

**T.C.**  
**BALIKESİR ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**  
**MAKİNA MÜHENDİSLİĞİ ANABİLİM DALI**



**ŞEKİL VE DOKU TABANLI YÖNTEMLERLE NESNELERİN  
BÖLÜTLENEREK EN UYGUN YAPITAŞLARI İLE DİZİLMESİ**

**ALP İMREK**

**DOKTORA TEZİ**

**Jüri Üyeleri :** Prof. Dr. Ali ORAL (Tez Danışmanı)  
Prof. Dr. Kadir ÇAVDAR  
Doç. Dr. Şevket GÜMÜŞTEKİN  
Doç. Dr. Alaaddin TOKTAŞ  
Doç. Dr. Sabri BIÇAKÇI

**BALIKESİR, TEMMUZ - 2021**

## **ETİK BEYAN**

Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak tarafımda hazırlanan “**Şekil ve Doku Tabanlı Yöntemlerle Nesnelerin Bölütlenerek En Uygun Yapı Taşları İle Dizilmesi**” başlıklı tezde;

- Tüm bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Kullanılan veriler ve sonuçlarda herhangi bir değişiklik yapmadığımı,
- Tüm bilgi ve sonuçları bilimsel araştırma ve etik ilkelere uygun şekilde sunduğumu,
- Yararlandığım eserlere atıfta bulunarak kaynak gösterdiğimi,

beyan eder, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ederim.

**Alp İMREK**

## ÖZET

### ŞEKİL VE DOKU TABANLI YÖNTEMLERLE NESNELERİN BÖLÜTLENEREK EN UYGUN YAPI TAŞLARI İLE DİZİLMESİ

DOKTORA TEZİ

ALP İMREK

BALIKESİR ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNA MÜHENDİSLİĞİ ANABİLİM DALI

(TEZ DANIŞMANI: PROF. DR. ALİ ORAL)

BALIKESİR, TEMMUZ - 2021

Bu çalışmada, mozaik üretiminde kullanılan, kare şeklinde kesilmiş mermer, traverten gibi doğal taşların çeşitli parlaklık, renk ve dokuya sahip yüzey görüntüleri kullanılarak, istenen bir sayısal görüntünün, mermer mozaik olarak elde edilmesi için gerekli teknikler ve yöntemler belirlenmiş ve esnek üretim sisteminin gerçekleştirilmesine yönelik aşamalar kaydedilmiştir.

Bir görüntünün veya bir desenin mozaikini dizen otomasyon sistemi oluşturmak için yapılan çalışmalar oldukça yenidir. Literatürde, mozaik oluşturulan görüntü ile bu görüntüye yerleştirilen mermer karolar arasındaki hatayı optimize eden ilk çalışma gerçekleştirilmiştir. Kombinatoriyal optimizasyon algoritması “Macar yöntemi” kullanılmıştır.

Mermer parçalarını çalışma düzleminde konumunu bulmak için görüntü işleme yöntemlerinden yararlanılmıştır.

Mermer karoların çalışma düzleminde buldukları noktadan alınıp yerleştirilmesi gereken noktaya taşınması ve döndürülerek yerleştirilmesi için tut ve yerleştir işlemlerinde yaygın olarak kullanılan delta robot prototipi tasarlanmış, mekanizmanın AC servo motorlarla hareket kontrolü ve Profinet üzerinden PC-PLC haberleşmesi gerçekleştirilmiştir. Delta robotun ileri ve ters kinematik denklemleri kullanılarak robot son işlevcisi istenen konumlara hareket ettirilmiştir.

Görüntü ile karoların görüntüleri arasındaki benzerliğin ölçülmesi için en hızlı ve etkin yöntemin Öklid uzaklık olduğu sonucuna varılmıştır.

Bilgisayarlı görü yöntemlerinden, evrişimli nöron ağları ile nesne algılayan yöntemler kullanarak, görüntüde insan yüzü, el, beden ve diğer nesnelere tespit eden, bu bölgelere yapılacak karo atamalarında hatayı küçük tutarak, nesne içermeyen bölgelere nazaran öncelik veren bir yöntem sunulmuştur. El, yüz, beden algılamak için OpenPose, diğer nesnelere algılamak için YOLO nesne detektörleri kullanılmıştır.

Farklı boyutlarda 8, 16, 32 piksel kare karo parçaları ile mozaik dizen bir yöntem geliştirilmiştir.

**ANAHTAR KELİMELER:** Mozaik dizme, görüntü işleme, bilgisayarlı görü, evrişimli nöron ağları ile nesne algılama, delta robot, atama problemi, Macar yöntemi

Bilim Kodları : 91428, 91438, 91404, 91423

Sayfa Sayısı : 130

## **ABSTRACT**

### **SEGMENTING OBJECTS WITH SHAPE AND TEXTURE BASED METHODS AND TILING WITH THE MOST SUITABLE BUILDING BLOCKS**

**PH.D THESIS**

**ALP İMREK**

**BALIKESİR UNIVERSITY INSTITUTE OF SCIENCE**

**MECHANICAL ENGINEERING**

**(SUPERVISOR: PROF.DR. ALİ ORAL )**

**BALIKESİR, JULY - 2021**

In this study, the necessary techniques and methods were determined for obtaining the marble mosaic of desired image by using surface images with various color and texture of square cut natural stones such as marble and travertine used in mosaic production and progress has been made towards realization of flexible manufacturing system.

The studies to create an automation system that tiles the mosaic of an image or a pattern is quite new. In the literature, the first study was carried out to optimize the total error between the image to be mosaicked and the image of tiles to be placed. Combinatorial optimization algorithm “Hungarian method” was used.

Image processing methods have been used to locate marble tiles on the working plane.

A prototype of the delta robot is designed for pick and place operations. The motion control of the mechanism with AC servo motors were performed and a software is developed for the PC to PLC communication through Profinet.

In order to measure the similarity between the image and the images of the tiles, it has been concluded that the fastest and most efficient method is the Euclidean distance.

A Method has been presented which detects regions with human and other objects in the image, keeps the error smaller in the tile assignments to these regions and prioritize than the object-free regions. OpenPose was used to detect body parts. YOLO was used to detect objects.

A method has been developed to tile mosaics with square tile pieces of different sizes.

**KEYWORDS:** Mosaic tiling, image processing, computer vision, object detection with convolutional neural networks, delta robot, assignment problem, Hungarian algorithm  
Science Codes: 91428, 91438, 91404, 91423 Page Number: 130

# İÇİNDEKİLER

## Sayfa

<b>ÖZET</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>İÇİNDEKİLER</b> .....	<b>iii</b>
<b>ŞEKİL LİSTESİ</b> .....	<b>v</b>
<b>TABLO LİSTESİ</b> .....	<b>viii</b>
<b>SEMBOL LİSTESİ</b> .....	<b>ix</b>
<b>KISALTMALAR LİSTESİ</b> .....	<b>x</b>
<b>ÖNSÖZ</b> .....	<b>xi</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
1.1 Tezin Önemi .....	2
1.2 Amaç ve Kapsam .....	3
1.3 Motivasyon.....	3
1.4 Mozaik Yapım Teknikleri ve Kullanılan Materyaller.....	4
1.4.1 Mozaik Materyalleri.....	4
1.4.2 Mozaik Uygulama Teknikleri .....	5
1.4.3 Mozaik Desen Tasarımı ve Üretimi .....	5
1.4.3.1 Sanatsal Mozaik .....	5
1.4.3.2 Sanayi Mozaiği.....	7
1.4.4 Mozaik Karolarının Yerleştirme Teknikleri.....	10
<b>2. LİTERATÜR ÖZETİ</b> .....	<b>12</b>
2.1 Giriş.....	12
2.2 Mozaik Dizme Otomasyonu İle İlgili Çalışmalar .....	12
2.3 Dijital Mozaik Teknikleri ile ilgili Çalışmalar.....	20
2.3.1 Görüntünün Mozaik Görüntüsü Haline Getirilmesi.....	20
2.3.2 Foto Mozaikler .....	20
2.4 Konunun Literatürdeki Önemi .....	21
<b>3. MATERYAL VE YÖNTEM</b> .....	<b>22</b>
3.1 Giriş.....	22
3.2 Karoların Algılanması .....	25
3.3 Görüntünün Parçalara Bölünmesi .....	28
3.3.1 Farklı Ölçülerde Karolar için Görüntünün Parçalara Bölünmesi.....	29
3.4 İki Görüntü Parçası Arasında Benzerliğin Ölçülmesi.....	30
3.4.1 Benzerlik ve Uzaklık.....	30
3.5 Atama .....	34
3.5.1 Macar Yöntemi.....	35
3.5.2 Maliyet Matrisi ile Macar Yöntemi.....	36
3.5.3 İki Parçalı Graf (Bipartite Graf) Kullanarak Macar Yöntemi .....	37
3.6 Farklı Ölçülerde Karoların (8, 16, 32 piksel), Görüntü Parçalarına Atanması .....	41
3.7 Nesne İçeren Görüntü Parçalarının Etiketlenmesi .....	47
3.7.1 Derin Öğrenme Temelli Nesne Detektörleri .....	48
3.7.2 YOLO, Nesne Detektörü.....	49
3.7.2.1 YOLO V3 Ağının Girişi.....	50

