning model. *In 2020 international conference on system, computation, automation and networking (ICSCAN)*, 1-5.

Jusman, Y., Firdiantika, I. M., Dharmawan, D. A., & Purwanto, K. (2021, March). Performance of Multi Layer Perceptron and Deep Neural Networks in Skin Cancer Classification. *In 2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech)*, 534-538.

Kaggle. (2020). Retrieved from Kaggle: https://www.kaggle.com/c/cifar-10

Kaggle. (2020). Retrieved from Kaggle: https://www.kaggle.com/datasets/fedesoriano/cifar100

Kalogerakis, E., Averkiou, M., Maji, S., & Chaudhuri, S. (2017). 3D shape segmentation with projective convolutional networks. *In proceedings of the IEEE conference on computer vision and pattern recognition*, 3779-3788.

Kaur, A. (2022). *Medium*. Retrieved from Medium.com: https://medium.com/@arshdeepkaur/neural-network-hyperparameter-tuning-in-a-nutshell-8c2b9a14a2c0

Kayalibay, B., Jensen, G., & van der Smagt, P. (2017). CNN-based segmentation of medical imaging data. arXiv preprint arXiv:1701.03056.

Kazdan, M. (2005, July). Reconstruction of solid models from oriented point sets. *In Proceedings of the third Eurographics symposium on Geometry processing*, 73-es.

Kazdan, M., Bolitho, M., & Hoppe, H. (2006, June). Poisson surface reconstruction. *In Proceedings of the fourth Eurographics symposium on Geometry processing*, *7*.

KDnuggets. (2022, 05 30). Retrieved from KDnuggets: https://www.kdnuggets.com/2019/06/gradient-descent-algorithms-cheat-sheet.html

Klokov, R., & Lempitsky, V. (2017). Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. *In Proceedings of the IEEE international conference on computer vision* , 863-872.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*.

KUDEB. (2022, June 02). Retrieved from kudeb.ibb.istanbul: https://kudeb.ibb.istanbul/

Kumar, R., Arthanari, M., & Sivakumar, M. (2012, March/June). Image Segmentation using Discontinuity-Based Approach. , *International Journal Multimedia and Image Processing (IJMIP), 2*(1/2).

Landrieu, L., & Simonovsky, M. (2018). Large-scale point cloud semantic segmentation with superpoint graphs. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 4558-4567.

Le, T., & Duan, Y. (2018). Pointgrid: A deep network for 3d shape understanding. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 9204-9214.

Le, T., Bui, G., & Duan, Y. (2017). A multi-view recurrent neural network for 3D mesh segmentation. *Computers & Graphics, 66*, 103-112.

Lee, A., Marshall-Ponting, A. J., Aouad, G., Wu, S., Koh, W. I., Fu, C., . . . Fisher, M. (2003). Developing a vision of nD-enabled construction. *Construct IT*.

Lee, R. (2022, 06 03). *Medium*. Retrieved from Medium: https://hengapher.medium.com/confusion-matrix-precision-recall-f1-score-76e579eb2d48

Leixian, S. (2022, June 02). *Researchgate*. Retrieved from www.researchgate.net: https://www.researchgate.net/figure/Softmax-function-image_fig1_325856086

Li, X., Wan, W., Cheng, X., & Cui, B. (2010, November). An improved Poisson surface reconstruction algorithm. *In 2010 International Conference on Audio, Language and Image Processing*, 1134-1138.

Lyu, Y., & Huang, X. (2018). Road segmentation using CNN with GRU.

Macher, H., Landes, T., Grussenmeyer, P., & Alby, E. (2014, November). Semi-automatic segmentation and modelling from point clouds towards historical building information modelling. *In Euro-Mediterranean Conference*, 111-120.

MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. *In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, 1*(14), 281-297.

Mader, K. S. (2021). *Kaggle*. Retrieved from Kaggle.com: https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000

Mallat, S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence, 11(7)*, 674-693.

Manson, J., Petrova, G., & Schaefer, S. (2008, July). Streaming surface reconstruction using wavelets. *In Computer Graphics Forum, 27*(5), 1411-1420.

Matei, B., Shan, Y., Sawhney, H. S., Tan, Y., Kumar, R., Huber, D., & Hebert, M. (2006). Rapid object indexing using locality sensitive hashing and joint 3D-signature space estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(7)*, 1111-1126.

