

ANKARA YILDIRIM BEYAZIT UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES



3D FOUNTAIN MODELING FROM SINGLE IMAGE

M.Sc. Thesis by

Şeyma CENGİZ

Department of Computer Engineering

June, 2019

ANKARA

3D FOUNTAIN MODELING FROM SINGLE IMAGE

A Thesis Submitted to

The Graduate School of Natural and Applied Sciences of

Ankara Yıldırım Beyazıt University

**In Partial Fulfillment of the Requirements for the Degree of Master of Science
in Computer Engineering, Department of Computer Engineering**

by

Şeyma CENGİZ

June, 2019

ANKARA

M.Sc. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**3D FOUNTAIN MODELING FROM SINGLE IMAGE**” completed by **ŞEYMA CENGİZ** under the supervision of **ASST. PROF. DR. M. ABDULLAH BÜLBÜL** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. M. Abdullah BÜLBÜL

Supervisor

Prof. Dr. Tolga Kurtuluş ÇAPIN

Jury Member

Asst. Prof. Dr. Hilal KAYA

Jury Member

Prof. Dr. Ergün ERASLAN

Director

Graduate School of Natural and Applied Sciences

ETHICAL DECLARATION

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of Graduate School of Natural and Applied Sciences,

- All data, information and documents are obtained in the framework of academic and ethical rules,
- All information, documents and assessments are presented in accordance with scientific ethics and morals,
- All the materials that have been utilized are fully cited and referenced,
- No change has been made on the utilized materials,
- All the works presented are original,

and in any contrary case of above statements, I accept to renounce all my legal rights.

2019, 01 June

Şeyma CENGİZ

ACKNOWLEDGMENTS

I would like to express my gratitude to Asst. Dr. M. Abdullah BÜLBÜL who did not spare his interest and support in the planning, research, execution and formation of my thesis, and to Ebubekir BÜBER who never undermined the spiritual support that I benefited from his knowledge and experience during my study.

2019, 01 June

Şeyma CENGİZ



3D FOUNTAIN MODELING FROM SINGLE IMAGE

ABSTRACT

Transferring the objects that people use in their daily life to digital media has become a frequently needed process nowadays. One category of such applications are the transfer of historical buildings to digital environment by modeling in three-dimensions. Among the historical buildings, places of worship were very important in many cultures. Since the past, many glorious places of worship have existed until today. Mosques and fountains are important architectural structures in the Muslim culture. 3D modeling of these structures has become a concern of Cultural Heritage applications.

In general, 3D modelling of an object requires relative pose of the camera for the different views, relation between the image points, and the corresponding line of sight and image points. There are some difficulties when performing 3D modeling even with images taken from different angles. With the advances in image processing and artificial intelligence, we focused on research on the ways in which an object can be modeled in 3D from a single image.

Since the physical properties of each object are different from each other, it is necessary to produce a system specific to the object / structure to be modeled. In this thesis, a system is proposed to automatically generate 3D models of fountains from a single photograph. Although a single image is usually not sufficient for determining the 3D structure in an image; here we utilize our prior knowledge of general characteristic structures of fountains over the history such as the similarity of their top parts and their symmetrical shape. Firstly, the object is detected and labelled on the image using Convolutional Neural Networks (CNNs). Then, the object is segmented from background with Graph Cut technique and the contour of the object is used to estimate the shape of 3D model. Finally, 3D model of the fountain is completed by applying the texture acquired from the input image.

Keywords: Computer Graphics, 3D Modeling, Convolutional Neural Networks, Deep Learning, Graph Cut, Cultural Heritage, Fountain.

TEK FOTOĞRAFTAN 3 BOYUTLU ŞADIRVAN MODELLEME

ÖZ

İnsanların günlük yaşamlarında kullandıkları objeleri dijital ortama aktarmak günümüzde ihtiyaç duyulan bir işlem haline gelmiştir. Bu uygulamalara örnek olarak, tarihi yapıların üç boyutlu modellenerek dijital ortama aktarılması verilebilir. İbadethaneler, tarihi yapılar arasında pek çok kültürde ibadethaneler önemli yer edinen yapıların arasında gelmektedir. Geçmişten günümüze dek tüm kültürlerde birçok ihtişamlı ibadethane inşa edilmiştir. Camiler ve şadırvanlar, Müslüman kültürünün önemli mimari yapılarıdır. Bu yapıların üç boyutlu modellenmesi günümüzde ihtiyaç duyulan bir problem haline gelmiştir.

Genel olarak, bir nesnenin 3 boyutlu olarak modellenebilmesi için, farklı açılardan çekilmiş görüntülere ihtiyaç duyulabilmektedir. Bu görüntülerin arasındaki ilişki, görüş açısı ve kamera pozunu 3 boyutlu modelleme yapılması için analiz edilmesi gereken önemli parametrelerdir. Farklı açılardan çekilmiş görüntülerde bile 3 boyutlu modelleme yaparken birtakım zorluklar vardır. Görüntü işleme ve yapay zekadaki ilerlemelerle, bir nesnenin tek bir görüntüden 3 boyutlu modellenebilmesi üzerine çalışmalar hız kazanmıştır.

Her nesnenin fiziksel özellikleri birbirinden farklı olduğu için, modellenecek nesneye / yapıya özgü bir sistem üretmek performans açısından gereklidir. Bu tezde, tek bir fotoğraftan otomatik olarak 3 boyutlu şadırvan modellerinin çıkarılabilmesi amacıyla bir sistem geliştirilmiştir. Bir görüntüdeki 3 boyutlu yapının belirlenmesi için tek bir görüntü yetersiz kalabilmektedir. Bu çalışmada, yapıların benzerliği ve simetrik şekilleri gibi karakteristik özelliklerinden faydalanılarak tek bir görüntüden 3 boyutlu modelleme yapılması hedeflenmiştir. 3 boyutlu modelleme işlemi için ilk olarak, Evrişimsel Sinir Ağları (CNN'ler) kullanılarak görüntü üzerinde nesne tespit edilir ve etiketlenir. Daha sonra, nesne Grafik Kesim (GrabCut) tekniği ile arka plandan ayrılır ve nesnenin dağılımı, 3B modelin şeklini tahmin etmek için kullanılır. Son olarak, girdi görüntüsünden elde edilen doku uygulanarak çeşmenin 3 boyutlu modeli tamamlanır.

Anahtar Kelimeler: Bilgisayar Grafikleri, 3B Modelleme, Evrişimsel Sinir Ağları, Derin Öğrenme, Grafik Kesim, Kültürel Miras, Şadırvan.



CONTENTS

M.Sc. THESIS EXAMINATION RESULT FORM	ii
ETHICAL DECLARATION	iii
ACKNOWLEDGMENTS	iv
ABSTRACT	v
ÖZ.....	vi
NOMENCLATURE	x
LIST OF TABLES	xi
LIST OF FIGURES	xii
CHAPTER 1 - INTRODUCTION	1
1.1 Scope of the Work	3
1.2 Contributions	4
1.3 Thesis Outline.....	4
CHAPTER 2 - FOUNTAIN	6
2.1 According to Plan Features.....	9
2.1.1 Polygon Planned Fountains	9
2.1.2 Circular Planned Fountains.....	9
2.2 Material and Technical Details	10
2.3 Structural Elements	10
2.4 Ornament	11
CHAPTER 3 - BACKGROUND	12
3.1 Related Works.....	12
3.2 Deep Learning	15
3.2.1 Convolutional Neural Networks (CNN).....	17
3.2.2 Transfer Learning.....	21
3.2.3 YOLO	22
3.3 Object Detection.....	28
3.3.1 Pre-processing.....	30
3.3.2 Object Detection with Deep Learning	31
3.4 Object Segmentation	32
3.5 3D Modeling.....	33
3.5.1 Models	33

3.5.2 Type of Modeling	34
3.5.3 Modeling Process.....	35
3.5.4 Modeling Techniques.....	36
3.6 Locating the camera at perspective location	36
CHAPTER 4 - IMPLEMENTATION	39
4.1 Overview	39
4.2 Fountain Detection	39
4.3 Fountain Segmentation.....	41
4.4 3D Fountain Modeling	43
4.5 Locating the camera at perspective location	48
4.5.1 Vanishing Point.....	49
CHAPTER 5 - RESULTS & DISCUSSION	52
REFERENCES	57
CURRICULUM VITAE	65

NOMENCLATURE

Acronyms

2D	Two Dimension
3D	Three Dimension
ANN	Artificial Neural Network
BRNN	Bi-Directional Recursive Neural Network
CNN	Convolutional Neural Networks
CUDA	Compute Unified Device Architecture
FFNN	Feed Forward Neural Network
GPU	Graphical Processing Unit
HOG	Histogram of Oriented Gradient
NLP	Natural Language Processing
NURBS	Non-Uniform Rational B-Splines
RANSAC	Random Sample Consensus
RCNN	Convolutional Recursive Neural Network
RGB	Red-Green-Blue
RNN	Recurrent Neural Networks
ROI	Region of Interest
SfM	Structure from Motion
SIFT	Scale-Invariant Feature Transform
VR	Virtual Reality

LIST OF TABLES

Table 4.1 Parameters of A Model.....	46
---	----



LIST OF FIGURES

Figure 1.1 Fountain Examples from Şehzadebaşı Mosque, Fatih Mosque, Konya Mevlana, and Hagia Sophia. Images are Courtesy of Mustafa Cambaz [2].	1
Figure 2.1 First Fountain of Mosul Ulu Mosque	7
Figure 2.2 First Ottoman Fountain of Fatih Mosque	7
Figure 2.3 Diyarbakır Behram Paşa Mosque Fountain [76]	9
Figure 2.4 Kastamonu Topcuoğlu Fountain [77]	10
Figure 3.1 An Example Artificial Neural Network.....	16
Figure 3.2 The Relationship Between Size of Data and Neural Network.....	16
Figure 3.3 Some Customized Deep Learning Approaches	17
Figure 3.4 CNN Architecture [22]	18
Figure 3.5 Two-Dimensional Convolution Process Representation	19
Figure 3.6 Horizontal and Vertical Filtering Process [74]	19
Figure 3.7 Convolutional Neural Networks	20
Figure 3.8 Illustration of Stride	20
Figure 3.9 A Simplified Illustration of the YOLO Object Detector Pipeline [75] ...	22
Figure 3.10 Boundary Box Example 1	23
Figure 3.11 Boundary Box Example 2.....	24
Figure 3.12 YOLO Predictions on 20 Classes [69].....	25
Figure 3.13 A sample Tensor	26
Figure 3.14 YOLOv3 Performs Very Well in the Fast Detector Category When Speed is Important. [70]	28
Figure 3.15 Object Detection and Recognition Stages	29
Figure 3.16 A Raw Input.....	29
Figure 3.17 An Example Detected Fountain.....	30
Figure 3.18 The Steps of Image Pre-Processing Stage	30
Figure 3.19 Left: Noisy Photo Right: After the Pre-process [78]	31
Figure 3.20 Side View [73]	36
Figure 3.21 LookAt (eye, at, up) [73]	37
Figure 3.22 Illustration of Vanishing Points, Eye Level, Parallel Lines [72]	38
Figure 4.1 Overview of the System.....	39
Figure 4.2 Prediction Images with the Bounding Box of the Detected Objects.	40
Figure 4.3 Diagram of the GrabCut [26].....	41

Figure 4.4 Prediction Image [27]	42
Figure 4.5 Segmented Object using GrabCut.....	43
Figure 4.6 Segmented Image.....	44
Figure 4.7 Foreground Image by Flood Filling.....	45
Figure 4.8 Extracted Contour of Fountain	45
Figure 4.9 Determining the Radii of Model for Each Height	46
Figure 4.10 Generating 3D Model with Radii for Each Height	47
Figure 4.11 3D model.....	47
Figure 4.12 Textured 3D model.	48
Figure 4.13 Output of Shifted Texture Coordinate of the Sample 3D Fountain Model	49
Figure 4.14 Finding Possible Vanishing Point of Sample Fountains with Rectangle Box 1	50
Figure 4.15 Finding Possible Vanishing Point of Sample Fountains with Rectangle Box 2.....	51
Figure 4.16 Point of Intersection of Lines Passing Through the Edges of the Fountain.1	51
Figure 4.17 Point of Intersection of Lines Passing Through the Edges of the Fountain.2	51
Figure 5.1 Results. Left: Input Images, Middle: Operations to Generate 3D Model, Right: Outputs	53
Figure 5.2 Classifications of Fountains.....	54
Figure 5.3 Detection of Fountain with Dome Roof 1	55
Figure 5.4 Detection of Fountain with Dome Roof 2	55

CHAPTER 1

INTRODUCTION

Architectural heritage is one of the most important components of cultural heritage. Regarding the number and diversity of cultural heritage components, Turkey is a distinguished country at a universal level. Ablution fountains (*şadırvan* in Turkish) form an important class of architectural heritage and they have a significant place in Turkish-Islamic architecture. Through the history, various styles of fountains are built representing different eras. Figure 1.1 shows several examples of Fountains.



Figure 1.1 Fountain Examples from Şehzadebaşı Mosque, Fatih Mosque, Konya Mevlana, and Hagia Sophia. Images are Courtesy of Mustafa Cambaz [1]

Fountains are the structures that people use to get prepared for worshipping. They are usually found inside or near a mosque. A Muslim who wants to worship, usually stops by the fountain and makes necessary preparations for worship. Then he enters the mosque and worship. Therefore, since the fountains are also seen as the areas

frequently used in daily life, these structures have continued to take place in Muslim cultures since the past.

Several application areas such as cultural heritage, computer games, and simulations require these structures to be transferred into virtual environments. 3D modeling is very time consuming and costly when done manually. Therefore, based on the visual features of these structures, automatic modeling by taking advantage of image processing approaches has become an area of increasing importance. In this study, a system has been developed for fully-automatic modeling of fountains by using deep learning methods and 3D modeling techniques.

Technological advances in the ability to measure the physical world and in computer modeling capabilities have accelerated the creation of 3D models of cultural heritage objects and environments. Generation of 3D models of these heritage elements on digital environment will contribute to the knowledge transfer to the generations;

- to the easier understanding of the architectural design
- to have information about the history of the structure
- to determine the current status of the structure
- to be able to identify various kinds of problems in a historical building or a group of buildings
- to ensure that the structure or group of structures is evaluated in today's contemporary use
- to obtain the sensitive data required for future conservation, restoration and restitution plans
- to have detailed information about the structure or the group of structures, and
- to obtain the necessary data to prepare the survey projects which constitute a basis for conservation projects

3D modelling of an object is a process that begins with the acquisition of data and ends with the creation of the 3D virtual model. 3D models are used in many areas such as visualization, animation, restoration, protection, VR applications, catalogs,

web geographic systems and visualization, and in recent years it has become an important and fundamental step especially in the field of archiving cultural heritage in digital environment [2-4]. Main consideration in this area is high geometric accuracy, i.e., how close the results are to the actual model and whether all details are modeled or not, as well as being low cost, portable and flexible.