3.7.2.2	YOLO V3 Ağının Çıkışı .....	50
3.7.2.3	YOLO V3 Ağının Çıkışının Kullanılması.....	52
3.7.2.4	YOLO İle Nesne İçeren Görüntü Parçalarının Etiketlenmesi .....	54
3.7.3	Dlib İle Yüz Anahtar Noktası İçeren Görüntü Parçalarının Etiketlenmesi.....	56
3.7.4	OpenPose Beden Anahtar Noktalarının Bulunması.....	58
3.7.4.1	Çok Sayıda İnsan Pozu Tahmin Modeli.....	60
3.7.4.2	Beden Anahtar Noktası İçeren Görüntü Parçalarının Etiketlenmesi.....	63
3.7.4.3	Yüz Anahtar Noktası İçeren Görüntü Parçalarının Etiketlenmesi .....	64
3.7.4.4	El Anahtar Noktası İçeren Görüntü Parçalarının Etiketlenmesi .....	66
3.7.5	Nesne İçeren Görüntü Parçalarının Etiketlenmesi Yöntemi .....	67
3.8	Delta Robot .....	73
3.8.1	Delta Robot İleri ve Ters Kinematik Denklemleri.....	75
3.8.2	Ters Kinematik.....	76
3.8.3	İleri Kinematik .....	80
3.8.4	Ölçülerin Belirlenmesi .....	84
3.8.5	Ölçülerin Sağladığı Çalışma Bölgesinin Çıkarılması .....	85
3.8.6	Hareket Kontrol.....	86
3.8.7	AC-Servo Motor, Sürücü ve Redüktör Belirlenmesi:.....	86
3.8.8	PLC Seçimi: .....	88
3.8.9	Motor Sürücü ve PLC Arasında Haberleşme.....	88
3.8.10	Dönme Eksen Kontrolü .....	91
3.8.11	PLC ile PC Arasında Haberleşme .....	94
<b>4.</b>	<b>UYGULAMA.....</b>	<b>97</b>
4.1	Aynı Ölçülerde Karolar ile Uygulama .....	97
4.1.1	Karo sayısının ve karo ölçüsünün etkisi.....	100
4.2	Farklı Ölçülerde Karolar ile Uygulama .....	105
4.3	Nesne İçeren Görüntü Parçalarına Öncelik Verilmesi.....	107
4.3.1	Uygulama 1 .....	107
4.3.2	Uygulama 2 .....	108
4.3.3	Uygulama 3 .....	111
<b>5.</b>	<b>SONUÇ ve ÖNERİLER .....</b>	<b>114</b>
5.1	Bulgular.....	114
5.1.1	En Benzer Karoyu Bulmak İçin Çeşitli Uzaklık Hesaplama Yöntemleri ve Yapısal Benzerlik Yöntemi: .....	114
5.1.2	Benzer Karoyu Bulmak İçin Görüntü Özellik Tanımlayıcıları:.....	116
5.1.3	Dizme 118	
5.1.4	Görüntünün Süper Pikseller İle Bölütlenerek Paledyen Görüntü Parçaları Elde Edilmesi .....	119
5.1.5	Görüntüyü Parçalayan Satır ve Sütunları Kaydırarak Grid Hücreleri İçinde Standart Sapmayı Azaltmak .....	120
5.1.6	32,16,8 piksel Karoların Bulunduğu Durumda, Desen Görüntüsünün Bölünmesi..	121
5.2	Sonuçlar: .....	122
5.3	Öneriler .....	123
<b>6.</b>	<b>KAYNAKLAR .....</b>	<b>124</b>
	<b>ÖZGEÇMİŞ.....</b>	<b>130</b>

# ŞEKİL LİSTESİ

## Sayfa

Şekil 1.1: Mermer tessera.....	1
Şekil 1.2: Bir simit (torus), tekrarlayan dörtgenler matrisi ile kaplanabilir [1]. .....	2
Şekil 1.3: Bedri Rahmi Eyüpoğlu'nun İstanbul adlı mozaik eseri,1965, İMÇ İstanbul [2]..	3
Şekil 1.4: Şekline göre a) Düzgün geometrik şekle sahip karolar b) Paledyen karolar [3].	4
Şekil 2.1: Scara robot ile mozaik dizgi otomasyonu [8]. .....	12
Şekil 2.2: Scara robot ile mozaik dizgi otomasyonu [9]. .....	13
Şekil 2.3: Ultimate Puma 500 robot cam mozaik dizeken [10]. .....	14
Şekil 2.4: Dört serbestlik dereceli Kartezyen mozaik dizgi robotu [11]. .....	15
Şekil 2.5: Sürekli bant cam mozaik dizme otomasyonu a) Genel görünüm. b) Sürgülü sistem ve röleler. c) Sıralama kutusu. d) Ana bant ve pnömatik sistem [13]. ..	16
Şekil 2.6: ABB 4400 robotu, seramik karolar ile Marilyn Monroe'nun mozaikini dizeken [15]. .....	17
Şekil 2.7: NAO robot ile zemin döşeme [16]. .....	17
Şekil 2.8: a) Görüntüdeki koyu bölgelere yerleştirilen karolar. b) Taş bırakma kapsülü [17]. .....	18
Şekil 2.9: O. Önal'ın mozaik çalışmaları [18]. .....	19
Şekil 3.1: Mermer levha görüntüleri .....	24
Şekil 3.2: Çalışma düzleminde algılanan karolar. ....	25
Şekil 3.3: Soldan sağa 60 mm, 40 mm, 20 mm, 10 mm ölçülerinde karolar [4]. .....	28
Şekil 3.4: Desen görüntüsüne uygulanmış 32x32 piksel düzenli ızgara. ....	29
Şekil 3.5: 32, 16, 8 piksel hücrelere sahip ızgara ile bölünmüş görüntü. ....	29
Şekil 3.6: Birbirine benzerliği hesaplanan solda karo, sağda görüntü parçası. ....	31
Şekil 3.7: Görüntü parçasına benzerliği hesaplanmak üzere üç kez 90° derece döndürülmüş karo. ....	32
Şekil 3.8: İç içe geçmiş iki paralel foreach döngüsü. ....	34
Şekil 3.9: İki parçalı Graf $G_{4,4}$ .....	38
Şekil 3.10: a) İki parçalı graf $G_{4,3}$ b) İki parçalı tam graf $K_{4,3}$ .....	38
Şekil 3.11: a) Düğüm Etiketleme. b) Eşitlik grafi. ....	39
Şekil 3.12: Tüm karoların aynı ölçüde olduğu durumda görüntü parçalarına atama. ....	41
Şekil 3.13: 32x32 hücrelerin, 32,16 ve 8 piksel karoların atanması. ....	43
Şekil 3.14: 32, 16 ve 8 piksel karo ve karo gruplarının yerleştirme hatalarının adlandırılması. ....	44
Şekil 3.15: Nesne detektörleri a) Sınırlayıcı kutu ile nesne algılama. b) Anahtar noktalar ile nesne algılama [39]. .....	49
Şekil 3.16: YOLO V3 (You Only Look Once- Version3) nesne detektörü boru hattının basitleştirilmiş blok diyagramı [41]. .....	50
Şekil 3.17: YOLO V3 ağının giriş ve çıkışı a) Ağın girişine uygulanan görüntünün, üç farklı ölçekte özellik çıkarımı için parçalara ayrılması ve b) her bir görüntü parçası için çıkarılan 255 adet özellik [42]. .....	51
Şekil 3.18: YOLO V3 ağının çıkışındaki her bir hücrenin derinliği [42]. .....	52
Şekil 3.19: YOLO V3 çıkış işlem adımları [42]. .....	52
Şekil 3.20: IOU değerinin hesaplanması [39]. .....	53
Şekil 3.21: YOLO V3 Nesnellik Puanlarının ifade ettiği sınır kutularına örnek [42]. .....	54
Şekil 3.22: Dlib yüz anahtar noktaları [44]. .....	57