Maturana, D., & Scherer, S. (2015, September). Voxnet: A 3d convolutional neural network for real-time object recognition. *In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 922-928.

Metz, C. E. (1978, October). Basic principles of ROC analysis. In Seminars in nuclear medicine. *8*(4), 283-298.

Mian, A. S., Bennamoun, M., & Owens, R. A. (2005). Automatic correspondence for 3D modeling: an extensive review. *International Journal of Shape Modeling, 11(02)*, 253-291.

Murtiyoso, A., & Grussenmeyer, P. (2020). Virtual disassembling of historical edifices: Experiments and assessments of an automatic approach for classifying multi-scalar point clouds into architectural elements. *Sensors, 20(8)*, 2161.

Naraei, P., Abhari, A., & Sadeghian, A. (2016, December). Application of multilayer perceptron neural networks and support vector machines in classification of healthcare data. *In 2016 Future Technologies Conference (FTC)*, 848-852.

Öğündür, G. (2022, June 02). *Medium*. Retrieved from Medium: https://medium.com/@gulcanogundur/do%C4%9Fruluk-accuracy-kesinlik-precision-duyarl%C4%B1l%C4%B1k-recall-ya-da-f1-score-300c925feb38

Ohtake, Y., Belyaev,, A., & Seidel, H. P. (2005, June). An integrating approach to meshing scattered point data. *In Proceedings of the 2005 ACM symposium on Solid and Physical Modeling*, 61-69.

open3d. (2021). Retrieved from open3d: http://www.open3d.org/docs/latest/tutorial/Advanced/surface_reconstruction.html

Oreni, D., Brumana, R., Banfi, F., Bertola, L., Barazzetti, L., Cuca, B., & Roncoroni, F. (2014). Beyond crude 3D models: from point clouds to historical building information modeling via NURBS. *In Euro-Mediterranean Conference* (pp. 166-175). Cham: Springer.

Örmecioğlu, H., & Unay, A. (2006). Seismic strengthening of historical structures by enfolding steel skeleton. *The 5th International Conference on Behaviour of Steel Structures in Seismic Areas* (pp. 835-840). Yokohoma: Stessa. Retrieved

from https://avesis.akdeniz.edu.tr/yayin/d6d5ba2b-d14c-4d22-9b6d-81df89d4aa37/seismic-strengthening-of-historical-structures-by-enfolding-steel-skeleton

Oses, N., Dornaika, F., & Moujahid, A. (2014). Image-based delineation and classification of built heritage masonry. *Remote Sensing, 6(3)*, 1863-1889.

Othake, Y., Belyaev, A., & Seidal, H. P. (2005). 3D scattered data interpolation and approximation with multilevel compactly supported RBFs. *Graphical Models, 67(3)*, 150-165.

Pak, M., & Kim, S. (2017, August). A review of deep learning in image recognition. *In 2017 4th international conference on computer applications and information processing technology (CAIPT)*, 1-3.

paperswithcode. (2020). Retrieved from paperswithcode: https://paperswithcode.com/datasets

Pătrăucean, V., Armeni, I., Nahangi, M., Yeung, J., Brilakis, I., & Haas, C. (2015). State of research in automatic as-built modelling. *Advanced Engineering Informatics, 29(2)*, 162-171.

Peng, S., Jiang, C., Liao, Y., Niemeyer, M., Pollefeys, M., & Geiger, A. (2021). Shape as points: A differentiable poisson solver. *Advances in Neural Information Processing Systems*.

Pierdicca, R., Mameli, M., Malinverni, E. S., Paolanti, M., & Frontoni, E. (2019, June). Automatic generation of point cloud synthetic dataset for historical building representation. *In International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, 203-219.

Pietroni, N., Tarini, M., Sorkine, O., & Zorin, D. (2011, December). Global parametrization of range image sets. *In Proceedings of the 2011 SIGGRAPH Asia Conference*, (pp. 1-10).

Powers, D. M. (2015). What the F-measure doesn't measure: Features, Flaws, Fallacies and Fixes.

Pu, S., & Vosselman, G. (2009). Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing, 64(6),* 575-584.

Qi, C. R., Su, H., Mo, K., & Gubias, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 652-660. Retrieved from http://openaccess.thecvf.com/content_cvpr_2017/html/Qi_PointNet_Deep_Learning_CVPR_2017_paper.html

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*.

Riegler, G., Ulusoy, O. A., & Geiger, A. (2017). Octnet: Learning deep 3d representations at high resolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition* , 3577-3586.