As the number and variety of 3D polygonal models increases in online repositories, there is a growing need for automatic algorithms that can obtain structural and semantic relationships from large model collections [5] which led researchers to increase their efforts on 3D modeling and reconstruction. In general, reconstructing a 3D structure requires multiple views of the scene so that the coordinates of the corresponding features in different views are solved in combination which is a challenging task; in case of presence of only a single image; however, generating a 3D model becomes even more challenging. In that case, prior knowledge about the objects in the image helps revealing the 3D structure. Recent improvements in Computer Vision enable robust object detection, which makes it possible to utilize our prior knowledge of objects in an image.

1.1 Scope of the Work

In this study, a system has been developed for the 3D modeling of the fountains which have historical and cultural importance. Given a single photograph, the developed system automatically generates the 3D model of the fountain present in the image by using deep learning-based image processing methods and 3D modeling tools.

The extraction of three-dimensional objects from a single photograph is a challenging process considering the current state of the technology. The target object must be segmented from its background as it contains numerous complex tasks and should be recognized from its 3D exposure, shape and structure projection.

Fountains have certain properties depending on the architectural style and the time period they are built in. These properties can be exploited to detect fountains and generate their 3D models. Generally, fountains have a polygonal base, a pointed

dome, pool, taps on the sides of the pool and columns on side surfaces. We developed a system that generate 3D model of detected fountain objects using this information. Thus, our system contributes documenting cultural heritage by providing 3D models of historical artifacts for which we have only a single photograph.

1.2 Contributions

In this study, a system is developed to create 3D model for fountains from a single image. There are studies in the literature on the 3D modeling of objects, to the best of our knowledge; there is no study on the 3D modeling of fountains in the literature. The developed system is a first in this field.

In order to create a 3D model of the object in the image, the object must be detected firstly. Object detection is an important step for modeling objects in the image. The low success rate in this step causes the performance of the whole system to decrease. In this study, YOLO architecture, which is quite popular in recent years, has been used for detection of fountains. YOLO is the name given to a special architecture that is developed for object detection with a deep learning approach. In the default version of Yolo 2000+ objects can be detected. However, the fountain is not one of them. In order to determine the fountains before the modeling step, pre-trained model was used on YOLO architecture and more training were made to include the fountain between the recognized objects. By using the data collected for fountains, YOLO algorithm was trained to recognize fountains.

After the determination of the fountain, 3D modeling was done by using the symmetrical properties of the fountain.

1.3 Thesis Outline

In this study, detailed explanations of all steps performed for 3D modeling of fountains are given. In Chapter 2, technical information about the fountains, In Chapter 3, information on 3D modeling studies in the literature is given. In addition,

information about the deep learning approaches used in this study and information about 3D Modeling are given in Chapter 3.

Technical details of the system developed in this study are explained in Chapter 4. Information was given about project implementation in this chapter. The deep learning approach used in the detection of fountains and information about the trainings to determine the fountains were given in Chapter 4. In Chapter 5, information on the evaluation of the results, the limits and future studies are given. Finally, the results are discussed.



CHAPTER 2

FOUNTAIN

One of the facilities belonging to water in Turkish architecture is the fountains. The fountains are high-edged pools located at the inner or outer courtyards of religious and social buildings such as mosques, madrasas, khans, caravansaries [6]. These were made for the public to take ablution and drink water. The origin of the word Fountain (*Şadırvan*) is Persian. It was formed by combining the words *shad* that means much and *revan* that means flow [7]. It is stated that the fountains were built for reasons such as religious belief, the motive of charity, gathering and resting of the communities, giving an elegant appearance to the courtyard or square where it is located, and cooling the place where it is located [8].

The first fountain in 542 Mosul Ulu Mosque on Figure 2.1 was built in the courtyard. [9] This building, which was built with marble stones, was built by I. Seyfeddin Gazi, who was living in the Seljuk period. The first purpose of the fountain is to add beauty to the mosques. Fountain architecture have evolved to its most mature and final shape in the Ottoman era. Moreover, the fountains built by the Ottomans were not only built as an ornamental pond but were built with the ablution feature. The fountains, which later became a work spreading all over the Islamic world, show different characteristics according to their culture and traditional architecture. The first fountain in the Ottoman architecture is the fountain in the Fatih Mosque courtyard in Istanbul in 875. The first fountain built with Ottoman architecture, the walls of this building are designed as octagonal and the dome carries eight columns.

The oldest known fountain in Anatolia is an octagonal planned fountain in Harran Ulu Mosque.



Figure 2.1 First Fountain of Mosul Ulu Mosque [10]



Figure 2.2 First Ottoman Fountain of Fatih Mosque [1]

Among the fountains in the mosque courtyards, Fatih Mosque (1470) on Figure 2.2, Suleymaniye Mosque (1557), Edirne Selimiye Mosque (1575) fountains, Istanbul Ayasofya Mosque Fountain (1740) are among the best examples of Ottoman fountains. Gedik in her master thesis titled “İzmir Su Yapıları” [11] discussed the fountains extensively and the Ottoman classical fountain architecture was divided into four groups.

Ablution fountains made of various shapes or their reservoirs or sprinkler hubs. Some of examples are;

- **Closed reservoir tankers:** they are fountains formed by covering the top of the chambers. Stone fountains placed on a pedestal, such as water, filled with water, are examples of these fountains.
- **Bowl Edges with Marble Cage and Crown Fountains:** it is formed by placing cage and crown which are formed by panels.
- **Fountains with Waterer:** some fountains, which are surrounded by cages, have bowls consisting of cantilevered mirrors with carved marble.
- **Pavilion Masjid Fountains:** the huts are raised from the ground by arches, which are found on the hills, which are based on the foot or pillars, and the pavilions are located under the masjid.

Ablution fountains carried with double support sequence of the top cover: The second support line was added to the legs and pillars carrying the top cover, and a larger fringed top cover were obtained.

Water Distribution and Ventilation Facilities in the Domestic courtyards of the Mosques: These fountains built as closed fountains were constructed as distribution and ventilation facilities.

2.1 According to Plan Features

2.1.1 Polygon Planned Fountains

Examples of polygonal fountains; Ankara Hacı Bayram Mosque Fountain (1427) [12], Kastamonu Taşköprü Abdal Hasan Zaviyesi Fountain (15 th century) [13], Diyarbakır Behram Paşa Mosque Fountain on Figure 2.3 (1572) [14], Istanbul Sultan Ahmet Mosque Fountain (1617) [15], Kocaeli Yukarı Halidere Fountain (17th century) [16].



Figure 2.3 Diyarbakır Behram Paşa Mosque Fountain [17]

2.1.2 Circular Planned Fountains

Some of the circular schemes made in Anatolia and Istanbul are as follows; Konya Dergah Şadırvanı (1512) [18], Kastamonu Topcuoğlu Fountain on Figure 2.4 (1727) [19], Istanbul Ayasofya Mosque Fountain (1739) [20], Aydın Cihanzade Mosque Fountain (1756) [21].



Figure 2.4 Kastamonu Topcuoğlu Fountain [22]

2.2 Material and Technical Details

The material used in the fountain architecture is marble and stone. Stone and wood materials were used in the top covers of the fountains and some fountains were covered with concrete. Şadırvanaltı Mosque, Lütfü Paşa Mosque, Başdurak Mosque, Şadırvanlı Mosque, Ali Pasha, Hacı Turan Kapan Mosque, Tire Yeni Mosque fountains were closed with dome made of stone. Rum Mehmet Pasha Mosque and Bergama Yeni Mosque fountains are covered with wooden domes. Şeyh Mosque and Yalınayak Mosque fountains are the top cover material concrete.

2.3 Structural Elements

The fountains are small structures with dimensions and architectural monuments formed by the merging of some parts. Water channel, seating areas (seats), water reservoir, faucet, faucet stone, details on the reservoir, core shell, carrier system are the main and secondary elements of the fountain [23].

- **Water Channel:** Circulating water flowing from the tap. Water channels are circular or polygonal.
- **Seating Place:** There are places to sit around the fountains and not all of the fountains. Some fountains do not have seating places; some fountains are made of marble, stone or wood.
- **Water reservoir:** The water reservoir where the water is deposited is the most important part of the fountain. After the principalities in Anatolia, the vessel edges were raised, and taps were fitted [24]. The water reservoirs are made of marble or stone and have a circular or polygonal plan.
- **Sink and Tap:** The external surface of the water reservoirs is surrounded by a tap stone composed of stone or marble panels. The faucet was fitted with taps inside the ornament. It is understood that fountains have been used in the middle of the 16th century [25].
- **Details on the Hopper:** Grids are placed on the sides of the hopper and onto the grids in order to prevent the water in the reservoir from getting dirty.
- **Belly Button:** There are small holes on the bowls in the middle of the Water Reservoir where the water is filled.
- **Carrier System and Upper Cover:** Over the fountains, the ablution or cooling for those who come with the dome or roof is closed. The first examples of fountains were covered in the 15th century [26].

2.4 Ornament

There are many beautiful motifs and details in the fountains that keep Turkish art alive [27]. The fountains of Turkish art are elegant examples of stone and marble workmanship. The ornamental, geometric, object-oriented and embroidered decoration is seen in the dome that covers the water tank of the fountains on the outside, the dome covering the roof and the arch at the column.

CHAPTER 3

BACKGROUND

3.1 Related Works

Studies specific to fountain modelling do not exist in the literature. However, there are many studies dealing with 3D modeling from 2D images. Fountain modeling is based on similar foundations with these studies. In this part, 3D modeling studies using 2D visuals are investigated and the methods learned were used to develop a system capable of modeling fountains.

Based on 2D images, 3D modeling is an application area that can be used in various application areas. Some examples of application areas are; creating simulation platform or mapping a surface.

Many studies in literature have different approaches for 3D modeling of objects from images. For example, Mao and Xu developed a system [28] to detect drinking cups from single photo and reconstruct the target objects in 3D. There are three steps in that study. First step is detection part to find location of the objects. Histogram of Oriented Gradient (HOG) method as template feature matching method with sliding windows is used to detect the target object. Second step is 3D reconstruction part to find the optimal 3D cup model for detected target object. They matched the edge maps of the target objects with parameters that specify shape and coordinates of cup. A cup model has 4 faces as outside wall, inside wall, outside bottom and inside bottom. Then, they project models on the background image and optimize their positions. Third step interactive image editing part as in [29] to adjust the parameters of the cup model.

Olmschenk et al [30] working on the corridor model from a single photo. Formed the 3D skeleton of the corridor from the intersections of parallel lines passing from the floor. The perspective and geometry of a corridor are used to extract important features of the picture. Guided robots in the environment, visually impaired people to

find doors and cross section corridors, providing a hallway detection that prevents the walls have developed. In their work, they made more than one assumption of camera knowledge, approached the perspective of the corridor, approached a perspective and focused on the issue of real time robotics and mobile robots on smartphones. In order to form the shape of the corridor, noise reduction and filter were applied, and vanishing points were calculated by canny edge detection method. Hough transform method was used with the help of tilt angle and heading to get surface feature lines.

In sweep-based modeling [31], researchers developed an interactive system to model and manipulate 3D man-made object from a single image. These methods, called 3-sweep, provide the user to explicitly define the three dimensions of cylinder, cuboid or similar primitives using three sweeps. First and second sweeps are used to define the first and second dimension of a 2D profile. Third sweep is used to define the main curved axis of the primitive. This method cannot be used for modeling highly complex objects.

A group of researchers [32-34] used part-based modeling for modeling objects having certain parts. Here, researches produce a part-based template algorithm that groups original models in clusters of models with their variations. The deformable templates are used to describe shape variations within a collection as in [35]. Each template represents a distribution of shapes with different geometric features. This algorithm is evaluated with different datasets such as collections of chairs. While this algorithm achieves higher accuracy for boxy parts, it is not very successful for shapes with complex parts.

In category-specific modeling [36, 37], researchers developed an algorithm that specifies category of models from images. The automatic object segmentation method which is 2D annotations present in computer vision datasets such as PASCAL VOC [38] is used to handle the objects from images. In their system, firstly, the objects are detected and segmented from image. Then, viewpoints and subcategories are predicted using a CNN based system. After that, the system learns

the 3D shape of the model at canonical bounding box scale. The mean shape learned from the predicted subcategory is scaled by given the predicted bounding box.

Yan et al. [39] propose an algorithm to detect the object given an arbitrary 2D view using a general 3D feature model of the class. The motivation of this study is an efficient object detection system from the same class under different viewing conditions. 3D modeling of the object is considered to obtain better accuracy. In the feature modeling phase, SIFT detector is used to compute the features of 2D model views. The features computed in 2D images are mapped to the 3D model by using homographic framework.

With general 3D modeling approach in [40], a method is presented for grasping from a single view of a 3D sensor. There are two approaches to make the model as grasp planning and locating objects in camera images using the information given by the geometric clues. The model is developed for fast estimation of different kinds of symmetries and unknown objects. In this system, firstly, data is cleaned from noise. Next, a region of interest (ROI) is applied to detect the object using a variant of the RANSAC algorithm. After, estimation is made over the remaining data a contour is fitted. Finally, all fitted models are triangulated and triangulation results are merged. Rothganger et al. [41] models objects in 3D in terms of local affine invariant descriptors of photos and video views. Geometric constraints occur due to different views of the same patches under affine projection. These constraints are combined with a normalized representation of their appearance. The study of Jiang et al. [42], builds symmetric architecture from single image, and the aim of this study is similar to ours; however, in this study user interaction is required while our approach is fully automatic.

There are also approaches based on structure from motion (SfM) that require a vast number of views of the modeled scene [43-45]. These approaches find visual features, e.g., SIFT features, in input images and match corresponding features among views. Then these correspondences are used to extract camera locations of the input images as well as 3D positions of the common features among images resulting in a sparse point cloud of the scene. The advantage of these systems is their

capability of generating complex scenes as long as the scene doesn't contain highly reflective and repetitive surfaces. On the other hand, for a robust reconstruction, a notable number of input images (<50) are required and the result is a sparse point cloud which is usually densified by other methods such as [46] before surface reconstruction.

3.2 Deep Learning

Deep learning is a subclass of machine learning [47]. Deep learning uses many nonlinear processing unit layers for feature extraction and conversion. Each successive layer takes the output from the previous layer as input [48]. Algorithms can be supervised (such as classification) or unsupervised (like pattern analysis).

In deep learning, there is a structure based on the learning of multiple property levels or representations of data. High-level features are derived from low-level features to create a hierarchical representation. This representation learns multiple levels of representation corresponding to different levels of abstraction [49]. Deep learning is basically based on learning from the representation of the data. When it comes to representation for an image; Features such as a vector or edge sets of density values per pixel, special shapes can be considered. Some of these features represent data better. As an advantage at this stage, deep learning methods use effective algorithms to extract hierarchical properties that best represent data rather than handcrafted features [50].

Deep learning approaches based on Artificial Neural Networks are composed of multiple layers and neurons. The first developed architectures are frequently referred to in the literature as the Advanced Feed Artificial Neural Network (AFANN). In this network structure, neurons are linked. A neuron may have many inputs, but each neuron may have one output. The network structure designed in Figure 3.1 is modeled to solve the problem of house price estimation. Therefore, input information is related to this problem.