Şekil 3.23: Çok sayıda insan pozu tahmini [46].	59
Şekil 3.24: Çok sayıda insanın, tüm beden poz tahmini ağının mimarisi [45].	60
Şekil 3.25: OpenPose tam beden girişine verilen iki insan görüntüsünün beden parçalarının gövdeye aitliklerinin belirlenmesi [45].	61
Şekil 3.26: COCO veri kümesi ile eğitilmiş OpenPose'nin 18 anahtar noktası.	63
Şekil 3.27: OpenPose yüzün anahtar noktaları [47].	65
Şekil 3.28: OpenPose el anahtar noktaları [48].	67
Şekil 3.29: Nesne içeren görüntü parçalarının etiketlenmesi.	69
Şekil 3.30: Görüntüye uygulanan OpenPose Çoklu Beden sonuçları.	71
Şekil 3.31: YOLO'nun algıladığı nesnelerin sınırlayıcı kutuları ve OpenPose nin algıladığı anahtar noktalar.	72
Şekil 3.32: a) Nesnelerin kalıp görüntüsü b) Kalıp görüntü ile kesişen 8 piksel görüntü parçaları. c) 8x8 piksel boyutlarında 19.200 adet görüntü parçasının yerleşimi.	73
Şekil 3.33: Delta Robot Eksen sayısı 4, Çalışma Alanı $\phi 1200 \times 250$ mm (silindir) yük 3 kg, dönel tork 0.5 Nm.	75
Şekil 3.34: İleri ve Ters Kinematik problemin şematik gösterimi.	76
Şekil 3.35: Delta Robot Parametreleri [53].	77
Şekil 3.36: Üst kol ve alt kolun çizdiği çemberler ve kesişim noktaları [53].	78
Şekil 3.37: Uç işlevci koordinatları [53].	78
Şekil 3.38: Bağlantı noktaları [53].	81
Şekil 3.39: J1', J2', J3' noktalarının x-y düzlemine izdüşümü [53].	82
Şekil 3.40: Delta robot prototipinin ölçüleri.	84
Şekil 3.41: Ters ve ileri kinematik denklemler arasındaki hata	85
Şekil 3.42: Çalışma bölgesi, 50mm aralıklarla örneklendirilmiştir.	86
Şekil 3.43: Kuvvet kolu, üst kol.	87
Şekil 3.44: Bir numaralı eksen darbe ve yön çıkışı.	89
Şekil 3.45: $\Theta_1$ , $\Theta_2$ , $\Theta_3$ mekanik ayarları.	89
Şekil 3.46: $\Theta_1$ , $\Theta_2$ , $\Theta_3$ dinamik ayarları.	90
Şekil 3.47: Her bir eksen için hareket kontrol komutları.	91
Şekil 3.48: Dönme eksenini darbe ve yön çıkışı.	92
Şekil 3.49: $\Theta$ dönme eksenini mekanik ayarları.	92
Şekil 3.50: $\Theta$ Dönme eksenini dinamik ayarları.	93
Şekil 3.51: $\Theta$ Dönme eksenini göreceli hareket kontrol blokları.	93
Şekil 3.52: TCP/IP haberleşme ayarları.	94
Şekil 3.53: Delta robotun PLC üzerinden PC ile kontrolü.	96
Şekil 3.54: Mozaik dizmek için dört serbestlik dereceli tut ve yerleştir robotu, delta robot prototipi.	96
Şekil 4.1: 32 piksel, 3.536 adet karo ataması ile oluşturulan mozaik (%15 ölçekli).	97
Şekil 4.2: 16 piksel, 14.144 adet karo ataması ile oluşturulan mozaik (%18 ölçekli).	98
Şekil 4.3: 8 piksel, 33.920 adet parçanın ataması ile elde edilen mozaik (%23 ölçekli).	99
Şekil 4.4: Karo sayısının ve karo ölçüsünün dizme hatası üzerine etkisi	101
Şekil 4.5: Orijinal Görüntü [59]	101
Şekil 4.6: 16 piksel ölçülerinde 22.000 karodan 8.036 adeti kullanılarak elde edilen mozaik	102
Şekil 4.7: Veri kümesindeki karo sayısının mozaik kalitesine etkisi (derzsiz).	103
Şekil 4.8: Veri kümesindeki karo sayısının mozaik kalitesine etkisi (derzli).	104
Şekil 4.9: Farklı ölçülerde karoların görüntü parçalarına atanması, derzli yerleşim.	105
Şekil 4.10: Farklı ölçülerde karoların görüntü parçalarına atanması 4176 adet 8 piksel, 5956 adet 16 piksel karonun yerleşimi derz olmaksızın (%51 ölçekli).	106



<b>Şekil 4.11:</b> YOLO ve OpenPose ile algılama ve kalıp görüntü. ....	107
<b>Şekil 4.12:</b> a) Nesne içeren görüntü parçalarını etiketleyen yöntem ile elde edilen mozaik. b) Aynı karoların, aynı görüntüye sadece Macar yöntemi ile atanması durumunda sol göz bölgesi. ....	108
<b>Şekil 4.13:</b> YOLO ve OpenPose ile algılama ve kalıp görüntü. ....	109
<b>Şekil 4.14:</b> a) 21.041 adet 16 piksel karodan 5.040 adet karonun atanması ile elde edilen mozaik. b) Aynı karoların, aynı görüntüye sadece Macar yöntemi ile atanması durumunda sağ göz bölgesi ve c) Kravat.....	110
<b>Şekil 4.15:</b> a) Orijinal görüntüden [58] YOLO ile algılanan kişi b) Dlib ile algılanan yüz anahtar noktaları ve çerçevesi c) Openpose ile algılanan yüz anahtar noktaları d) Nesnenin kalıp görüntüsü e) Kalıp görüntü ile kesişen 32 piksel görüntü parçaları. ....	112
<b>Şekil 4.16:</b> a) Sadece Macar yöntemiyle atama b) Orijinal Görüntü [58]. c) Nesne içeren görüntü parçaları etiketlenerek atama. d) Macar yöntemi ile elde edilen sonucun büyütülmüş göz bölgesi. e) Nesnelere etiketlenerek elde edilen sonucun büyütülmüş göz bölgesi.....	113
<b>Şekil 5.1:</b> A ve B vektörleri arasında kosinüs benzerliği $\Theta$ . ....	115
<b>Şekil 5.2:</b> Emgu CV SIFT Features örnek uygulama.....	117
<b>Şekil 5.3:</b> Superpixel SLIC m=2 için bölütleme sonucu. ....	120
<b>Şekil 5.4:</b> Soldan sağa; 1. Satır: yatay kaydır, düşey kaydır, opus tessellatum yatay,.....	120
<b>Şekil 5.5:</b> Bağlı standart sapmanın büyük olduğu bölgelerde görüntü parçasının küçültülerek değişimin azaltılması. ....	121

## TABLO LİSTESİ

	<u>Sayfa</u>
<b>Tablo 3.1:</b> Bilgisayar ortamında oluşturulmuş karo bilgileri. ....	24
<b>Tablo 3.2:</b> Bir mozaik üreticisinden alınan çeşitli karolar.....	25
<b>Tablo 3.3:</b> YOLO'nun algıladığı nesnelerin listesi .....	55
<b>Tablo 3.4:</b> Dlib yüz anahtar noktaları ve öğeleri. ....	58
<b>Tablo 3.5:</b> İnsan bedeni anahtar noktalarının etiketlenmesi. ....	64
<b>Tablo 3.6:</b> Openpose yüz anahtar noktaları ve yüz öğeleri. ....	66
<b>Tablo 3.7:</b> OpenPose el anahtar noktaları ve öğeleri. ....	67
<b>Tablo 3.8:</b> Bir mozaik üreticisinden alınan çeşitli örneklerin ölçü ve ağırlıkları. ....	86
<b>Tablo 3.9:</b> Delta robot bileşenlerinin ağırlıkları. ....	87
<b>Tablo 3.10:</b> Tampon (buffer) 25 baytlık paketin içeriği. ....	95
<b>Tablo 4.1:</b> Karo ölçüsünün ve karo sayısının mozaik dizme hatasına etkisi .....	100