Riveiro, B., Lourenço, P. B., Oliveira, D. V., González-Jorge, H., & Arias, P. (2016). Automatic morphologic analysis of quasi-periodic masonry walls from LiDAR. *Computer-Aided Civil and Infrastructure Engineering, 31(4),* 305-319.

Ronaghan, S. (2022). *Medium*. Retrieved from Medium.com: https://srnghn.medium.com/deep-learning-overview-of-neurons-and-activation-functions-1d98286cf1e4

Rumelhart, D. E., Durbin, R., Golden, R., & Chauvin, Y. (1995). Backpropagation: The basic theory. Backpropagation: Theory. *Architectures and Applications*, 1-34.

Sáez, A., Bergasa, L. M., López-Guillén, E., Romera, E., Tradacete, M., Gómez-Huélamo, C., & Del Egido, J. (2019). Real-time semantic segmentation for fisheye urban driving images based on ERFNet. *Sensors, 19(3)*(503).

Samson, H. (2022, June 02). *towardsdatascience*. Retrieved from towardsdatascience: https://towardsdatascience.com/getting-to-know-activation-functions-in-neural-networks-125405b67428

Saygı, G., & Remondino, F. (2013). Management of Architectural Heritage Information in BIM and GIS: State-of-the-art and Future Perspectives. *International Journal of Heritage in the Digital Era, 2(4)*, 695-713.

semanticscholar.org. (n.d.). Retrieved from semanticscholar.org: https://www.semanticscholar.org/paper/Learning-Features-from-Music-Audio-with-Deep-Belief-Hamel-Eck/86e951e190586b84c530f9f03504f9ad70cc650a

Seyyarer, A., & Aydın, T. (2017). Değişmez Momentler Kullanarak İçerik Tabanlı Görüntü Erişim Sistemi ve İmge Sınıflandırma Yöntemlerinin Karşılaştırılması. *Anatolian Science, 2(1)*, 1-9.

Seyyarer, E., Ayata, F., Uçkan, T., & Karcı, A. (2020). Derin Öğrenmede Kullanılan Optimizasyon Algoritmalarının Uygulanması ve Kıyaslanması. *Anatolian Journal of Computer Sciences, 5*(2), 90-98.

Shama, S. (2022, June 02). *towardsdatascience*. Retrieved from towardsdatascience.com: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Sharma, R. (2022, June 06). *Medium*. Retrieved from Medium: https://medium.com/nerd-for-tech/deep-learning-activation-functions-their-mathematical-implementation-b620d536d39b

Shen, Y., Feng, C., Yang, Y., & Tian, D. (2018). Mining point cloud local structures by kernel correlation and graph pooling. . *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 4548-4557.

Simonovsky, M., & Komodakis, N. (2017). Dynamic edge-conditioned filters in convolutional neural networks on graphs. *In Proceedings of the IEEE conference on computer vision and pattern recognition* , 3693-3702.

Sithole, G. (2008). Detection of bricks in a masonry wall. The International Archives of the Photogrammetry. *Remote Sensing and Spatial Information Sciences*, 1-6.

Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. *In Proceedings of the IEEE international conference on computer vision*, 945-953.

Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. *In Proceedings of the IEEE international conference on computer vision*, 945-953.

Suresh, A. (2022, June 02). *Medium*. Retrieved from Mediım: https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., & Rabinovich, A. (2015). Going deeper with convolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition* , (pp. 1-19).

Tang, P., Huber, D., Akıncı, B., & Lipman, R. (2010). Lytle Automatic reconstruction of as-built building information models from laser-scanned point clouds: a review of related techniques. *19*(7), 829-843.

Tang, P., Huber, D., Akinci, B., Lipman, R., & Lytle, A. (2010). Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Autom. Constr*, 829–843.

Tensorflow.org. (2022, June 02). Retrieved from Tensorflow: https://www.tensorflow.org/overview

Torlig, E. M., Alexiou, E., Foncesa, T. A., Querioz, R. L., & Ebrahimi, T. (2018, September). A novel methodology for quality assessment of voxelized point clouds. *In Applications of Digital Image Processing XLI*(10752), 107-520.

towardsdatascience. (2022). Retrieved from towardsdatascience: https://towardsdatascience.com/how-do-you-know-you-have-enough-training-data-ad9b1fd679ee

Unay, A. İ. (2002). Tarihi Yapıların Depreme Karşı Dayanımı. *ODTÜ*. Ankara.