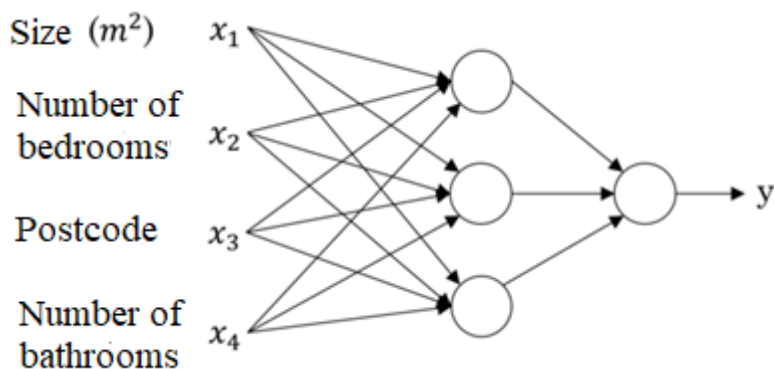


Figure 3.1 An Example Artificial Neural Network

The learning process in the AFANN is provided by learning the weights between the nodes. Each data instance causes some update with inter-node weights. With the use of a large number of data, the correct parameters for problem definition can be learned. Deep learning approaches require large amounts of data to achieve high success. Increasing the number of layers gives deep learning approaches the ability to model more complex problems. It is effective in increasing the data performance that increases in proportion to the increasing number of layers for complex problems. The relationship between the dimensions of the neural network that can be used in complex problems with increasing data is given in Figure 3.2.

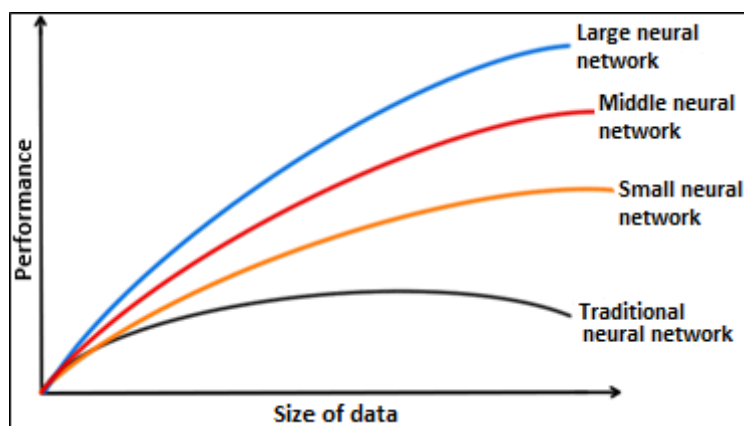


Figure 3.2 The Relationship Between Size of Data and Neural Network

Deepening the neural network for complex problems can lead to successful results. However, according to the structure of the problem solved, shallow architectures consisting of fewer layers can also yield successful results.

In addition to traditional deep learning approaches, different deep learning architectures have been developed to solve some problem types more successfully. For example, Convolutional Neural Network structures for some image processing problems produce very successful results. For some problems with DDI, it has been proposed approaches such as Recurrent Neural Networks (RNN), Bi-Directional Recursive Neural Network (BRNN), and Convolutional Recursive Neural Network (RCNN). An example of the different deep learning approaches developed is given in Figure 3.3

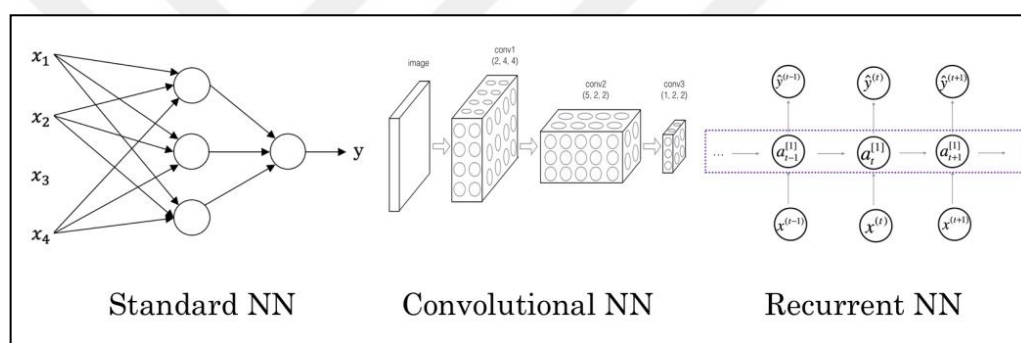


Figure 3.3 Some Customized Deep Learning Approaches [51]

3.2.1 Convolutional Neural Networks (CNN)

Nowadays, CNNs are very popular in deep learning that recently have proven to be very successful at image recognition. CNNs are inspired by the visual cortex of the human brain and they are a special case of Artificial Neural Network (ANN). CNNs include feature extraction function before the normal feed-forward neural network. Computational process of operations is slow because they involve intensive processing. GPU programming with CUDA is used to speed up operations in training and testing phases. CNNs architecture is shown in Figure 3.4. The input image passes to the first convolutional layer. The filters applied in the convolution layer extract relevant features from the input image to pass further. Pooling layers are then

added to further reduce the number of parameters. Several convolution and pooling layers are added before the prediction is made.

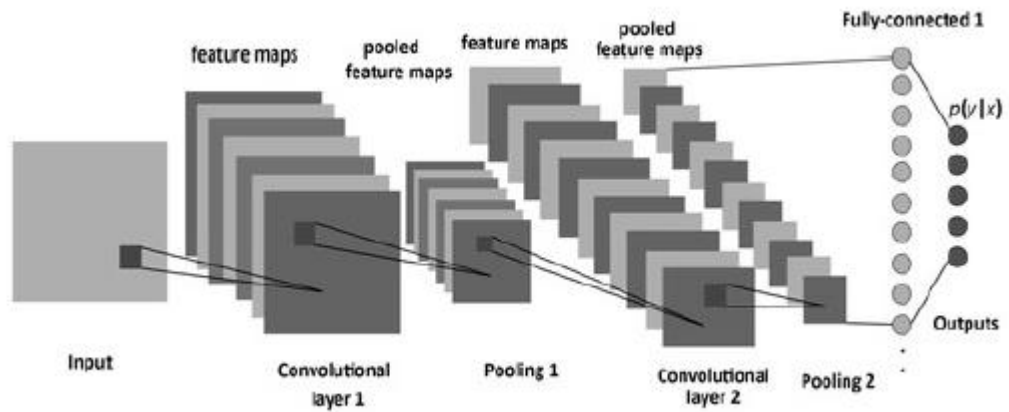


Figure 3.4 CNN Architecture [52]

In the two-dimensional convolutional neural network, the symmetry of the filter to be applied to two-dimensional information according to x and y axis is taken. All values are multiplied by the element in the matrix and the sum of all values is recorded as the corresponding element of the output matrix as shown in Figure 3.5. This is also referred to as cross-correlation. This can be done simply when the input data is single channel. However, the input data may be in different formats and number of channels.

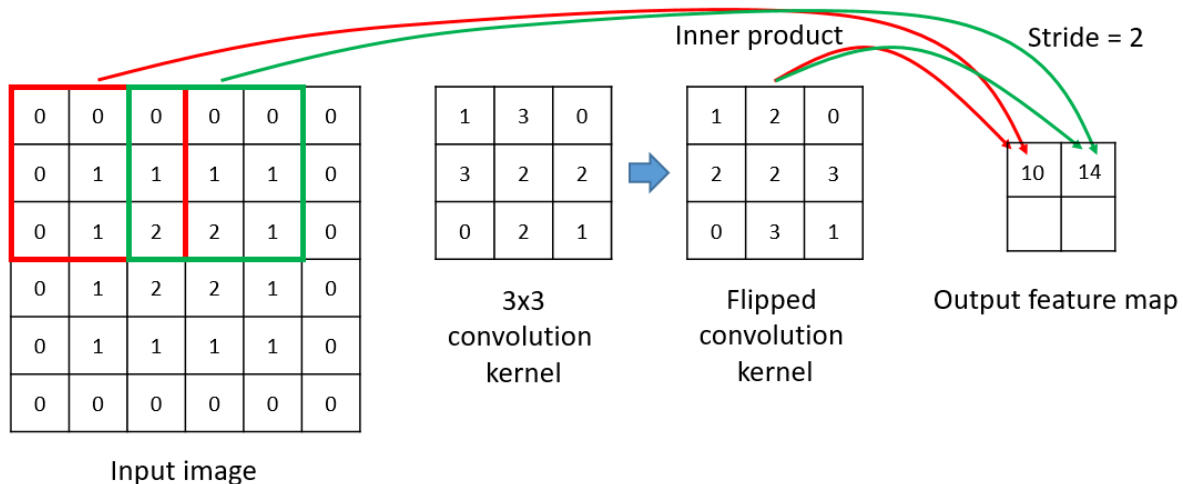


Figure 3.5 Two-Dimensional Convolution Process Representations [53]

Color images consist of three Red-Green-Blue (RGB) channels. In this case, convolution is done for three channels. The number of channels of the output signal is also calculated equal to the applied filter channel/number (see Figure 3.6)

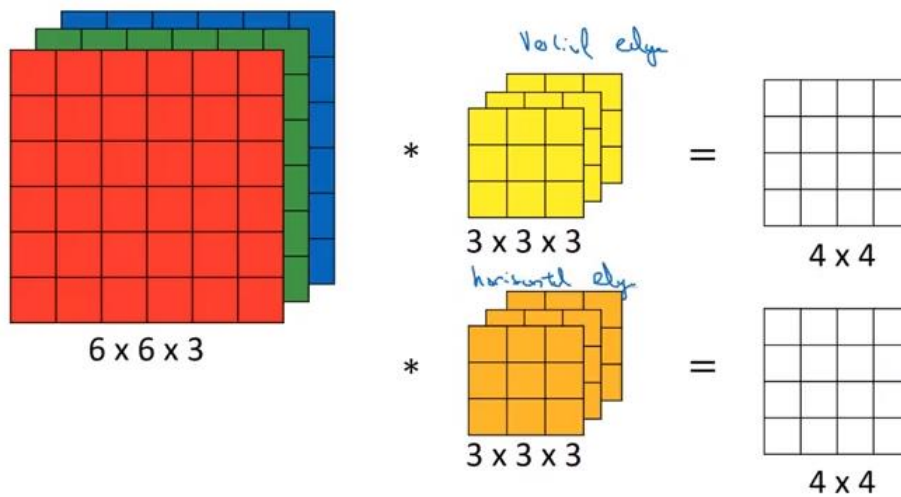


Figure 3.6 Horizontal and Vertical Filtering Process [51]

Let's imagine this calculation as a layer in the neural network. The input image and the filter are in fact the matrix of weights updated with continuous back-spread. The

last scalar b (bias) value is added to the output matrix applied to the activation function. You can examine the visual convolution process flow in Figure 3.7.

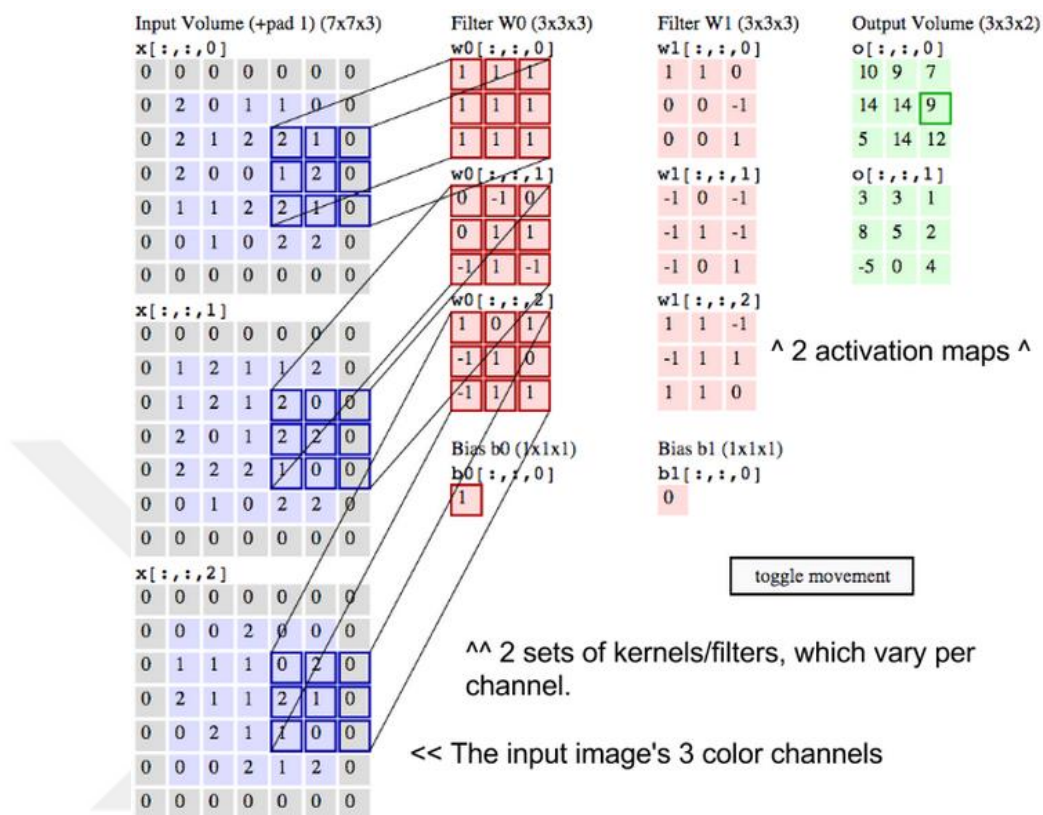


Figure 3.7 Convolutional Neural Networks [53]

If calculated for $n = 5$ and $f = 3$, the output size is $(3) \times (3)$. The pixels added in the padding process can be made up of 0 as in the example in Figure 3.8. Another mode of application is to copy the value of the next pixel.

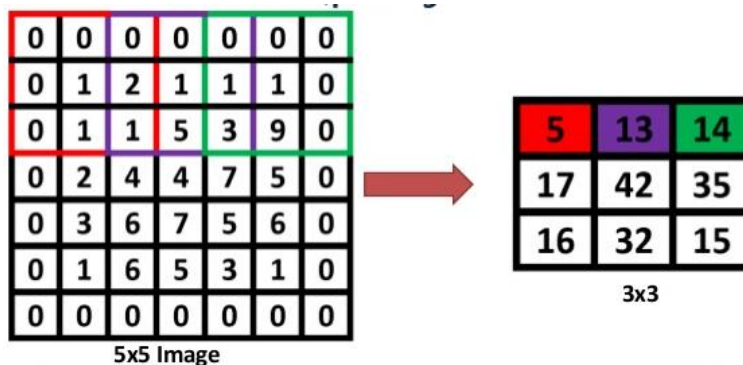


Figure 3.8 Illustration of Stride [54]

3.2.2 Transfer Learning

In some types of problems, there is a need for a lot of processing power too many system resources to train deep learning architectures. The process power and the time required for the training phase increase in proportion to the problem complexity. Therefore, researchers working on problems requiring high processing power and time benefit from Transfer Learning.