## SEMBOL LİSTESİ

$Açı[i,j]$	: i. karonun j. görüntü parçasına kaç derece döndürülerek yerleştirileceğini tutan matris
$c$	: Renk uzayı kanal sayısı
$(C_x, C_y)$	: Karonun kütle merkezi
$D[i,j]$	: i. karonun, j. görüntü parçasına uzaklığını gösteren uzaklık (maliyet) matrisi
$d(X_i, X_j)$	: $X_i$ ve $X_j$ vektörleri arasındaki uzaklık
$f$	: Delta robot üst üçgenin bir kenarı
$e$	: Delta robot alt üçgen bir kenarı
$e_{ij}$	: i. karonun j. görüntü parçasına yerleştirme hatası
$e_1, e_2, e_3$	: Görüntüyü parçalayan ızgaranın 32 piksellik bir hücreğine, sırasıyla 32, 16 ve 8 piksel katmanlarda atanan mermer karoların toplam yerleştirme hatalarıdır
$ee[i]$	: 16x16 piksel karoların her birinin yerleştirme hatası $i \in \{0,1,2,3\}$
$eee[i]$	: 8x8 piksel karoların 4erli yerleştirme hataları toplamları $i \in \{0,1,2,3\}$
$eniyeEş[i]$	: i. karonun hangi görüntü parçasına atandığını veren dizi. Macar Alg. çıkışı
$h$	: Görüntünün yüksekliği
$I^2$	: Görüntü Parçası
<b>Karo</b>	: Karo parçalarının özelliklerinin tutulduğu yazılım nesnesi
<b>konturNoktaDizi(x,y)</b>	: Bir nesnenin kenar noktaları dizisi
<b>Liste&lt;Karo&gt;</b>	: Karolardan oluşan küme, liste.
$m_{ji}$	: Görüntünün j,i. momenti
$M_d$	: Döndürme momenti (Nm)
<b>MSE</b>	: Ortalama karasel hata (Mean Square Error)
$p_j$	: Nesne veya arka plan olmalarına göre, görüntü parçalarına atanan etiket.
$\mathbb{R}^2$	: $\mathbb{R}$ Gerçek sayılar kümesi olmak üzere $\mathbb{R}^2 = \mathbb{R} \times \mathbb{R} = \{ (x,y) \mid x,y \in \mathbb{R} \}$ . $\mathbb{R}$ nin kendisiyle Kartezyen çarpımı. $\mathbb{R}^2$ kümesinin öğeleri $(x,y)$ biçimindedir
$re$	: Delta robot alt kol uzunluğu
$rf$	: Delta robot üst kol uzunluğu
$saye_1, saye_2, saye_3$	: Görüntüyü parçalayan ızgaranın 32 piksellik bir hücreğine, sırasıyla 32,16,8 piksel katmanlarda atanan mermer karoların sayısıdır
$S(X_i, X_j)$	: $X_i$ ve $X_j$ vektörleri arasında benzerlik
$sK$	: Karo Sayısı
$sD$	: Görüntü parçası sayısı
$w$	: Görüntünün genişliği
$x_{ij}$	: 0,1 değerleri alan, i. kişinin j. göreve atamasını gösteren değişken
$(x,y,w,h)$	: Bir görüntü bölgesinin sol üst kenar noktası $(x,y)$ , eni, boyu
$z$	: Amaç fonksiyon çıktısı
$\Theta_1, \Theta_2, \Theta_3$	: Delta robot motor açıları
$\mu$	: Ortalama, görüntünün bir renk kanalının ortalaması
$\sigma$	: Standart sapma, görüntünün bir renk kanalının standart sapması
$\frac{\sigma}{\mu}$	: Bağlı standart sapma

## KISALTMALAR LİSTESİ

<b>AP</b>	: Atama Problemi, Assignment Problem
<b>API</b>	: Application Programming Interface, Uygulama Programlama Arayüzü
<b>Arcmin</b>	: arcminute, 1°/60, 1drecenin 60 da biri
<b>CAD</b>	: Computer Aided Design, Bilgisayar Destekli Tasarım
<b>CNN</b>	: Convolutional Neural Network, Evrişimli Nöron Ağı
<b>DNN</b>	: Deep Neural Network, Derin Nöron Ağı
<b>COCO</b>	: Büyük ölçekli nesne tespit, bölütleme ve etiketleme veri kümesi
<b>CPU</b>	: Central Processing Unit, Merkezi İşlem Birimi
<b>Dlib</b>	: Makine öğrenmesi, yapay zeka, görüntü işleme vb. içeren genel amaçlı yazılım kütüphanesi,
<b>dpi</b>	: Dot per inch, inç başına nokta sayısı. Görüntü çözünürlüğü ölçüsü.
<b>EmguCV</b>	: OpenCV kütüphanesinin, Csharp dili ile kullanılabilmesi için sarmalanmış hali, yarı ticari görüntü işleme, bilgisayarlı görü yazılım kütüphanesi.
<b>Full HD</b>	: 1920x1080 piksel çözünürlük
<b>GPU</b>	: Graphics Processing Unit, Grafik İşlemci Birimi
<b>HOG</b>	: Histogram of Oriented Gradient, Yönlü Gradyanların Histogramı. Doku özellik çıkartma yöntemi
<b>IOU</b>	: Intersection Over Union, Kesişimin Birleşime Oranı
<b>JPEG</b>	: Joint Photographic Experts Group tarafından standartlaştırılmış bir kayıplı sayısal görüntü sıkıştırma yöntemi
<b>k-NN</b>	: k-Nearest Neighbors Algorithm, k En Yakın Komşu sınıflama yöntemi
<b>kmt</b>	: Komut
<b>Lab</b>	: L: ışık, a: yeşil-kırmızı, b: mavi-sarı temsil eden renk uzayı
<b>MC</b>	: Motion Control, Hareket Kontrol
<b>MPII</b>	: Human Pose Dataset, İnsan pozu veri kümesi
<b>OpenCV</b>	: Gerçek zamanlı bilgisayarlı görü ve görüntü işleme uygulamaları geliştirmek için yazılım kütüphanesi
<b>OpenPose</b>	: İlk gerçek zamanlı çok sayıda insan gövdesi, yüz, el ve ayak algılayıcısı
<b>PLC</b>	: Programlanabilir Lojik Denetleyici
<b>PTO</b>	: Pulse Train Output, Darbe Treni Çıkış, darbe sayısı ile motor kontrolünde kullanılan kare dalga formunda sinyal
<b>Pulse/Dir</b>	: PLC den servo motor sürücüsüne gönderilen motor hızı ve yönü sinyalleri
<b>PVC</b>	: Polivinil Klorür, plastik polimer
<b>R-CNN</b>	: Regional CNN, Bölgesel Evrişimli Nöron Ağı
<b>RGB</b>	: Kırmızı, yeşil, mavi renk uzayı
<b>rpm</b>	: Revolutions per minute, dakikada devir sayısı
<b>SCARA</b>	: Selective Compliance Articulated Robot Arm, Seçici Uyumluluk Eklemlili Robot Kolu
<b>TCP/IP</b>	: Transmission Control Protocol/Internet Protocol. İletim Kontrol ve internet Protokolleri
<b>UV</b>	: Ultraviyole, morötesi
<b>VGG</b>	: Oxford University Visual Geometri Group, Oxford Üniversitesi Görsel Geometri Grubu

## **ÖNSÖZ**

Yaşadığımız topraklarda derin geçmişe sahip, sanatsal, kültürel, teknik, insan görsel algısı ve bilgisayarlı görü alanlarında bilgi birikimi gerektiren mozaik dizme otomasyonu alanında öncü araştırmalara sahip, danışmanım sayın Prof. Dr. Ali ORAL'a bu alanda bana çalışma olanağı sunduğu, desteklediği ve motive ettiği ve çalışmamızı bitiş çizgisine kadar taşıdığı için çok teşekkür ederim. Uzakta olsa da her zaman yakınlığını hissettiğim bilgi ve deneyimleri ile çalışmamızı her zaman destekleyen Doç. Dr. Şevket GÜMÜŞTEKİN'e teşekkür ederim. Değerli tecrübelerini paylaşan Dr. Öğr. Üyesi Nedim GERGER'e teşekkürlerimi sunarım. Doç. Dr. Sabri BIÇAKÇI'ya değerli katkılarından dolayı çok çok teşekkür ederim.

**Balıkesir, 2021**

**Alp İMREK**

## 1. GİRİŞ

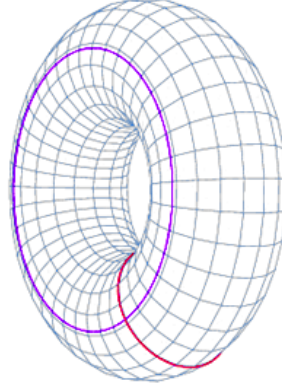
İnsan çevresiyle etkileşime girerek ona şekil vermektedir. Ortamdaki basit nesnelere yapı taşı olarak kullanarak, ihtiyaç duyduğu daha yüksek seviyeli yapılar oluşturmaktadır. Hayatı, doğayı yaşamak istediği biçime dönüştürmektedir. Sonuç bazen doğada karşılığı olmayan bir nesneyken bazen de doğayı taklit eder. Bu bazen bir inşaat gibi teknik, bazen de resim, heykel gibi sanatsal olabilmektedir. İnsan çalı, çit bitkilerini dahi kendi haline bırakmayıp budayarak onlara şekil vermektedir.