Visin, F., Kastner, K., Cho, K., Matteucci, M., Caurvılle, A., & Bengio, Y. (2015). Renet: A recurrent neural network based alternative to convolutional networks.

Volk, R., Stengel, J., & Schultmann, F. (2014). Corrigendum to "Building Information Modeling (BIM) for existing buildings—Literature review and future needs"[Autom. Constr. 38 (March 2014) 109–127]. . *Automation in construction*, 204.

Vyas, K., Jiang, L., Liu, S., & Ostadabbas, S. (2021). An Efficient 3D Synthetic Model Generation Pipeline for Human Pose Data Augmentation. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* , 1542-1552.

Wang, C., Cho, Y. K., & Kim, C. (2015). Automatic BIM component extraction from point clouds of existing buildings for sustainability applications. *Automation in Construction, 56*, 1-13.

Wang, L., Huang, Y., Zhang, S., & Shan, J. (2019). Graph attention convolution for point cloud semantic segmentation. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10296-10305.

Wang, P. S., Liu, Y., Guo, Y. X., Sun, C. Y., & Tong, X. (2017). O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions On Graphics (TOG), 36(4)*, 1-11.

Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B., & Yuille, A. L. (2015). Joint object and part segmentation using deep learned potentials. *In Proceedings of the IEEE International Conference on Computer Vision* , 1573-1581.

Wang, Z., Liu, H., Qian, Y., & Xu, T. (2012, October). Real-time plane segmentation and obstacle detection of 3D point clouds for indoor scenes. *In European Conference on Computer Vision*, 22-31.

Wikipedia. (2020, June 02). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Convolutional_neural_network

Wikipedia. (2022, June 02). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Poisson_distribution#Assumptions:_When_is_the_Poisson_distribution_an_appropriate_model.3F

Wikipedia. (2022, June 02). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Softmax_function

Wikipedia. (2022, June 02). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Softmax_function

Williams, F., Schneider, T., Silva, C., Zorin, D., Bruna, J., & Panozzo, D. (2019). Deep geometric prior for surface reconstruction. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10130-10139.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., & Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. *In Proceedings of the IEEE conference on computer vision and pattern recognition* , 1912-1920.

Xie, Y., Tian, J., & Zhu, X. X. (2020). Linking points with labels in 3D: A review of point cloud semantic segmentation. *IEEE Geoscience and Remote Sensing Magazine, 8(4)*, 38-59.

Xiong, S., Zhang, J., Zheng, J., Chai, J., & Liu, L. (2014). Robust surface reconstruction via dictionary learning. *ACM Transactions on Graphics (TOG)*, 1-12.

Xiong, X., Adan, A., Akinci, B., & Huber, D. (2013). Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in construction, 31*, 325-337.

Xu, J., Ma, Y., He, S., & Zhu, J. (2019). 3D-GIoU: 3D generalized intersection over union for object detection in point cloud. *Sensors, 19(19)*, 4093.

Yi, L., Kim, V. G., Ceylan, D., Shen, I. C., Yan, M., Su, H., & Gubias, L. (2016). A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG), 35(6)*, 1-12.

Yi, L., Su, H., Guo, X., & Gubias, L. J. (2017). Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2282-2290.

Yılmaz, A., & Kaya, U. (2020). *Derin Öğrenme* (2 ed.). (G. Aksan, Ed.) İstanbul: KodLab. Retrieved July 2021

Yu, H., Yang, Z., Tan, L., Wang, Y., Sun, W., Sun, M., & Tang, Y. (2018). Methods and datasets on semantic segmentation: A review. *Neurocomputing, 82*(103), 304. Retrieved from https://www.sciencedirect.com/science/article/pii/S0925231218304077?casa_token=MPPXS_z3ERQAAAAA:T32kU_-KB-BeVDkI1aBP9UjoWWaTfdhQEG-6ohChfrZo_E08slDky7-BrL5sLNW5xpKSfAGT49I

Zhang, J., Zhao, X., Chen, Z., & Lu, Z. (2019). A review of deep learning-based semantic segmentation for point cloud. *IEEE Access*, 179118-179133.

Zhou, Y., & Tuzel, O. (2018). Voxelnet: End-to-end learning for point cloud based 3d object detection. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 4490-4499.

Zhu, H., Meng, F., Cai, J., & Lu, S. (2016). Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *Journal of Visual Communication and Image Representation*, 12-27.