In this way, they can build great deep learning architectures with less effort. Transfer learning is the use of previously learned parameters to solve a problem. For example, if you are working on an image processing problem for object recognition, the large number of objects you want to detect is a very important criterion for the system. However, laboratories with highly powerful computers may be needed to train systems that recognize thousands of objects. In such a case, we can have a system capable of recognizing thousands of objects by reading only the models and parameters that have been trained and maintained for a long time by computers with very good system resources. With the transfer of learning, researchers can transfer their experiences to each other. We can start using that system by reading only the learned parameters of a model that can recognize thousands of objects. We can also continue to develop on this system. It is sufficient to train some target object to the trained model to include an object of our own among the recognized objects. Transfer learning allows many researchers to save time and system resources in many applications. In this way, researches will progress cumulatively, and step-by-step systems will be developed more successfully. For example; this is advantageous when the data set you use is not large enough. If you have a model with a set of 15 million different images, such as ImageNet, if you have a very few images, you will have more successful results as the learning process is performed. How many different ways to learn how to transfer?

- Freeze the whole model (trainable parameter = 0, freeze = 1) by adjusting the softmax output to the class number of your own problem,

- Keeping several layers of the model constant and keeping the last layers different,
- The entire network can be used for training in your own data set (learnable parameter = 1, freeze = 0)

3.2.3 YOLO

With Evolutionary Deep Networks, there are rapid developments in object recognition. Art vision was already the first field of "learning" of deep learning, the best error rate in the so many problems can be solved with deep learning approaches with high success rate.

New and very fast deep learning-based object detection approach is YOLO. The abbreviation is derived from the phrase "You Only Look at Once", which means that people can only quickly find the objects in a single frame at a glance. The aim of YOLO is to achieve this speed.

According to YOLO designers, it is not necessary to scan the image frame to the left (to look at the pixels that fall into a small box at that time), or to produce quotations about the object location (R-CNN). A global decision can be made about the entire square and all the object types and places in that frame like on Figure 3.9.

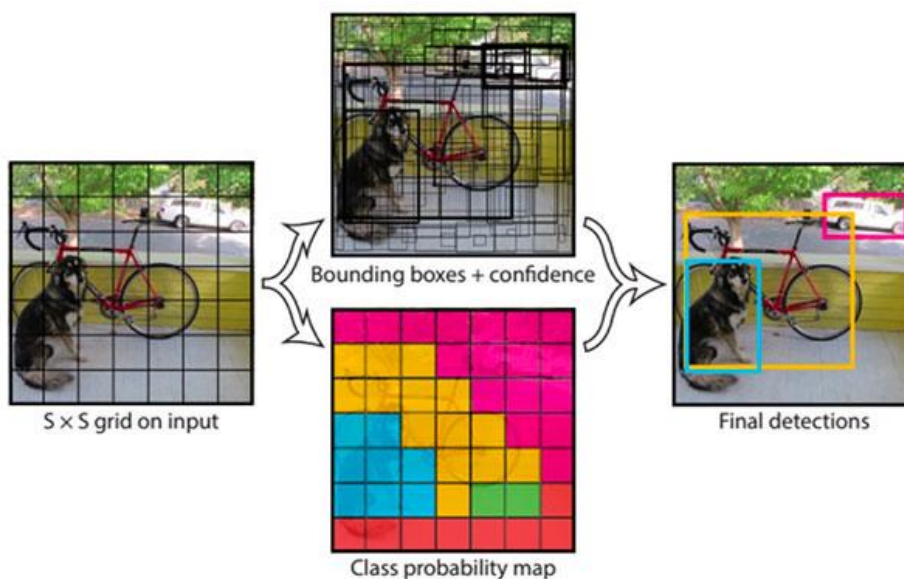


Figure 3.9 A Simplified Illustration of the YOLO Object Detector Pipeline [55]

The method is actually simple: the whole image is divided into $S \times S$ grid before (let's say $S = 7$). Now when the training data is prepared, the objects whose frame is known are taken, and the 7×7 pieces are prepared in the training data according to whether or not they fall into these object frames / boxes.

Each grid cell estimates only one object. For example, the following yellow grid cell attempts to estimate the object that locating in the center (pink dot) grid cell. Each grid cell estimates a fixed number of boundary boxes. In this example, the yellow grid cell makes the border box prediction (pink box) to find out where the fountain is in Figure 3.10 and Figure 3.11.

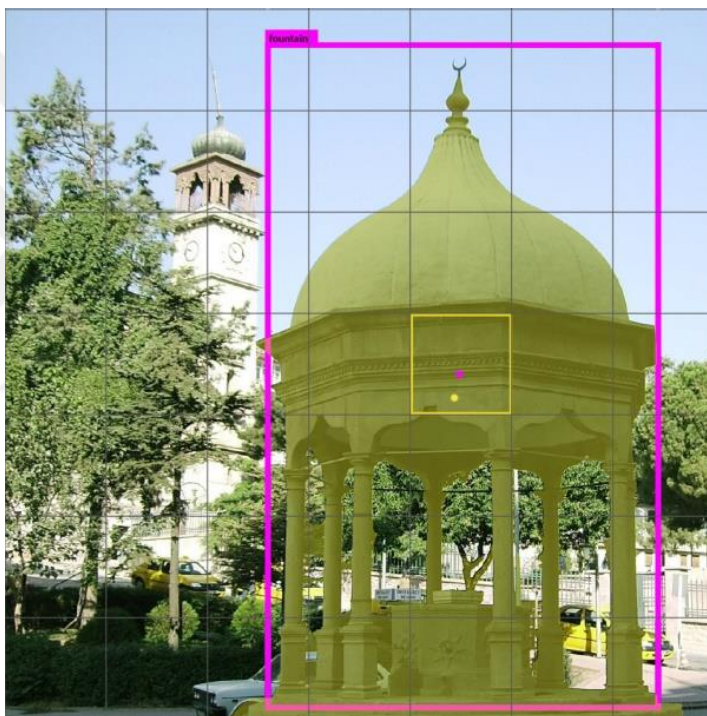


Figure 3.10 Boundary Box Example 1

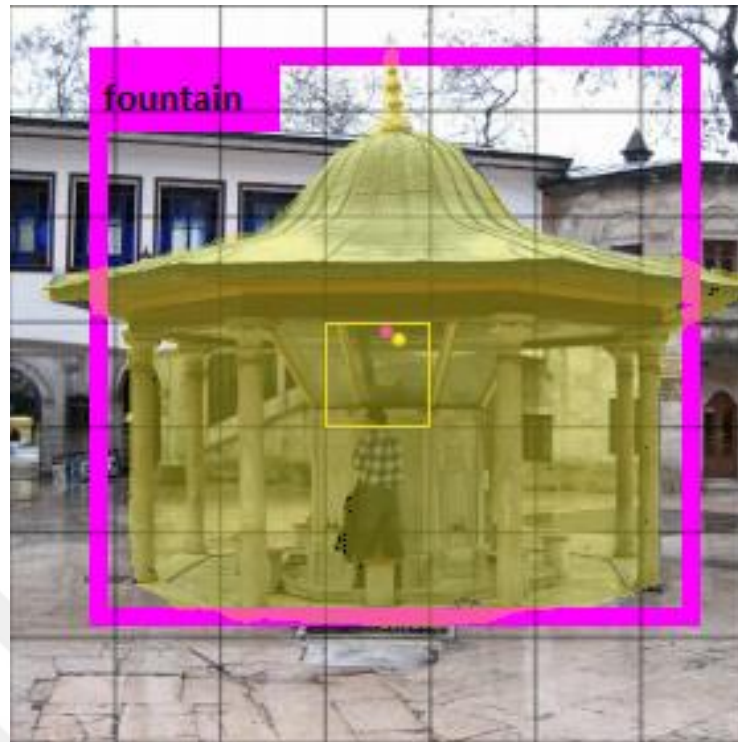


Figure 3.11 Boundary Box Example 2

However, an object rule limits how close objects can be detected. For this, YOLO has some limitations about how close objects can be. For example, suppose there are 10 people in any photo, but YOLO can only detect six. For each grid cell, YOLO predicts B boundary boxes and each box has one box confidence score, detects one object only regardless of the number of boxes B and predicts C conditional class probabilities that mean one per class for the likeliness of the object class.

The example structure where YOLO uses 7×7 grids ($S \times S$), 2 boundary boxes (B) and 20 classes (C) is shown below in Figure 3.12.

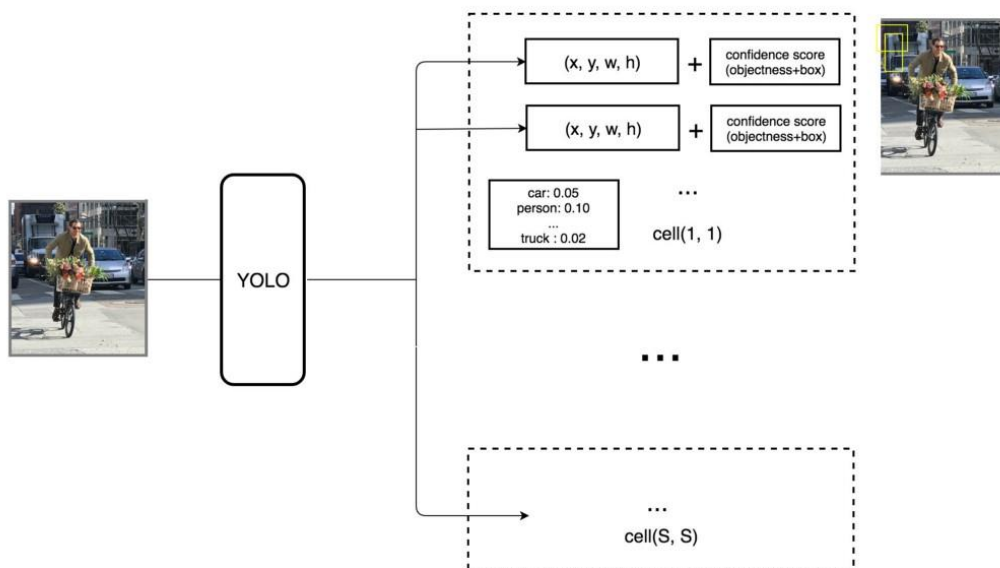


Figure 3.12 YOLO Predictions on 20 Classes [56]

Each boundary box contains 5 elements with x , y , w , h and a box confidence point. The confidence score shows how likely the box contains an object and how accurate the boundary box is. Boundary box width w and height h normalize with picture width and height. x and y are offset to the corresponding cell. Thus, x , y , w and h are between 0 and 1. Each cell has the possibility of a conditional class of 20. The conditional class probability is the probability that the detected object belongs to a particular class (one probability per category for each cell). Therefore, the prediction of YOLO $(S, S, B \times 5 + C) = (7, 7, 2 \times 5 + 20) = (7, 7, 30)$ means tensor. A sample tensor is shown below in Figure 3.13.

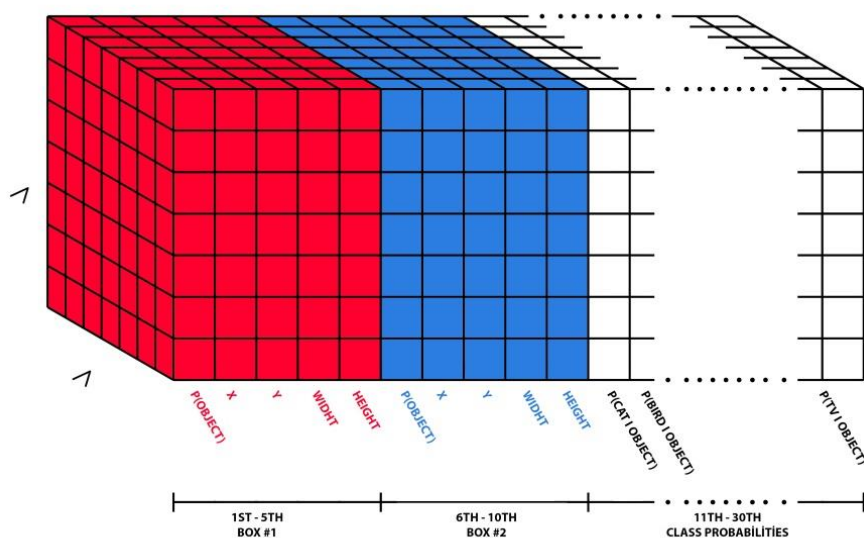


Figure 3.13 A sample Tensor

The main concept of YOLO is to establish a CNN network to predict one $(7, 7, 30)$ tensor. A tensor is for each single image as above a label. If a box falls into the center of an object frame in the training data, the upper P (Object) corresponding to that box, i.e. the trust coefficient, is set to '1'. Since the other boxes fall into the center of the object, they are 1 on the same matrix. One box can estimate up to a maximum of B ($B = 2$ on top), red zone on top 1, blue zone on top 2. The box will also predict the coordinate of the object within the whole image, and its magnitude in the same tensor, these matrices stand side by side in the section.

It uses a CNN network to reduce the spatial size to 7×7 at 1024 at each location. YOLO performs a linear regression using two completely interconnected layers to make the $7 \times 7 \times 2$ boundary box estimates (see image below). In order to make a precise estimate, we keep the ones with high confidence levels (greater than 0.25) as our latest estimates (correct image).

The class confidence score for each prediction box is calculated as follows:

$$\text{class confidence score} = \text{box confidence} \times \text{conditional class prob.} \quad (2.1)$$

It measures the confidence in both classification and localization, where an object is located. We can easily mix these scoring and probability terms. Here are the mathematical definitions for your future reference.

$$\text{box confidence score} \equiv P_r(\text{object}) \cdot \text{IoU} \quad (2.2)$$

$$\text{conditional class probability} \equiv P_r(\text{class}_i | \text{object}) \quad (2.3)$$

$$\text{class confidence score} \equiv P_r(\text{class}_i | \text{object}) \cdot \text{IoU} \quad (2.4)$$

$$\text{class confidence score} = \text{box confidence} \times \text{class confidence score} \quad (2.5)$$

Where;

- $P_r(\text{object})$: The probability of the box contains an object.
- IOU (intersection over union): intersection between the predicted box and the ground truth
- $P_r(\text{class}_i | \text{object})$: The probability of the object belongs to class_i given an object presence
- $P_r(\text{class}_i)$: The probability of the object belongs to class_i

A few months back, the third version of YOLO came out YOLOv3 uses several tips to improve training and improve performance: multi-scale estimates, a better backbone classifier and more. YOLO v3 now performs multi-label classification for objects detected in images. Previously, in YOLO, the authors used the class to softmax the class points and to get the maximum score as the class of the object in the bounding box. This was changed in YOLOv3.

As shown in Figure 3.14 YOLOv3 uses several tips to improve training and improve performance: multi-scale estimates, a better backbone classifier and more.