“Nesnelerin bölütlenerek en uygun yapı taşları ile dizilmesi” başlığı altında çalışma alanımız “Mozaik” tir. Çeşitli renklerdeki küçük doğal taş, cam ve seramik parçalarının (tessera) yan yana getirilmesi ile oluşturulan süs veya resimlere "mozaik" denir. İnsan eli ile üretilen 7000 yıllık bir geleneğe sahiptir. Latince tessera, mozaik yapmakta kullanılan küp şeklinde seramik, taş veya cam parçasıdır (Şekil 1.1). Erken antik dönemde renklendirilmiş doğal çakıl taşı kullanılmıştır. MÖ 2. yüzyılın sonlarında küp şeklinde kesilmiş kireç taşı ve mermer kullanılmıştır. Tessera döşeme işine tessellasyon (tessellate) denir. Tesselasyona, sanat tarihi boyunca antik mimariden modern sanatlara kadar rastlanabilir [1].



**Şekil 1.1:** Mermer tessera.

Matematikte bir yüzeyin kaplanması, karo adı verilen bir veya daha fazla geometrik şeklin birbirlerini örtmeden ve boşluk bırakmadan bir düzleme yerleştirilmesidir. Matematikte mozaikler daha yüksek boyutlara ve çeşitli geometrilere genelleştirilebilir (Şekil 1.2) [1].



**Şekil 1.2:** Bir simit (torus), tekrarlayan dörtgenler matrisi ile kaplanabilir [1].

Karo, Fransızca “carreau” dan gelen çeşitli malzemelerden yapılmış dört köşe döşeme taşına verilen isimdir [1]. Çalışmada düzlemi kaplayan geometrik şekiller "karo" olarak adlandırılacaktır.

### **1.1 Tezin Önemi**

Mozaik üretimi, kültürümüzün hem batıdan hem de doğudan bir parçası olarak, yakın zamana kadar daha sanatsal uygulanmıştır (Şekil 1.3). Tezin önemi, günümüzde otomatikleşmiş mozaik üretimini, görüntü işleme, bilgisayarlı görü ve yapay zeka gibi güncel teknolojiler ile bütünleştirerek, esnek üretim yöntemleri ile yeniden bir tarz ve sanat olarak sürdürülmesini ve geliştirilmesini desteklemesidir.



**Şekil 1.3:** Bedri Rahmi Eyüpoğlu'nun İstanbul adlı mozaik eseri,1965, İMÇ İstanbul [2].

## 1.2 Amaç ve Kapsam

Mermer mozaik sanayinde desen kalıbı kullanmaksızın nesnelere içeren görüntünün mozaikini dizebilecek, esnek üretim yapan, otomasyon sisteminin gerçekleştirilmesi için gerekli bilimsel ve teknik bilgi, yazılım ve donanımın oluşturulmasıdır.

## 1.3 Motivasyon

Batı Anadolu, mermer ocaklarının yaygın bulunduğu, işlendiği, nihai ürün haline getirildiği bir bölgedir. Marmara Bölgesinin adı zaten mermer anlamına gelmektedir. Mermer sanayi Balıkesir'in katma değer ürettiği, ihracat yaptığı önemli sektörlerden biridir. Mermer mozaikler, Balıkesir mermer sanayinin iç ve dış piyasalarda rağbet gören önemli ürünlerinden biridir.

Mozaik yüzyıllardır uygulanan bir sanat olmanın yanında, ocaktan çıkarılan mermerin işlenmesi sırasında ortaya çıkan kırıkların da kullanılarak, ekonomik fayda elde edilmesini sağlamaktadır.

El emeği ile gerçekleştirilen mermer mozaik üretiminde sağlanacak otomasyon, üretim maliyetlerini azaltırken, kaliteyi arttıracak, ürün gamı genişleyecek, firmaların üretim kapasitelerini ve rekabet gücünü arttıracaktır. Dolayısıyla bu alanda yapılacak çalışmaların bölge ekonomisine katkıları ve geri dönüşleri olacaktır.



#### 1.4 Mozaik Yapım Teknikleri ve Kullanılan Materyaller

Mozaigi oluşturan parçalar düzgün geometrik şekilli veya şekilsiz olabileceği gibi çakıl taşları gibi yuvarlak yüzeyle de olabilir. Bu parçalar daha sonra bir harç veya yapıştırıcı madde ile birbirine tutturularak sağlamlaştırılır. Mozaik resim veya süslemeler yapıların yer döşemelerinde, duvarlarında veya tavanlarında mimari dekorasyon amacıyla kullanılır.

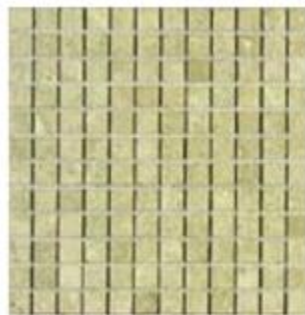
Bilinen en eski dekoratif sanatlardan olan mozaik, M.Ö. 7000 yılından itibaren Anadolu'da döşeme ve süsleme sanatı olarak uygulanmaya başlanmıştır.

Günümüzde mozaik sanatı, cam ve mermer materyalleri üzerinde yoğunlaşmış ve ticari bir boyut almıştır. Türkiye'nin mermer açısından çok zengin renk ve desen çeşitliliğine sahip olması, mozaik çalışmalarını kolaylaştırmaktadır. Ender bulunan renkler için cam veya seramik malzeme kullanılmaktadır [3].

##### 1.4.1 Mozaik Materyalleri

Mermer mozaik yapımında, mermer, kalker, traverten, granit, diyabaz, siyenit gibi blok ve levha halinde kesilebilen doğal taşlar kullanılmaktadır. Ayrıca 305x305x10 mm mermer karo üretiminde ortaya çıkan artıklar, üretilmek istenen ürünün kalitesine göre mozaik yapımında kullanılarak, ekonomiye katkı sağlanırken çevreye verilen zarar azaltılmaktadır [3].

Mozaik yapımında kullanılan karolar 10 mm kalınlığında olup kalibreden geçirilmiş, honlu, cilalı veya eskitilmiş yüzey şekillerine sahip taşlardır. Bu karolar, düzgün geometrik şekle sahip karolar ve paledyen karolar olarak iki grupta ele alınabilirler (Şekil 1.4) [3].



a)



b)

**Şekil 1.4:** Şekline göre a) Düzgün geometrik şekle sahip karolar b) Paledyen karolar [3].

### **1.4.2 Mozaik Uygulama Teknikleri**

Temel olarak bir mozaik yapımı desenin tasarlanması, çizilmesi, karoların döşenmesi ve yapıştırılması aşamalarından oluşur. Mermer parçalarını sabitlemek için alçı ya da yapıştırıcılar kullanılır. Mozaiklerin bir atölyede imalatı, bugün en çok kullanılan yöntemdir. Bu şekilde çok küçük boyutlardaki karolar ile en zor motifler işlenmekte ve daha sonra kalıcı olarak döşeneceği yere, bir mukavva, sunta, karton veya file üzerine taşınmaktadır. Bu teknik “dolaylı teknik” olarak adlandırılır.

Diğer yöntem ise “doğrudan teknik” tir. Mozaik döşenecek bölgeye dökülmüş renkli çimento üzerine, mozaik karoları doğrudan yerleştirilir. Bu sırada karoların belirli yönlerde, belirli eğimlerle yerleştirilmesi yansıtma özelliğini belirginleştirir ve desen empresyonizmi andıran bir canlılık kazanır. Roma ve Bizans döneminde dış mekanlarda uygulanan teknik bu olup, çeşitli büyüklüklerde ve elips şeklindeki “çakıl taşları” ile de uygulanmıştır [3].

### **1.4.3 Mozaik Desen Tasarımı ve Üretimi**

Mozaik uygulamalarında genel olarak iki çeşit üretim yapılmaktadır. Bunlardan biri "sanatsal mozaik", diğeri "sanayi mozaığı" olarak adlandırılır. Sanatsal mozaik uygulamalarında bir desene ihtiyaç duyulurken, sanayi mozaığı uygulamalarında bir desen olmaksızın karolar standart kalıplara dizilebilirler. Bu kalıplar genellikle bir mermer fayansı (305x305 mm) ölçüsündedir [3].

#### **1.4.3.1 Sanatsal Mozaik**

Sanatsal mozaik çalışmalarına atık çağlardan beri rastlayabiliyoruz. Bu mozaiklerde günlük yaşamdan alınmış konuların yanı sıra savaş mücadelelerine, insan, hayvan, doğa ve dinsel figürlere rastlanmaktadır. Uygulanacak desen şeffaf bir kağıt üzerine çizilir (Şekil 1.5). Mozaik tamamlandığında uygulanacağı yere taşımak için bir platform hazırlanır. Desen çizili olan kağıt bu platform üzerine yapıştırılır. Deseni içerecek ölçülerde bir fiberglas file kesilerek resmin üzerine yanlardan yapıştırılarak sabitlenir [3].



**Şekil 1.5:** Mozaïği dizilecek resmin kağıt üzerine çizilmesi ve karoların yerleştirilmesi [2].

Mozaik dizmeye başlanmadan önce kullanılacak karoların renkleri ve boyutlarının belirlenerek hazırlanması gereklidir. Karoların yerleştirilmesinin yanı sıra seçilen taşların görünümleri de sonucu doğrudan etkiler. İstenilen ebatlarda kesilerek hazırlanan karolar, renklerine göre ayrı ayrı kaplarda depolanmalıdır. Bazen sanatçı istediği ölçü ve renkteki taşı çekiç veya kerpeten ile kırarak şekillendirir [3].

Sanatsal mozaiklerde taşları istenilen şekil ve ölçülere küçültmek için elmas uçlu çift bıçaklı kerpeten, giyotin kerpeten, elmas uçlu mozaik çekici kullanılır (Şekil 1.6). Mermer, kalker ve traverten gibi şekil vermesi kolay yumuşak taşlar kullanılır [3].



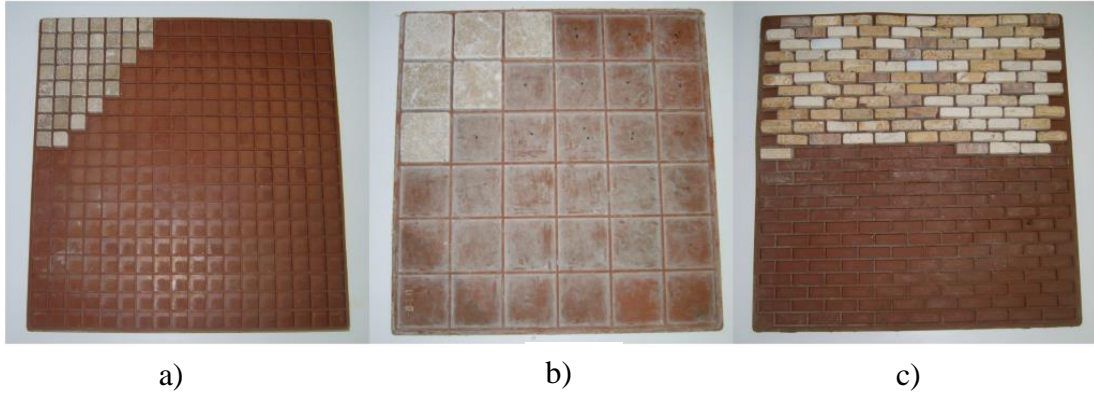
**Şekil 1.6:** Mozaik kesim aletleri [3].

#### **1.4.3.2 Sanayi Mozaığı**

Sanayi mozaikleri yer ve duvar kaplamaları için imal edilirler. Madalyon veya bordür deseni içerebilir ya da desensiz olabilirler. Standart olarak 10 mm kalınlığında ve 305x305 mm ölçülerinde üretilen mozaiklerden, yan yana döşenerek istenildiği kadar büyük alanlar kaplanabilir [4].

Sanayi mozaığı imalatında mermer plakalar çok diskli kesim makinalarında 10x10x10mm, 15x15x10 mm, 20x20x10 mm, 25x25x10 mm, 40x40x10 mm ve 48x48x10 mm 50x50x10 mm, 60x60x10 mm, 80x80x10 mm, 120x120x10 mm ebatlarında kare şeklinde veya bunlara benzer dörtgen ölçülerinde kesilerek plastik kalıplarda kullanılmak üzere hazırlanır. Eğer isteniyorsa, bu taşlar eskitme makinalarında eskitilerek kullanılır. 305x305 mm ebatlarındaki kalıp içerisine 15x15x10 mm ebatlı mozaik karolarından yaklaşık olarak 324 adet (Şekil 1.7(a)), 25x25x10 mm ebatlı mozaik karolarından 144 adet ve 48x48x10 mm ebatlı mozaik karolarından ise 36 adet kullanılmaktadır (Şekil 1.7(b)) [5]. 1m<sup>2</sup> alanı

kaplamak için gereken, 11 adet 305x305x10 mm ölçülerinde fayansın toplam ağırlığı 22-25 kg arasındadır [4].



**Şekil 1.7:** Hazır kalıplar a) 25x25x10 mm<sup>3</sup> opus regulatum. b) 48x48x10mm<sup>3</sup> opus regulatum. c) Opus tessellatum [5].

Desen içermeyen, düzenli ızgara veya tuğla deseni (opus regulatum veya tessellatum) önceden seçilen renklerde, hepsi aynı ölçüdeki karolar rastgele yerleştirilir. Mozaik, bordür desen içeren kalıplar da vardır. Bu kalıplara uygun, her birinin yeri belirli renk, şekil ve ölçülerde karoların önceden hazırlanması gerekir. Karolar Şekil 1.8(a)'da görüldüğü gibi kalıplara yerleştirildikten sonra Şekil 1.8(b)'de görüldüğü gibi fayans boyutunda kesilmiş fiberglas file, üzerine yapıştırıcı uygulanarak kalıba dizilmiş karoların üzerine yapıştırılır. Mozaik karolar arası boşluk (derz) 1-2 mm olarak uygulanmaktadır. Yapıştırma işleminden sonra kalıptan çıkarılan ve fileye yapıştırılmış olan mozaikler son bir defa gözden geçirildikten sonra ambalajlanır.



**Şekil 1.8:** a) Kalıplara dizilen karolar. b) Fileye yapıştırılmış karolar [6].

5 mm ile 50 mm arasında ölçülere sahip karolar sanatsal mozaiklerde, 10 mm ile 305 mm arasında ölçülere sahip karolar sanayi mozaiklerinde kullanılmaktadır [3].

Sanayi mozaiklerinde kullanılacak mozaik parçalarının elde edilmesi için çeşitli makineler kullanılmaktadır. Tecnema Breton Grup'un ECO LINE FS 400/N kesme ve profil oluşturma hattı (Şekil 1.9), doğal taş ve seramikleri sürekli kesmek için tasarlanmıştır [7].



**Şekil 1.9:** Kesme ve profil oluşturma hattı [7].

Mozaikler bir file üzerine yapıştırılarak bir arada tutulmaktadır. Sanayi mozaiklerinde fiberglas ağ üzerine sürülen tek bileşenli reçine ile çalışan yarı otomatik yapıştırma makinesi (Şekil 1.10) kullanılabilir. Yapıştırma işleminden sonra kurutma işleminin yapılması gereklidir. Bu amaçla kullanılan kurutma makinesi belirli bir dalga boyunda ultraviyole radyasyon ile çalışmaktadır. Ultraviyole lamba ve özel tasarım bir yansıtıcı UV/3 tek bileşenli seramik yapıştırıcısına uygulanır ve su emilimi olmaksızın kuvvetli yapıştırma sağlanır [7].