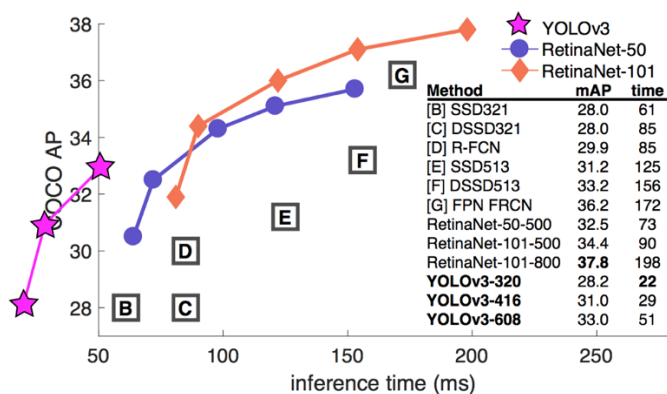


Figure 3.14 YOLOv3 Performs Very Well in the Fast Detector Category When Speed is Important. [57]

3.3 Object Detection

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos.

Object detection and object recognition have long been an indispensable need for computerized vision applications. Different algorithms have been developed for this topic, which has been studied for many years. But the revolutionary algorithm was the Viola Jones algorithm developed by Paul Viola and Michael Jones in 2001 [58]. This algorithm has been announced with the article entitled Simple Object Detection using a Simple Object Detection [59]. Many algorithms were developed in the following process. Recently, GPU technology has been used to accelerate the development of deep learning methods with much more accuracy.

In the Figure 3.15, the object detection and recognition processes are summarized under 5 main topics [60]. These are the stages of data entry, data preprocessing and stages, attribute extraction and attribute selection, and finally the classification stages of data.

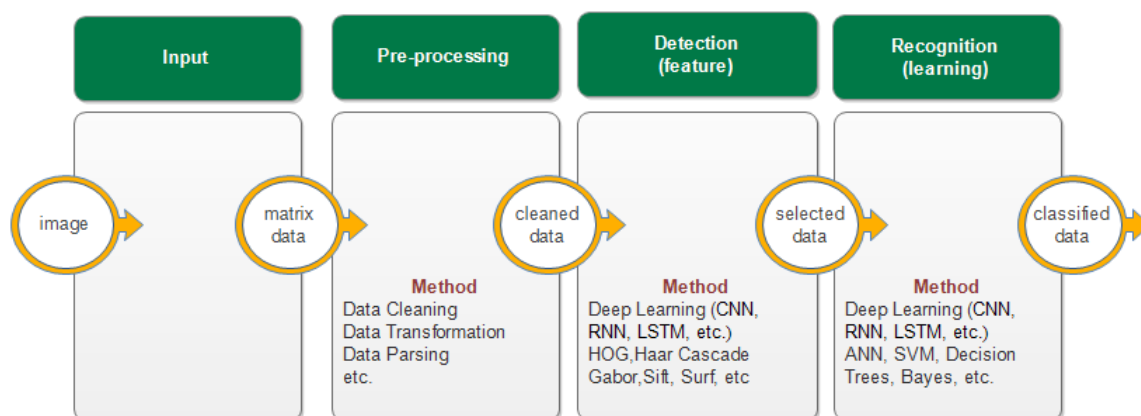


Figure 3.15 Object Detection and Recognition Stages

The data prepared in this step are given as input to the system for purposes such as being separated from the noises and being brought to the desired format. A human, automobile or house can be defined as input to a desired data, but a forest photo with a human being on it. Pictures taken with cameras, digitally generated images, video movies, and scanned text are examples of these data types. Data format is used in technology, language, and platform so on. It depends on many factors but the valid format for the algorithms to be used is matrixed image pixels. An example of raw input and the output of the detected object is given in Figure 3.16 and Figure 3.17.



Figure 3.16 A Raw Input [1]



Figure 3.17 An Example Detected Fountain

3.3.1 Pre-processing

In the image preprocessing stage on Figure 3.18, filtering is performed on the image, graying the picture and converting it to binary picture respectively [61]. After these processes are carried out, the objects of interest on the image are rendered more prominent and easier to process.

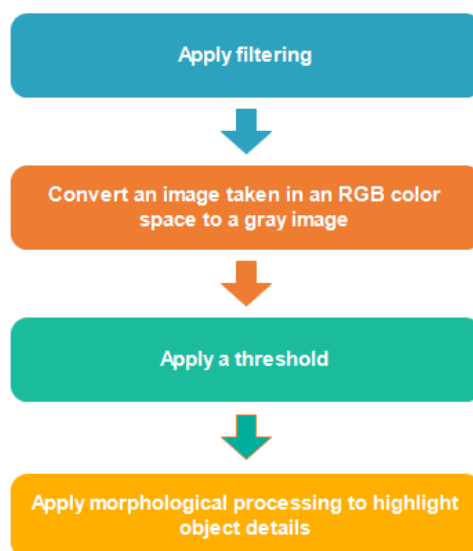


Figure 3.18 The Steps of Image Pre-Processing Stage

In the filter application step, it is provided to remove the salt pepper noises on the image and to reduce the unnecessary details in the picture. After filtering, the coloration of the color image is carried out. On the gray image, only the sections of the relevant objects are used by applying the thresholding process. The smallest (min) and maximum values (max) used in the thresholding process are determined by experimental studies. By comparing the pixel values in the gray image between the min and max values, a new value assignment is made for the binary image. After the panning process, an image containing black and white colors is created. Inside the image, there are undesirable white spots in black areas and black spots in white areas. Morphological process is applied in order to erase the noise on the double image obtained. In the morphological process step, the image is updated using neighboring pixel values in the structural element and binary image values. An example image is given in Figure 3.19.



Figure 3.19 Left: Noisy Photo Right: After the Pre-process [62]

There are a number of algorithms or methods for data preprocessing, which can be controlled by the human eye or even by complex neural networks. These methods include regression, stripping, clustering, filtering, binning, decision trees and so on. Another point that should not be forgotten is that sometimes it is not enough to apply a stand-alone method.

3.3.2 Object Detection with Deep Learning

Recognition is to extract from the detected image, the stage of understanding what this image is. In this section we need to touch on two clusters; classification and

clustering. The classification is to include the data in one of the predetermined classes. Supervised learning is carried out by estimation and pattern recognition methods. In clustering, similar data are collected in the same group.

In the field of machine learning, artificial neural networks have been frequently used in solving many problems. However, due to hardware constraints and other problems in the period known as Artificial Intelligence Winter Sleep, works in this area have come to a halt.

Artificial neural networks, which started to be a field again in the early 2000s, have shifted from shallow to deep networks with the development of GPUs. This approach has been used successfully in a wide range of areas ranging from image processing, natural language processing, and medical applications to activity recognition.

3.4 Object Segmentation

Before applying any recognition technique, some manipulation is required in the images [63]. This process prepares the image and improves its quality to eliminate the unrelated information and increase the selection of important features that should be recognized, this is known as pre-processing. Image processing, which is used in many scientific fields such as pattern recognition, robotic systems, diagnostic medical imaging and different applications in engineering and technology, is considered to be the most important area in terms of image processing [64]. Therefore, one of the biggest challenges of computerized vision is image segmentation [65], which is performed to improve the robustness of the extracted features to assist the classification process. It is the process that divides the digital image into regions that share the color, density, or texture of the object.

Several methods have been proposed to recognize objects in the image [66]. Some of the authors relied on multiple thresholds to break down objects like [67]. The other used the color for object segmentation, which it classifies by using artificial neural networks (ANN). It is proposed by coin segmentation [68], which depends on local entropy and gray value ranges. Moreover, in the medical field, the image is used to divide the image into various regions to identify tumors, for example [69] they

focused on two categories, namely deformable models and classification approaches. This approach is based on the limit of delimiting the object as important factors for separating the RGB color and object from its background.

Another efficient method is foreground extraction from the background in an image. The idea is to find the foreground and remove the background. Foreground extraction is a technique that allows an image's foreground to be extracted for further processing like object recognition, tracking, modelling etc. The algorithm used for foreground extraction is GrabCut Algorithm [70]. In this algorithm, the region is drawn according to the foreground and a rectangle is drawn on it. This is the rectangle that surrounds the main object. Regional coordinates were decided to understand the foreground mask. However, this partitioning is not perfect because it may have marked some foreground areas as the background and vice versa. This problem can be prevented manually.

3.5 3D Modeling

In 3D computer graphics [71], 3D modeling or direct modeling is a mathematical expression of an object in all three-dimensional form. Various computer programs are used to model the image of any living or non-living entity in three dimensions. The resulting product is called 3D model. 3D modeling can be done automatically or manually. Preparing a geometric model for the object to be designed manually is similar to dealing with plastic arts or sculpture. You will prepare the geometric data in 3D form.

3.5.1 Models

3D models define a physical entity with specific points in space geometry. These models emerge with the combination of various geometric shapes. In a holistic system, triangles, circles, lines and curves are used. If the data are available, the 3D design can be performed manually using the simulation program or by scanning.

3D modeling is widely used in 3D graphics. As a matter of fact, 3D models existed before 3D graphics were widely used on personal computers. For example, although

many computer games have 3D modeling, 3D graphics have not yet been designed to be personalized on a personal computer. There were also 3D models before the spread of computer programs.

Today, 3D models are used in a wide variety of areas. 3D models of organs in the medical sector are used very effectively. For this, the data were generated by combining the large number of 2-dimensional photographs taken in the MR machine. In the cinema sector, 3D animations are widely used in animation images. The game industry still uses 3D models in detail in many computer games. The use of 3D models in science is also important. For example, we have the ability to symbolize a chemical compound in 3D with the 3D model. In this way, the molecular geometry is also revealed.

Many programs for 3D modeling have been created in the field of architecture. Because of the 2-dimensional plan of the structure to be revealed as well as the final analysis to be taken in the 3D state is desired to be monitored. Prototype is prepared before any product in engineering. This is only possible with the 3D model. Earth scientists have also recently started to study landforms through 3D models. 3D models also form the basis of a 3D object to be produced.

3.5.2 Type of Modeling

3D design can be done in two ways. These are solid modeling and surface modeling. Solid modeling reflects the actual volume of the object to be modeled. This type of modeling is very realistic, but it is also difficult to reveal as a result. For example, a stone model is presented by a 3D stone with the same shape. Simulations in the field of health and engineering are not visual. Therefore, solid modeling is used more often in these areas.

In surface modeling, volume is not modeled. Only the shell part of the existing object is modeled in order to form a border. Imagine an evacuated egg inside. However, only hollow models are obtained by surface modeling. Such models are easier to find, but not as solid modeling for production.

Because the image of an object is very important, surface modeling in computer graphics is widely used. 2D surfaces help us to understand models in computer graphics. Because the images presented here are not certain. Because of the lack of precision, graphic design is based on a design that is close to reality. In recent years, point designs have also become widespread in this field. After defining the external surfaces of the object with the points in the coordinate system, these points are combined with the appropriate lines to determine the boundaries.

3.5.3 Modeling Process

There are three common methods for introducing a model. These are:

- **Polygonal modeling:** The nodes defined in 3D space are connected to each other by various lines. They form a polygon model together. Today, the most widely used modeling management. This is because computers can easily detect and implement these models. However, the model obtained in polygonal modeling is planar. For this reason, curvilinear images can only be drawn at certain proximity using the lines.
- **Curve modeling:** The surfaces are expressed with curves in this kind of modeling. These curves arise by fusing various points. These points are the imaginary points on the curves. Each point has a weight on the curve. As the weight of the point increases, the likelihood of the curve being close to that point increases in the same way. Types of curves also include NURBS structures, which are an abbreviation of non-uniform rational B-spline expression.
- **Digital Sculpting:** It is a very new system in modeling. The spread has only begun in the last few years. Digital sculpting can be done in different ways. The displacement may be in the form of volumetric and dynamic mosaic. Displacement or placement is the most widely used type of digital modeling. In this system, the necessary points in the image are made by sculpting. A very beautiful image appears in the models introduced by digital modeling.

3.5.4 Modeling Techniques

There are many different techniques used in modeling. There is much software available to implement these techniques. Cinema 4D, Maya, Blender and Solid Thinking are the most commonly used applications. Besides, AutoCAD, SolidWorks and Rhino programs which are used extensively in engineering designs are used for 3D design. The more complicated structures put the model into use by using more than one system. The both curves, the right and the polygon can be used in the same model.

3.6 Locating the camera at perspective location

It is necessary to move the camera position in the computer environment in order to get the image of an object taken at different angles. It is assumed that the camera position is directly opposite the object in that picture while acquiring an eye level image of a captured image. The camera can be moved using sequence of rotations and translations.

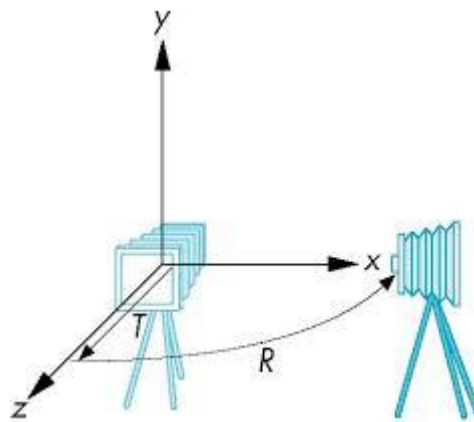
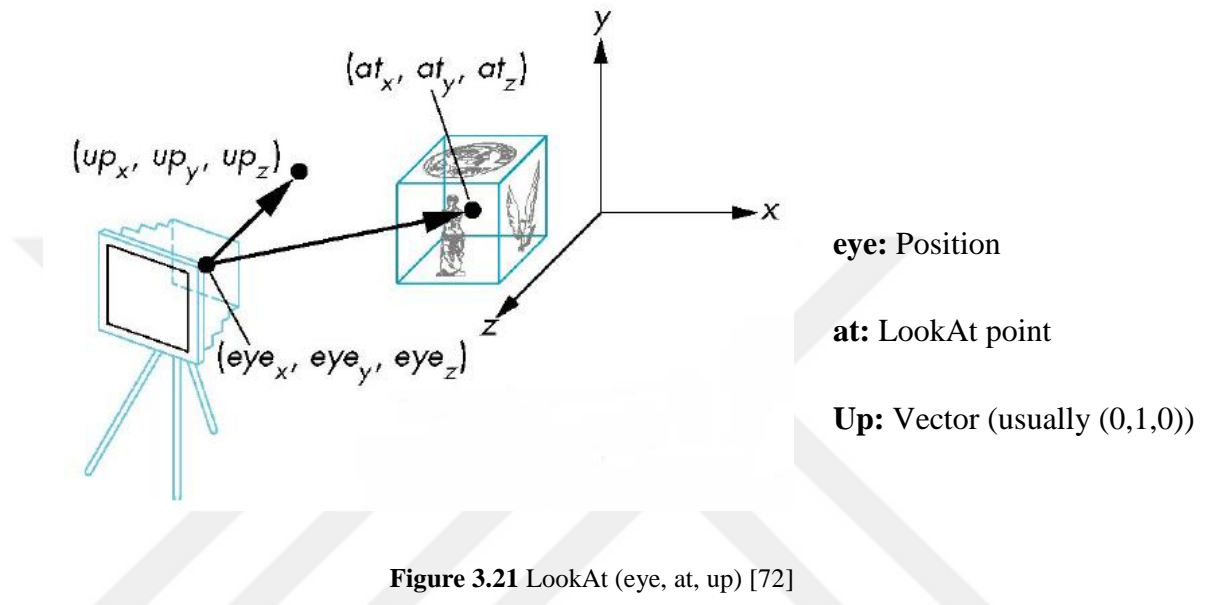


Figure 3.20 Side View [72]

As shown in the example in Figure 3.20, the camera is rotated, moved away from the origin and the model-view matrix calculation is performed to obtain the side view. The object distances are calculated according to the camera determined by the model view matrix.