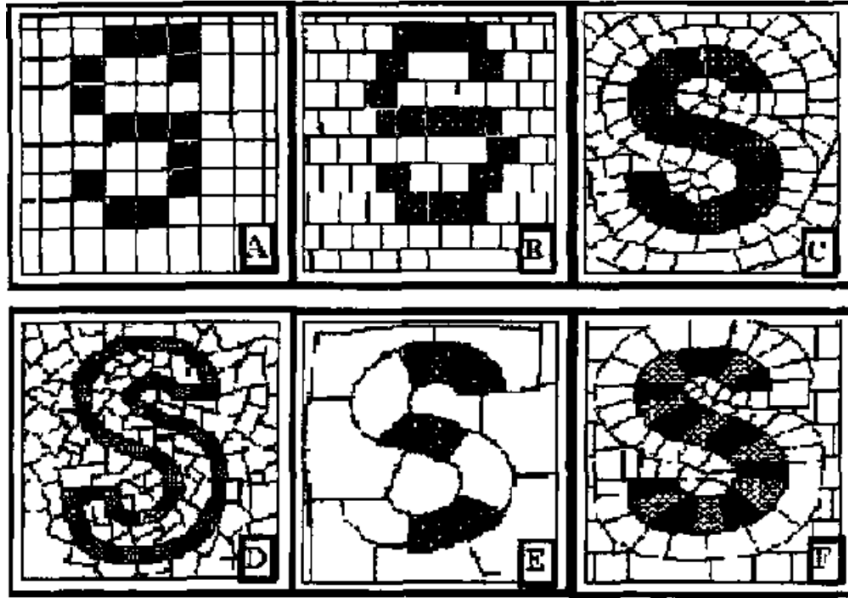


**Şekil 1.10:** Karo yapıştırma ve kurutma makinesi [7].

#### 1.4.4 Mozaik Karolarının Yerleştirme Teknikleri

Desenin oluşturulması sırasında karoların desen üzerine yerleştirilmesi belirli bir sistem ve düzene göre yapılır. Kullanılacak karoların geometrik şekli bazı durumlarda önem taşımakla beraber hem geometrik şekilli hem de paledyen şeklindeki karolar, bu amaçla kullanılabilir [4].

Mozaik yapımında antik çağlardan beri kullanılan geleneksel mozaik yerleştirme teknikleri günümüzde de geçerliliklerini korumuşlardır (Şekil 1.11). Geleneksel olarak bu döşeme tekniklerine Latince isimler verilmiştir. Bu teknikler şunlardır [4]:



**Şekil 1.11:** Mozaik taşlarının antik yerleştirme stilleri a) Opus regulatum. b) Opus tessellatum. c) Opus vermiculatum. d) Opus palladianum. e) Opus sectile/Florentine f) Opus classicum [4].

- a) Opus regulatum (ızgara deseni): Karolar düzenli bir ızgara oluşturacak şekilde döşenir. Aynı boyutlu karolardan meydana gelen desen, karoların birleşim hatları boyunca bir ızgara görüntüsü sunar. Genellikle bordürler ve büyük boyutlu mozaik uygulamalarında tercih edilen bir döşeme yöntemidir. Şekil 1.11(a)'da görülmektedir.

- b) Opus tessellatum (tuğla deseni): Karolar, tuğla duvar örülmesine benzer şekilde döşenir. Karoların yerleştirilmesi “opus regulatum” a benzer, farkı alttaki karo, üstteki karonun tam ortasına gelecek şekilde kaydırılarak yerleştirilir. Şekil 1.11(b)’de görülmektedir.
- c) Opus vermiculatum (solucan deseni): Bu tip mozaik döşeme tekniği, bir solucana benzer kıvrımlı bir yapı göstermesi sebebiyle bu isimle anılmıştır. Öncelikle nesne merkez olarak alınmakta ve nesnenin etrafı boyunca karolar dizilerek nesne vurgulanmaktadır. Mozaik yapmanın en etkili şekillerinden birisi olmakla beraber yüksek oranda işçilik ve hüner gerektirmektedir. Opus vermiculatum Şekil 1.11(c)’de görülmektedir.
- d) Opus palladianum (paledyen deseni): Bu teknikle yapılan mozaikler çatlak toprak görünümünü vermektedir. Mozaik yapımında kullanılacak olan karolar düzgün bir geometrik şekilden ziyade gelişi güzel parçalara kırılmaktadır. Daha sonra mozaik yapılacak nesne üzerine ve nesnenin arka planına rastgele döşenmektedir. Özellikle eskitilmiş taş parçalarının kullanılması ile etkili paledyen şekilli desenler oluşturulmaktadır. Opus palladianum Şekil 1.11(d)’de görülmektedir.
- e) Opus sectile/Florentine (Floransa deseni): Bu mozaik tekniğinde, nesnenin etrafında diğerlerine oranla daha büyük olarak kesilmiş karo parçaları kullanılmaktadır. Büyük boyutlu desen çalışmalarında etkili görünüşler ortaya çıkmaktadır. Floransa deseni Şekil 1.11(e)’de görülmektedir.
- f) Opus classicum (klasik desen): Bu teknik, opus vermiculatum (solucan deseni) ile opus tessellatum (tuğla deseni) tekniklerinin birleşmesinden meydana gelmiştir. Tasarımın ana nesnelere solucan deseni tekniği ile oluşturulurken, arka plan tuğla tekniği ile yapılmaktadır. Her bir nesnenin arka planında iki farklı desen oluşması nedeniyle, bu teknik ile çok etkili desenler ortaya çıkmaktadır. Klasik desen Şekil 1.11(f)’de görülmektedir.



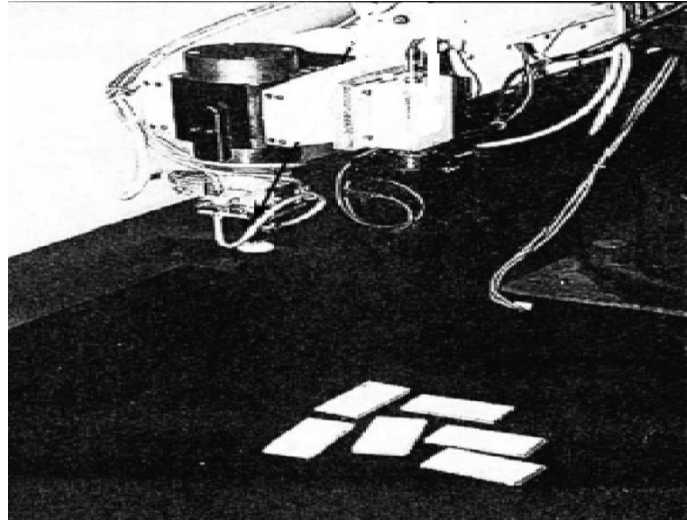
## 2. LİTERATÜR ÖZETİ

### 2.1 Giriş

Yapboz bulmaca çözen algoritmalar, hasar görmüş dokümanları ve tarihi eserleri onaran yöntemler ve foto gerçekçi olmayan kaplamalar kapsamındaki foto-mozaikler literatürdeki kardeş araştırma alanlarıdır. Yapboz bulmacalar ve hasar görmüş eserler, kenarlarının birbirine benzerliğine dayalı olarak parçaları bir araya getirirler. Foto-mozaiik çalışmalarında, mozaiği oluşturan parçalar (karolar) için yapay görüntülerden yararlanılmış, doğal taşlara ait görüntüler kullanılmamıştır. Kişinin albümündeki fotoğrafların küçültülmüş formlarını kullanarak sayısal bir fotoğrafın foto-mozaiğini dizen mobil uygulamalar mevcuttur.

### 2.2 Mozaik Dizme Otomasyonu İle İlgili Çalışmalar

Navon [8] grafik simülasyon ortamında, prefabrik bileşenleri duvarlara ve zemine monte etmek için, altı serbestlik derecesine sahip bir zemin döşeme robotu geliştirdi (Şekil 2.1). Navon, gerçek zamanlı kalite güvence süreçleri ve hasarlı karoları tespit eden bilgisayarlı görü içeren prototip çalışma prensipleri ve deneylerini sunup, sistemin sınırları ve ileri araştırma konularını değerlendirmiştir.



Şekil 2.1: Scara robot ile mozaik dizgi otomasyonu [8].

Oral ve Erzincanlı [9] kare ve dikdörtgen mermer parçalarını bir noktadan alarak mozaik üzerine yerleştiren bir SCARA robotu (Şekil 2.2) kontrol eden bir yazılım geliştirmişlerdir. Yazılım, mermer parça şekil ve ölçülerini, derz kalınlığını girdi olarak kullanıcıdan alarak, mermer parçaların kütle merkezi koordinatlarını hesaplamakta, tutucu rotasını oluşturarak,

robotun kinematik kontrolünü gerçekleştirmektedir. Bu çalışmada Oral ve Erzincanlı, kare şeklinde mermer parçalar kullanarak mozaik desenleri oluşturmuşlardır.