The camera does not move at all in a computer program. In fact, the world moves in the opposite direction and directs you how you want the camera to actually move.

Using the LookAt method, we can calculate the appearance matrix directly from the camera's world position (*eye*), a global up vector, and a target point we want to look at (Fig. 3.21).



The point where the two or more lines that are actually parallel are extended indefinitely, the point where the lines come together is the vanishing point.

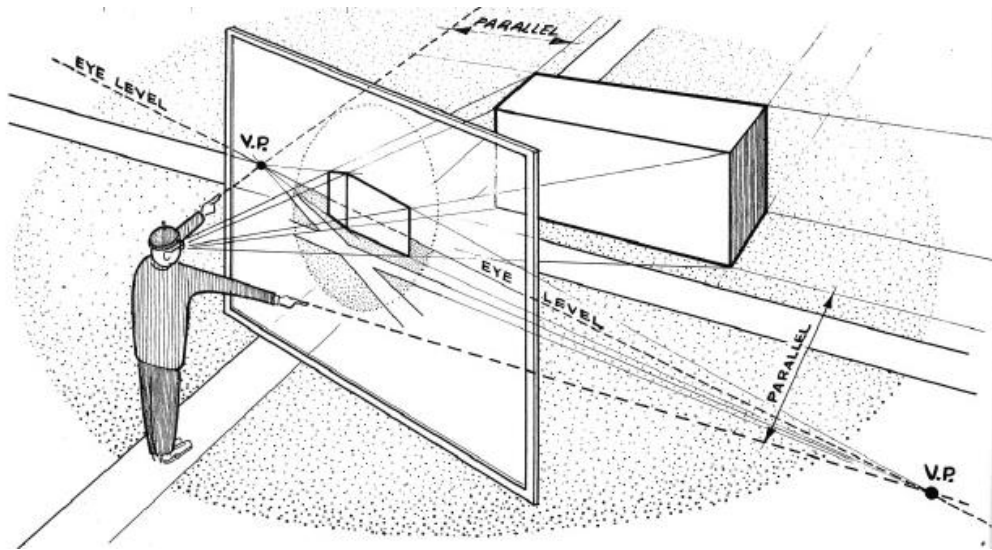


Figure 3.22 Illustration of Vanishing Points, Eye Level, Parallel Lines [73]

The example of positioning the horizon points shown in Figure 3.22 is described in the Perspective Drawing handbook [73]. "To pull down the object, we first create a top view or plan (left), which shows the object, the picture plane (seen as a line) and the position of the observer. On this plan, 'line of sight' is drawn, which points parallel to the object lines to accommodate the horizon points in the picture plane. Other lines of vision project the object itself into the image plane. Therefore, the picture plane line shows the relationship of the visible size of the object to the horizon points. This 'measurement line' is then transferred to the original image (right) where it is superimposed on the horizontal horizon line (eye level). The relationship remains the same, whether it is now on the line or underneath it or on the line."

CHAPTER 4

IMPLEMENTATION

4.1 Overview

A system is built for image-based modeling of fountains in 3D. The system takes a single image as the input and produces a 3D mesh model as the output by applying the steps summarized in Figure 4.1. In the first step, our model employs deep learning for detecting the fountain object in the input image. Then the background in the image is removed and boundaries of the detected object are determined. These boundaries are used for generating a 3D mesh model resembling generalized cylinders. Finally, a texture made from the input image is mapped over the 3D model. This study was presented at International Conference Eurasia Graphics [74].

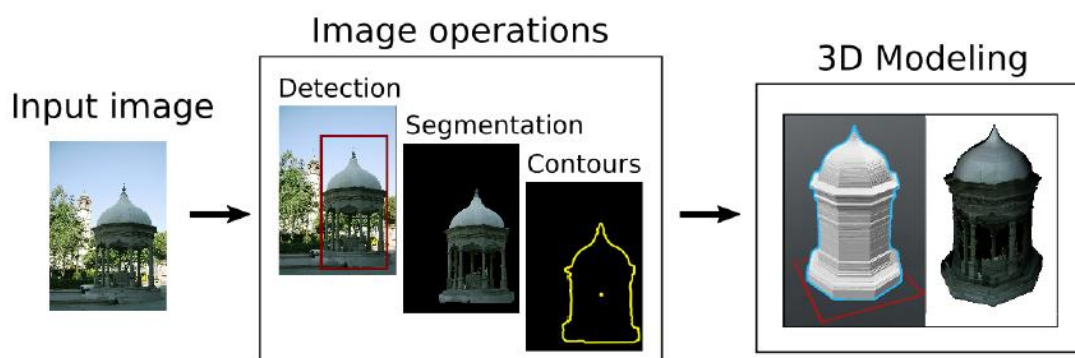


Figure 4.1 Overview of the System [74]

4.2 Fountain Detection

The goal of object detection is to find an object of a predefined class in a static image or video frame. There are lots of detection algorithms [75] in computer vision including feature extraction and training steps. In this study, Convolutional Neural Networks (CNNs) that include both feature extraction and classification was used to detect fountain objects.

Detection process is implemented with Darknet Yolo [76] which is a framework for detection and classification with CNN architecture. YOLO, abbreviated from “You Only Look at Once”, is an open source neural network framework written in C and CUDA. It is currently the state of the art on real time object detection methods. This model has several advantages over classifier-based systems. It applies a single neural network to the full image, so its predictions are informed by global context in the image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. We have trained the system with a dataset of 200 fountain images acquired from the Internet. After training the system with CNNs, the fountain objects are detected on test images as shown in Figure 4.2.



Figure 4.2 Prediction Images with the Bounding Box of the Detected Objects.

4.3 Fountain Segmentation

The process of separating an image into foreground and background is known as image segmentation and it is a major preprocessing step in many vision-based applications. The fountain object is segmented out from the background by using GrabCut technique [70]. GrabCut uses information encapsulated in the image as in most segmentation techniques. It makes use of both edge and region information. This information is used to create an energy function which, when minimized, produces the segmentation. Initially user draws a rectangle around the foreground region. In this study, the foreground region is determined by the bounding box obtained from the object detection step. A Min-cut/Max-Flow algorithm is used to segment the graph. This algorithm determines the minimum cost cut that will separate the Source and Sink nodes. The cost of the cut is determined by the sum of all the weights of the links that are cut. Once the Source and Sink nodes are separated, all pixel nodes connected to the Source node become part of the foreground, and the rest become part of the background. Figure 4.3 shows a simplified diagram of the GrabCut approach.

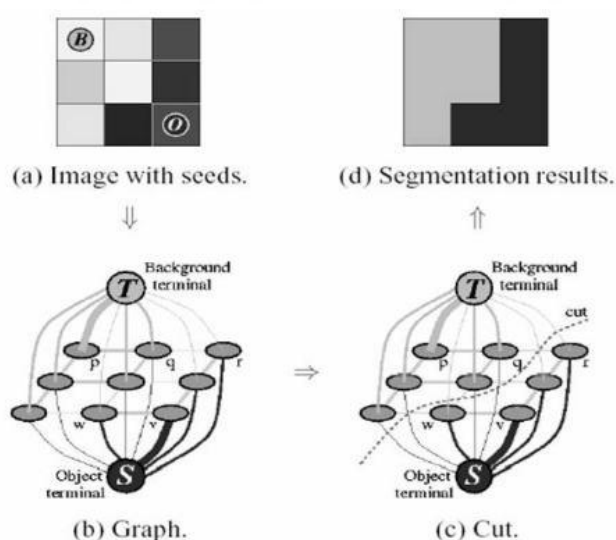


Figure 4.3 Diagram of the GrabCut [77]

We have used OpenCV's implementation of GrabCut for Figure 4.4 and Figure 4.5 shows a sample result of segmentation, eliminating background while keeping the fountain as the foreground object.

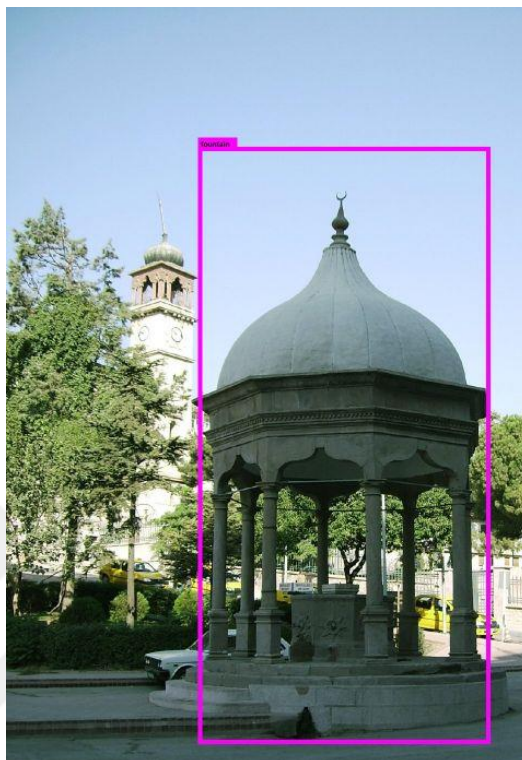


Figure 4.4 Prediction Image [78]

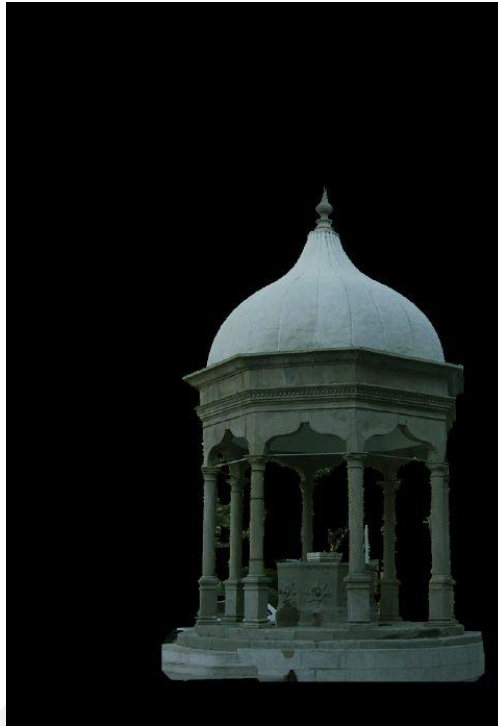


Figure 4.5 Segmented Object using GrabCut

4.4 3D Fountain Modeling

After the segmentation step, we have an isolated fountain image which needs to be converted to a 3D model. Here, we utilize the symmetric form of fountains. The general pattern of the model is formed by giving approximate radius values from top to bottom according to the symmetrical structure of all the side surfaces of the fountain. These radius values are calculated using contours of the fountain image. Contours are extracted by flood filling the foreground object and applying an edge detection method. In order to remove high frequency noise from the extracted boundaries, the image is low-pass filtered before edge detection. Contour extraction is shown in Figure 4.6, Figure 4.7, and Figure 4.8.

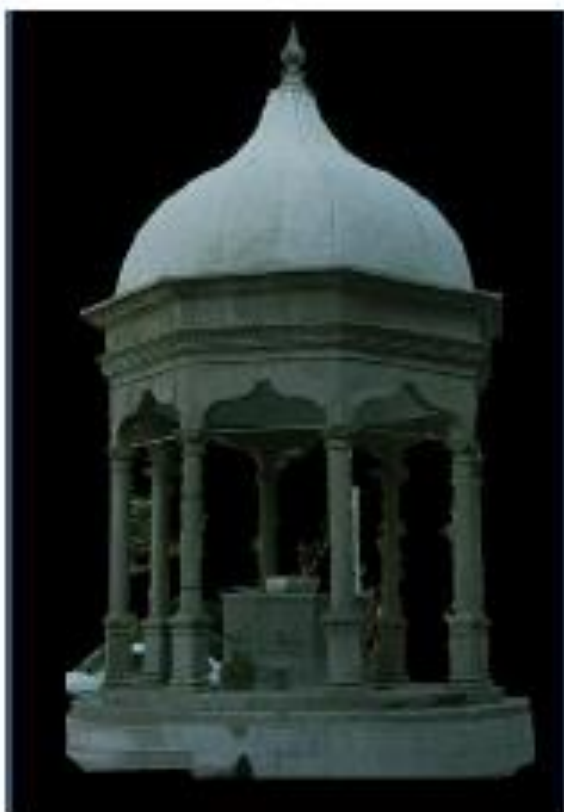


Figure 4.6 Segmented Image

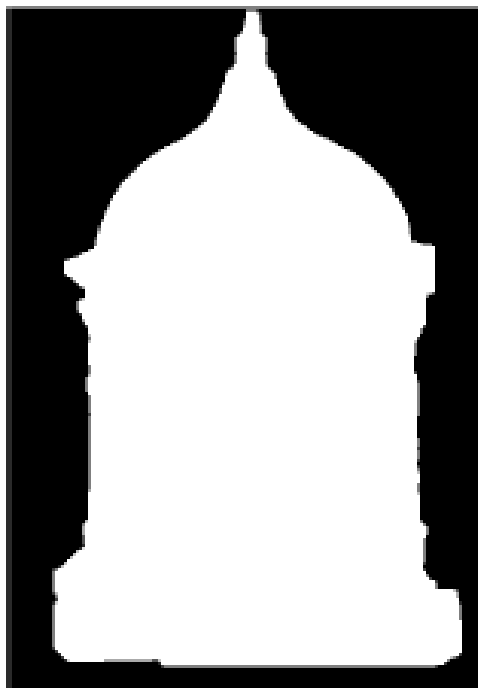


Figure 4.7 Foreground Image by Flood Filling.

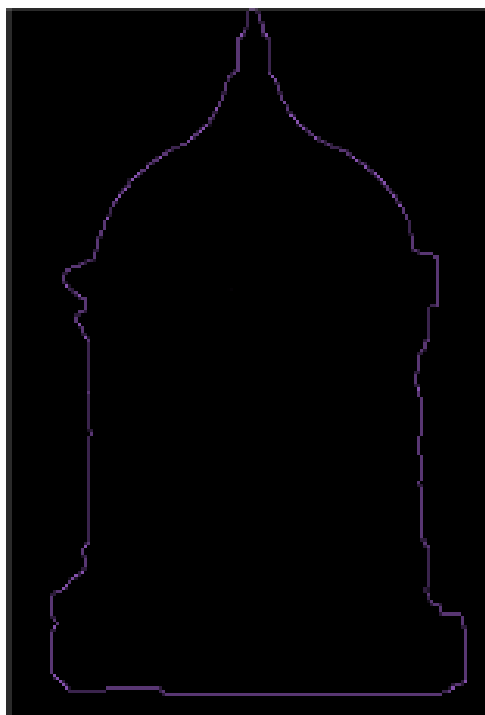


Figure 4.8 Extracted Contour of Fountain

Radius values of model, r_i , are measured according to distance between the x coordinate of every contour point and x coordinate of the fountain center. The important parameters are shown in Table 4.1.

Table 4.1 Parameters of A Model

r_i	the radius of the index-i circle face
x_0	the centers coordinate on the drawing
x	the x coordinate on the drawing
$r_i = (x_0, x)$	where $x_0 < x$

Having the radius values, a 3D model of the fountain is generated in the form of generalized cylinders as shown in Figure 4.9, Figure 4.10, and Figure 4.11.

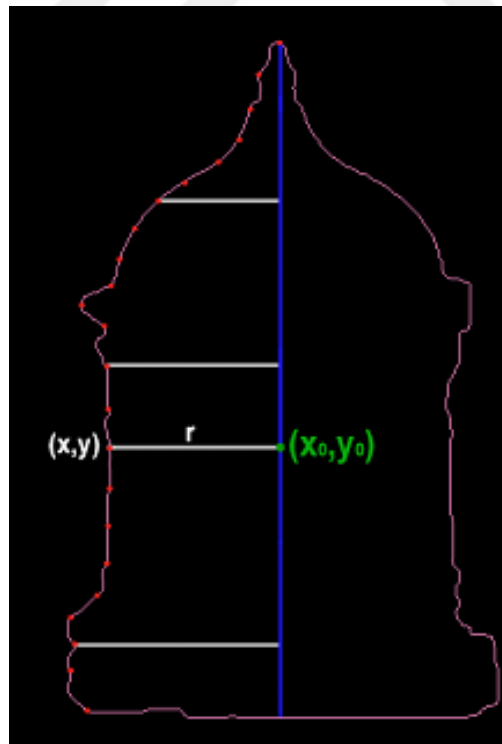


Figure 4.9 Determining the Radii of Model for Each Height

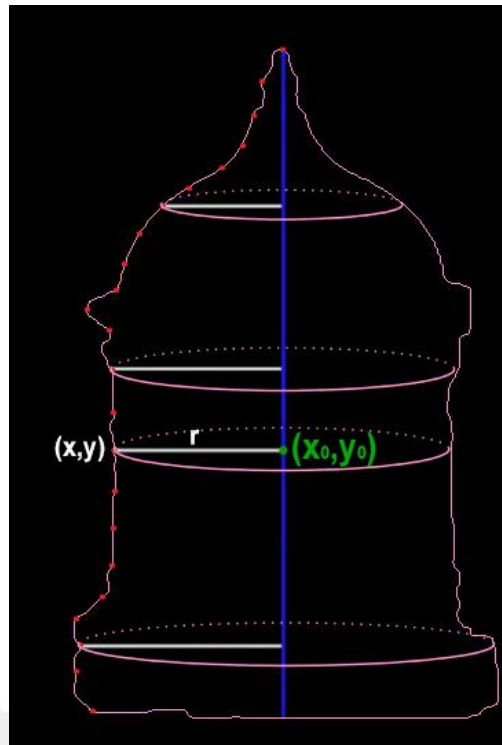


Figure 4.10 Generating 3D Model with Radii for Each Height

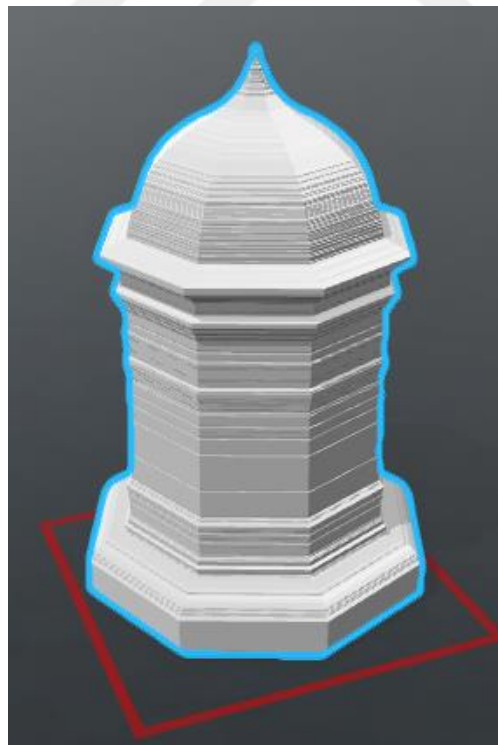


Figure 4.11 3D model

After the generation of the 3D model of the fountain object, texture coordinates are calculated for every vertex of the model and the input image is used as the texture map to increase realism of the output. Figure 3.12 shows texturing to 3D model of sample fountain.



Figure 4.12 Textured 3D model.

3.5 Locating the camera at perspective location

In the dataset we collected for training and testing, mostly the fountains were drawn from different angles, not from the opposite direction. When taking the fountain, the camera position is not the opposite of the fountain, and the photographs are one of the biggest problems we encounter for 3D modeling and especially texturing. If the camera was taken above or below the center of the fountain, when we modeled these photographs, the photo and the model did not match exactly, as shown in Figure 3.13 and shifts occurred.



Figure 4.13 Output of Shifted Texture Coordinate of the Sample 3D Fountain Model

In order to solve this problem, we tried to find the position of the camera and transform it with the matrix calculations.

3.5.1 Vanishing Point

Since the fountains have a very edged structure, this method, which is parallel to each other, was not suitable for every problem. For some fountains, the camera position where the photo was taken was found to be correct.

As shown in Figure 4.14, Figure 4.15, Figure 4.16, and Figure 4.17 the point where the intersection of the edge of the fountain in the first picture can actually be the same point as the eye level of the camera can estimate. However, in the second

picture, we cannot make an accurate prediction about the intersection of the edges of the fountain and the camera location.

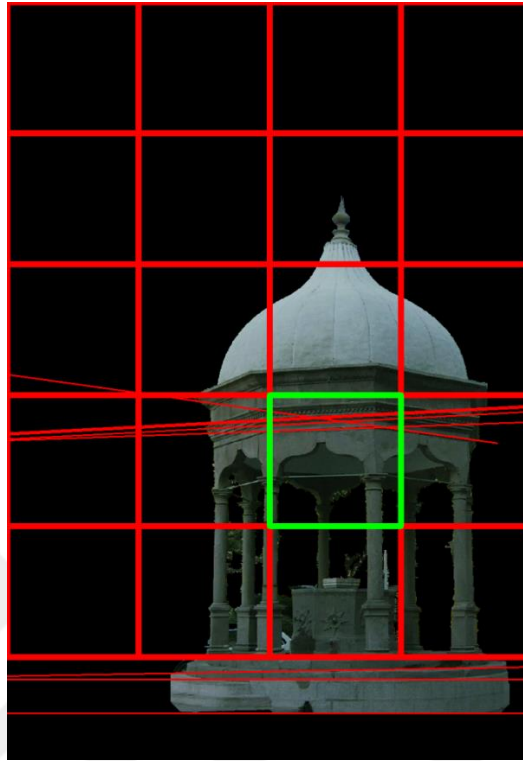


Figure 4.14 Finding Possible Vanishing Point of Sample Fountains with Rectangle Box 1

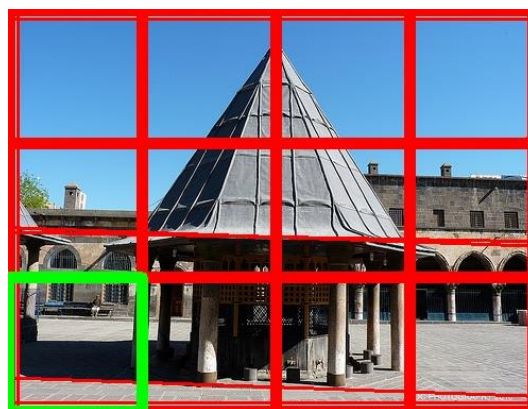


Figure 4.15 Finding Possible Vanishing Point of Sample Fountains with Rectangle Box 2 [55]

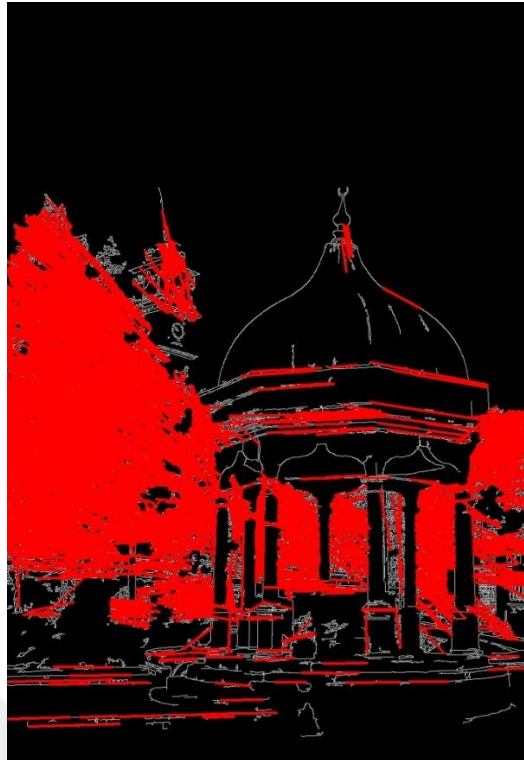


Figure 4.16 Point of Intersection of Lines Passing Through the Edges of the Fountain 1

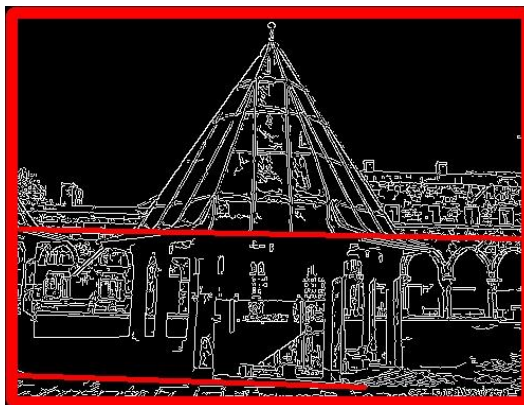


Figure 4.17 Point of Intersection of Lines Passing Through the Edges of the Fountain 2

Because of these results, we have seen that it would be of no use to find vanishing points in finding the eye-level location of the camera in order to get the image taken right across the fountain.

CHAPTER 5

RESULTS & DISCUSSION

In this study, we have proposed a simple way for generating a 3D model of fountain object from a single image. The example outputs of project is shown in Figure 5.1. We create models based on the structural characteristics of the fountain objects. Firstly, to find location of the fountain on an image, detection operation is performed by taking advantage of the information that the fountains are in a hierarchical structure with a dome on top, a pool in the middle, and columns. Although they are generally in the same hierarchical structure, there are fountains with different architectural design such as differently shaped dome, hexagonal, octagonal, etc. Therefore, it would be difficult to recognize them with traditional machine learning methods [79]. If data show a hierarchical structure such as a cat (has whiskers, paw, etc.), deep learning has no difficulty in learning. It learns this hierarchical structure.

We used CNNs and GPU acceleration from the deep learning methods to get robust results in the detection step. The results suggest that CNNs can successfully detect fountain objects despite the variety of their styles. After the detection step, 3D model is generated by taking advantage of the symmetrical structure of all the side surfaces of the fountain. To find symmetric profile of the fountain, fountain image is separated from background by applying image segmentation technique using GrabCut method. Then, contours of the fountain image, which helps determining the radius of surface of revolution, are found.

This approach gives satisfactory outputs if fountain images are not taken from very close distances, ideal case would be images with an orthographic projection which is not possible with ordinary cameras. To increase the quality of outputs, camera position relative to the image can be estimated using vanishing points in the image. Although the system is currently used for modeling fountains, it can also be used for other structures having rotational symmetry.



Figure 5.1 Results. Left: Input Images, Middle: Operations to Generate 3D Model, Right: Outputs

Limitations: Despite the success of the proposed system to generate 3D models, it has several limitations that need further efforts. First, the current algorithm does not model every features of the fountain, but only models using symmetric surface information. Second, boundary box of some detected fountain cannot fully enclose the fountains, they occasionally extend outside the boundary boxes, causing incorrect segmentations. As stated before, determining the camera location would increase the success of 3D modeling and texture mappings steps.

We mentioned that the fountains differed according to their forms, periods and materials. According to the forms of fountains, also roofless or roofs according to the column and polygonal structure varies. There are different roofs like flat, pyramid, cone, dome. Novel, onion and ottoman, as well as different types of dome. Different fountain styles have been built in different periods throughout history, including the unique design of that period. Styles of Late Abbasi, Ayobi, Seljukian, Andalusi, Ottomon are known. At the same time, while building fountains, materials such as stone, wood, bricks and copper lead provide information about the decoration and construction of the fountain.

As shown in Figure 5.2, we can see that the fountain like form of root, material and period can be classified with each individual feature.

Form of Roof	Material	Period
<ul style="list-style-type: none"> • Roofless • Roof <ul style="list-style-type: none"> • Flat • Pyramid • Cone • Dome <ul style="list-style-type: none"> • Roman • Onion • Ottoman 	<ul style="list-style-type: none"> • Stone • Wood • Bricks • Copper • Lead 	<ul style="list-style-type: none"> • Late Abbasi Style • Ayobi Style • Seljukian Style • Andalusi Style • Ottoman Style

Figure 5.2 Classifications of Fountains

Now, to be able to detect fountains according to their styles, and then we continue our studies for a system which can create a 3D model according to the style of the detected fountain. In the following as shown in Figure 5.3, which is classified according to the form of the roof, the dome is detected by examples of the fountain detection.



Figure 5.3 Detection of Fountain with Dome Roof 1



Figure 5.4 Detection of Fountain with Dome Roof 2

During the 3D modeling phase, the information of that feature will enable the creation of a 3D pattern model. For example, in each fountain style; assuming that the 3D model of a fountain with a roof-like structure may be more or less the same.

Also, the detection of the materials used in the construction and decoration of fountains causes the 3D model to appear more realistic in the texturing phase.

In our current study, we aim to provide a system to be transferred the feature knowledge of the detected fountain by deep learning methods to a 3D model as a parameter .

As a conclusion, we have proposed a system for generating fountains from single images. The algorithm employs recent advances in machine learning to extract the fountain in an image and generates its 3D model exploiting its symmetric form. The results indicate that the system can reconstruct the general structure of the fountains in the photos. This study also helps cultural heritage by enabling easy modeling and documentation of historical architecture. In the future, the systems performance can be further increased by including more training images in object detection part. We also plan to automatically estimate the polygonal structure (how many columns) of fountains and plan to model every properties of the fountains, e.g., taps, chairs etc, using procedural modeling. Also, texturing the 3D model according to the camera viewpoint remains as a future work. We expect to develop a fully automatic system to store 3D models of historical artifacts, not only fountains, in a database. Lastly, our study can be extended to recognize the architectural style of the modeled architecture and categorize them accordingly.

REFERENCES

- [1] M. Cambaz, <http://www.mustafacambaz.com/>, accessed: 2017-08-29.
- [2] Guidi, G., Beraldin, J. A., & Atzeni, C. (2004). High-accuracy 3D modeling of cultural heritage: the digitizing of Donatello's" Maddalena". IEEE Transactions on image processing, 13(3), 370-380.
- [3] Koller, D., Frischer, B., & Humphreys, G. (2009). Research challenges for digital archives of 3D cultural heritage models. Journal on Computing and Cultural Heritage (JOCCH), 2(3), 7.
- [4] Logothetis, S., Delinasiou, A., & Stylianidis, E. (2015). Building information modelling for cultural heritage: a review. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2(5), 177.
- [5] Gingold, Y., Igarashi, T., & Zorin, D. (2009, December). Structured annotations for 2D-to-3D modeling. In ACM Transactions on Graphics (TOG) (Vol. 28, No. 5, p. 148). ACM.
- [6] Ödekan, "Şadırvan" Maddesi, s.1719; Aytöre, "Türklerde Su Mimarisi", s.66; Önge, Türk Mimarisinde Selçuklu Ve Osmanlı Dönemlerinde Su Yapıları, s.27; Şapolyo, "Şadırvanlar", s.10; Arseven, "Sebil" Maddesi, s.1859; Pakalın, "Şadırvan" Maddesi, s.303.
- [7] Arseven, Türk Sanatı Tarihi, s.511; Özyurt, "Teknik Yönleriyle Bazı Konya Camiilerinin Şadırvanları ve Bunların Türk Şadırvan Mimarisindeki Yeri", s.117.
- [8] Bilaloğlu, Şadırvanların Mimarideki Yeri, Onarım, Bakım, Koruma, Geliştirme, Mimari ve Çevre Ögesi Olarak Kullanılma İmkânlarının Araştırılması, Konya Şadırvanlarından Örnekler, s.42-44.
- [9] İslâm Mîmârîsinde Şadırvan, <https://www.zdergisi.istanbul/makale/islam-mimarisinde-sadirvan-112>, accessed on May 2019.

- [10] Musul Ulu Camisi ve Katar Krizi <http://erdoganerken.blogspot.com/2017/03/yaz-deneme.html>, accessed: 2019-07-03.
- [11] Geyik G., “İzmir Su Yapıları”, Yüksek Lisans Tezi, Erzurum: Atatürk Üniversitesi, 2007.
- [12] Bilaloğlu, Şadırvanların Mimarideki Yeri, Onarım, Bakım, Koruma, Geliştirme, Mimari ve Çevre Ögesi.
- [13] Önge, Türk Mimarisinde Selçuklu Ve Osmanlı Dönemlerinde Su Yapıları, s.88.
- [14] Tuncer, Diyarbakır Camileri, s.144.
- [15] Önge, Türk Mimarisinde Selçuklu Ve Osmanlı Dönemlerinde Su Yapıları, s.92; Sözen, Türk Mimarisinin Gelişimi ve Mimar Sinan, s.252.
- [16] Galitekin, Kocaeli Su Medeniyeti Tarihinden Birkaç Damla, s.196.
- [17] Diyarbakır Behram Paşa Camii, <http://www.svbusinesshotel.com/diyarbakir-sehir-rehberi/behram-pasa-camii/>, accessed: 2019-05-31.
- [18] Özyurt, “Teknik Yönleriyle Bazı Konya Camiilerinin Şadırvanları ve Bunların Türk Şadırvan Mimarisindeki Yeri”, s.118; Bilaloğlu, Şadırvanların Mimarideki Yeri, . Onarım, Bakım, Koruma, Geliştirme, Mimari ve Çevre Ögesi Olarak Kullanılma İmkânlarının Araştırılması, Konya Şadırvanlarından Örnekler, s.238.
- [19] Çifci, F., “Kastamonu Camileri-Türebeleri ve Diğer Tarihi Eserler”, Ankara, 1995, s.298.
- [20] Tokay, E., “İstanbul Şadırvanları”, İstanbul, 1951, s.19; Tansuğ, “18. Yüzyılda İstanbul Çeşmeleri ve Ayasofya Şadırvanı”, s.100; Eminönü Camileri, Türkiye Diyanet Vakfı Eminönü Şubesi, s.26; TÖZ, İstanbul Camileri I.-II. Cilt, C.I, s.29.

- [21] Önge, “Türk Mimarisinde Selçuklu Ve Osmanlı Dönemlerinde Su Yapıları”, 94.2019-01-05.
- [22] TOPÇUOĞLU ŞADIRVANI - Kastamonu, <https://www.kulturportali.gov.tr/turkiye/kastamonu/kulturenvanteri/topcuoglu-sadirvani>, accessed: 2019-05-31.
- [23] Bilaloğlu A., “Şadırvanların Mimarideki Yeri, Onarım, Bakım, Koruma, Geliştirme, Mimari ve Çevre Ögesi Olarak Kullanılma İmkânlarının Araştırılması, Konya Şadırvanlarından Örnekler”, s.51.
- [24] Bilaloğlu A., “Şadırvanların Mimarideki Yeri, Onarım, Bakım, Koruma, Geliştirme, Mimari ve Çevre Ögesi Olarak Kullanılma İmkânlarının Araştırılması, Konya Şadırvanlarından Örnekler”, s.52.
- [25] Önge, “Türk Mimarisinde Selçuklu Ve Osmanlı Dönemlerinde Su Yapıları”, s.30.
- [26] Önge, “Türk Mimarisinde Selçuklu Ve Osmanlı Dönemlerinde Su Yapıları”, s.32.
- [27] Tokay, “İstanbul Şadırvanları”, s.6.
- [28] Mao, J. and Xu, L., “Automatic 3d reconstruction via object detection and 3d transformable model matching cs 269 class project report,” 2014.
- [29] Kholgade, N., Simon, T., Efros, A., & Sheikh, Y. (2014). 3D object manipulation in a single photograph using stock 3D models. ACM Transactions on Graphics (TOG), 33(4), 127.
- [30] Olmschenk, G., & Zhu, Z. (2014). 3D hallway modeling using a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 158-164).

- [31] Chen, T., Zhu, Z., Shamir, A., Hu, S. M., & Cohen-Or, D. (2013). 3-sweep: Extracting editable objects from a single photo. *ACM Transactions on Graphics (TOG)*, 32(6), 195.
- [32] Kim, V. G., Li, W., Mitra, N. J., Chaudhuri, S., DiVerdi, S., & Funkhouser, T. (2013). Learning part-based templates from large collections of 3D shapes. *ACM Transactions on Graphics (TOG)*, 32(4), 70.
- [33] Eslami, S., & Williams, C. (2012). A generative model for parts-based object segmentation. In *Advances in Neural Information Processing Systems* (pp. 100-107).
- [34] Amit, Y., & Trouvé, A. (2007). Pop: Patchwork of parts models for object recognition. *International Journal of Computer Vision*, 75(2), 267-282.
- [35] Jain, A. K., Zhong, Y., & Dubuisson-Jolly, M. P. (1998). Deformable template models: A review. *Signal processing*, 71(2), 109-129.
- [36] Jain, A. K., Zhong, Y., & Dubuisson-Jolly, M. P. (1998). Deformable template models: A review. *Signal processing*, 71(2), 109-129.
- [37] López-Sastre, R. J., Tuytelaars, T., & Savarese, S. (2011, November). Deformable part models revisited: A performance evaluation for object category pose estimation. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (pp. 1052-1059). IEEE.
- [38] “The PASCAL Object Recognition Database Collection”, <http://host.robots.ox.ac.uk/pascal/VOC/databases.html>, accessed: 2017-08-29.
- [39] Yan, P., Khan, S. M., & Shah, M. (2007, October). 3d model based object class detection in an arbitrary view. In *2007 IEEE 11th International Conference on Computer Vision* (pp. 1-6). IEEE.
- [40] Marton, Z. C., Pangercic, D., Blodow, N., Kleinhellefort, J., & Beetz, M. (2010, October). General 3D modelling of novel objects from a single view. In

2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 3700-3705). IEEE.

- [41] Rothganger, F., Lazebnik, S., Schmid, C., & Ponce, J. (2006). 3D object modeling and recognition from photographs and image sequences. In *Toward Category-Level Object Recognition* (pp. 105-126). Springer, Berlin, Heidelberg.
- [42] Jiang, N., Tan, P., & Cheong, L. F. (2009, December). Symmetric architecture modeling with a single image. In *ACM Transactions on Graphics (TOG)* (Vol. 28, No. 5, p. 113). ACM.
- [43] Snavely, N., Seitz, S. M., & Szeliski, R. (2006, July). Photo tourism: exploring photo collections in 3D. In *ACM transactions on graphics (TOG)* (Vol. 25, No. 3, pp. 835-846). ACM.
- [44] Snavely, N., Seitz, S. M., & Szeliski, R. (2008). Modeling the world from internet photo collections. *International journal of computer vision*, 80(2), 189-210.
- [45] Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., & Szeliski, R. (2011). Building rome in a day. *Communications of the ACM*, 54(10), 105-112.
- [46] Furukawa, Y. & Ponce, J., "Accurate, dense, and robust multiview stereopsis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.
- [47] Şeker, A., Diri, B., & Balık, H. H. (2017). Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme. *Gazi Mühendislik Bilimleri Dergisi (GMBD)*, 3(3), 47-64.
- [48] Deng, L. and Yu, D., "Deep Learning: Methods and Applications," *Found. Trends® Signal Process*, vol. 7, no. 3–4, pp. 197–387, 2014.

- [49] Bengio, Y., “Learning Deep Architectures for AI,” *Found. trends® Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
- [50] Song, H. A., & Lee, S. Y. (2013, November). Hierarchical representation using NMF. In *International conference on neural information processing* (pp. 466-473). Springer, Berlin, Heidelberg.
- [51] Coursera-deeplearning.ai, <https://www.coursera.org/deeplearning-ai>, accessed:
- [52] Albelwi, S., & Mahmood, A. (2017). A framework for designing the architectures of deep convolutional neural networks. *Entropy*, 19(6), 242.
- [53] CS231n Convolutional Neural Networks for Visual recognition, <http://cs231n.github.io/convolutional-networks/>, accessed: 2019-07-03.
- [54] DERİNE DAHA DERİNE: Evrişimli Sinir Ağları, <https://medium.com/@ayyucekizrak/deri%CC%87ne-daha-deri%CC%87ne-evri% C5%9Fimli-sinir-a% C4%9Flar% C4%B1-2813a2c8b2a9>, accessed: 2019-07-03.
- [55] O. Cakoglu, <https://www.flickr.com/photos/ondercakoglu/4534409162/in/photostream/>, accessed: 2017-08-30.
- [56] Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3, https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088, accessed: 2019-05-31.
- [57] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*. [71] Viola, Paul & Jones, Michael. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. *IEEE Conf Comput Vis Pattern Recognit.* 1. I-511. 10.1109/CVPR.2001.990517.
- [58] R. Y., S. T., P. B., and K. B, “Segmentation and object recognition using edge detection techniques,” *International Journal of Computer Science & Information Technology (IJCSIT)*, vol. 2, no. 6, pp. 153–161, 2010.

- [59] The Viola/Jones Face Detector (2001), <https://www.cs.ubc.ca/~lowe/425/slides/13-ViolaJones.pdf>, accessed: 2019-05-31.
- [60] Nesne Tespiti ve Nesne Tanıma Süreçleri, <http://mesutpiskin.com/blog/nesne-tespiti-ve-nesne-tanima.html>, accessed: 2019-05-31.
- [61] Solak, S., & ALTINIŞIK, U. (2018). Görüntü işleme teknikleri ve kümeleme yöntemleri kullanılarak fındık meyvesinin tespit ve sınıflandırılması. Sakarya University Journal of Science, 22(1), 56-65.
- [62] Denoising results obtained with simulated noisy images, <https://www.io.csic.es/PagsPers/JPortilla/image-processing/bls-gsm/61-denoising-results-obtained-with-simulated-noisy-images->, accessed: 2019-05-31.
- [63] Rasheed, N. A., & Nados, W. L. (2018). Object Segmentation from Background of 2D Image. Journal of University of Babylon, 26(5), 204-215.
- [64] Rasheed, N. A., & Nados, W. L. (2018). Object Segmentation from Background of 2D Image. Journal of University of Babylon, 26(5), 204-215.
- [65] Python | Foreground Extraction in an Image using Grabcut Algorithm, <https://www.geeksforgeeks.org/python-foreground-extraction-in-an-image-using-grabcut-algorithm/>, accessed: 2019-05-31.
- [66] Rasheed, N. A., & Nados, W. L. (2018). Object Segmentation from Background of 2D Image. Journal of University of Babylon, 26(5), 204-215.
- [67] Reddi, S. S., Rudin, S. F., & Keshavan, H. R. (1984). An optimal multiple threshold scheme for image segmentation. IEEE Transactions on Systems, Man, and Cybernetics, (4), 661-665.
- [68] Zambanini, S., & Kampel, M. (2008). Segmentation of ancient coins based on local entropy and gray value range. In Proc. of the 13th Computer Vision Winter Workshop (pp. 9-16).

- [69] Prasantha, H.S., Dr.Shashidhara, H. L., Dr. Murthy, K.N.B., Madhavi Lata. G., “Medical image segmentation,” *International Journal on Computer Science and Engineering*, vol. 2, no. 4, pp. 1209–1218, 2010.
- [70] Rother, C., Kolmogorov, V., & Blake, A. (2004, August). Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)* (Vol. 23, No. 3, pp. 309-314). ACM.
- [71] 3D Tasarım 3D Modelleme Nedir, <https://www.tridi.com.tr/3d-tasarim-3d-modelleme-nedir>, accessed: 2019-05-31.
- [72] Prof Agu, Emmanuel, “Computer Graphics (CS 543) Lecture 4 (Part 3): Viewing & Camera Control”.
- [73] Joseph D'Amelio, “Perspective Drawing handbook”.
- [74] Cengiz, Ş., & Bülbül, A. (2017). 3D Fountain Modeling from Single Image. *Eurasia Graphics*.
- [75] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9), 1627-1645.
- [76] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
- [77] Marsh, M., “Implementing the ‘grabcut’ segmentation technique as a plugin for the gimp,” <http://www.cs.ru.ac.za/research/g02m1682/>, accessed: 2017-09-29.
- [78] Cali, E., <http://www.panoramio.com/photo/4983965>, accessed: 2017-08- 30.
- [79] Weber, M., Welling, M., & Perona, P. (2000, June). Unsupervised learning of models for recognition. In *European conference on computer vision* (pp. 18-32). Springer, Berlin, Heidelberg.

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname : Şeyma CENGİZ
Date of Birth : 1992
E-mail : seyma.cengiz06@gmail.com.tr

EDUCATION

High School : Yıldırım Beyazıt Anatolian High School / Ankara (2006-2010)
(82.11/100)
Bachelor : Ankara Yıldırım Beyazıt University / (2011-2016)
(3.17/4.0)
Master Degree : Ankara Yıldırım Beyazıt University / (2016-continued)
(3.75/4.0)

WORK EXPERIENCE

AI R&D Engineer : Novelty Bilisim / (2018-continued)

Teaching Assistant for the course Human Computer Interaction

Ankara Yıldırım Beyazıt University / (2018- Spring Semester)

TOPICS OF INTEREST

- Artificial Intelligence
- Machine Learning
- Pattern Recognition
- Speech Recognition