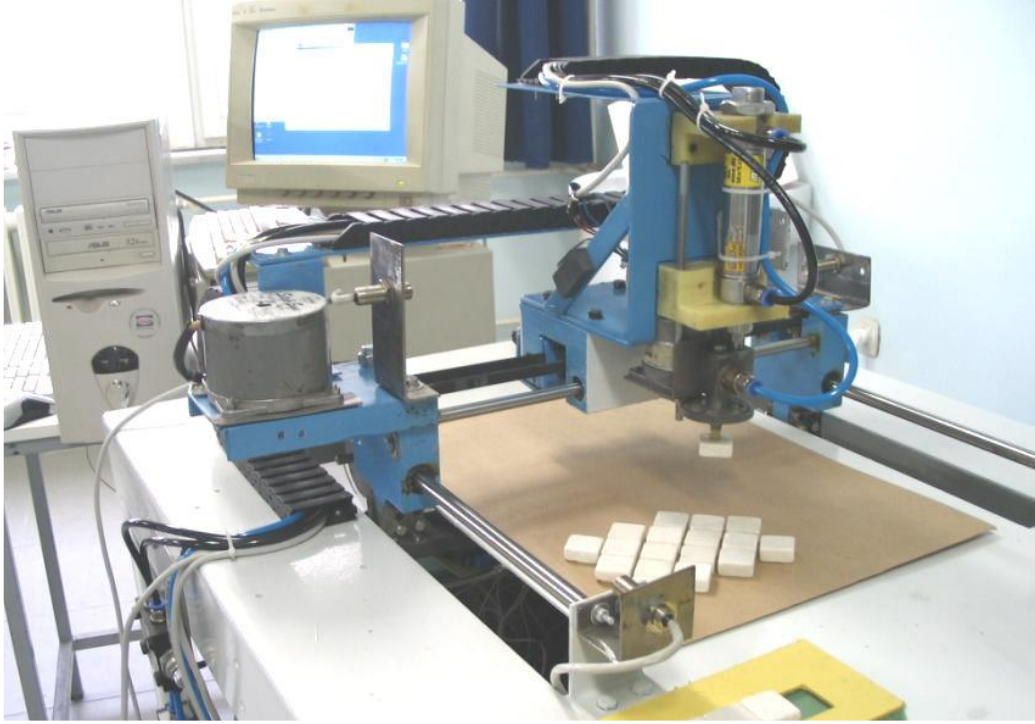
**Şekil 2.2:** Scara robot ile mozaik dizgi otomasyonu [9].

Kaya ve diğ. [10], altı serbestlik derecesine sahip endüstriyel Ultimate Puma 500 robotu kullanarak cam mozaik dizmek için C++ diliyle bir yazılım geliştirmişlerdir. Geliştirdikleri yazılım, jpg görüntüyü kullanılarak karo ve dizilecek mozaik çerçevesinin boyutlarına uygun olarak ölçeklemektedir. Çalışmada, hazır ve kararlı bir endüstriyel robot kol kullanılmış olup, bilgisayar görüntüleri kullanılarak geniş bir yüzey üzerinde farklı mozaik desenleri üretilmesine olanak sağlanmıştır. Mozaik oluşturmak için aynı boyutlarda kare şeklinde cam mozaik parçaları ile çalışılmış, farklı şekiller kullanılmamıştır.



**Şekil 2.3:** Ultimate Puma 500 robot cam mozaik dizerken [10].

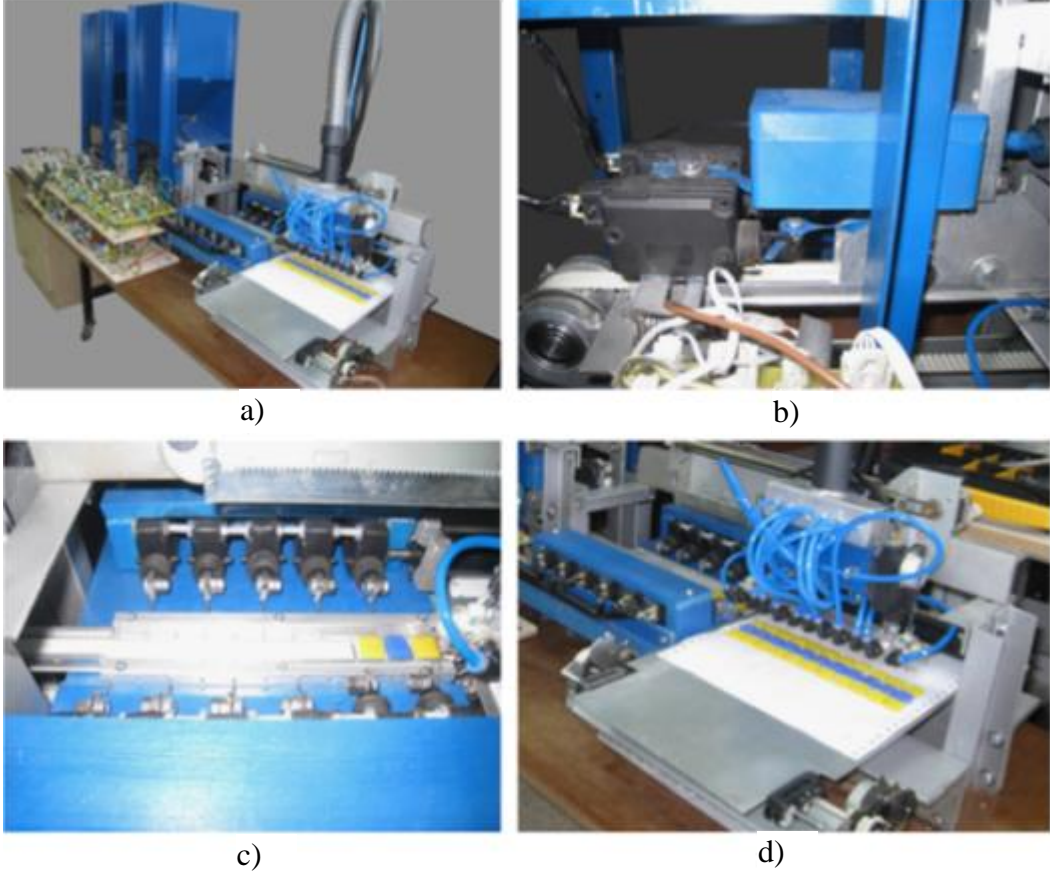
Oral ve İnal [11], “Dört Serbestlik Dereceli Robot ile Mozaik Dizme Otomasyonu” çalışmasında CAD yazılımı ile oluşturulan iki boyutlu DXF formatındaki desen dosyalarını okuyarak, tasarlanan mozaikleri Kartezyen robot ile dizen bir yazılım geliştirmişlerdir. Tasarım ve üretimde araştırmacılar tarafından yapılan dört serbestlik dereceli Kartezyen robot X ve Y eksenlerindeki hareketi adım motorlar yardımıyla sağlanırken, Z ekseninde yukarı aşağı hareket pnömatik silindir tarafından sağlanmaktadır (Şekil 2.4). Karoya istenen açığı vermek için Z ekseninde dönme hareketini veren bir adım motor daha vardır. Adım motor sürücüleri ve vakum sistemini kontrol devreleri, yazılım ile seri port aracılığıyla haberleşir. Çalışmada, mermer parça şekillerine üçgen, dikdörtgen, eşkenar dörtgen ve daire şeklinde parçalar da dahil edilmiştir.



**Şekil 2.4:** Dört serbestlik dereceli Kartezyen mozaik dizgi robotu [11].

A. Oral [12] tarafından yapılan diğer bir çalışmada üçgen, dikdörtgen ve kare şeklinde mermer parçaları kullanılarak, madalyon olarak adlandırılan mozaikler SCARA robot yardımıyla oluşturulmuştur.

Çayıroğlu ve Demir [13] tarafından bilgisayar yardımıyla cam mozaik dizme otomasyonu çalışmasında, sürekli bant kullanarak her sıraya 11 cam mozaik parçası yerleştiren otomatik mozaik dizme işlemi yapan bir sistem geliştirilmiştir (Şekil 2.5). Karoların alt ve üst kenarlarını ayırt etmek için karo görüntüleri Sobel filtreden geçirilip yapay sinir ağına verilerek yapıştırılacak pürüzlü kenar, cilalı üst kenardan ayırt edilmiştir. Sistem, istenen hız ve renk sayısında, kare karolarla desen oluşturacak şekilde tasarlanmıştır. Karo boyutları değiştirilmek istendiğinde, sistem 16 mm'lik kare karolar için tasarlandığından, makinanın ölçülerinin değiştirilmesi gerekmektedir. Geliştirilen yazılım ile görüntünün dizilmek istenen bölgesi seçilip ölçeklendikten sonra her bir piksel, cam karo renklerinden biri ile eşleştirilir. Makinenin kontrolünü sağlayan yazılım, üretim öncesi simülasyon da yapabilmektedir.



**Şekil 2.5:** Sürekli bant cam mozaik dizme otomasyonu a) Genel görünüm. b) Sürgülü sistem ve röleler. c) Sıralama kutusu. d) Ana bant ve pnömatik sistem [13].

Falomir ve diğ. [14] dijital görüntüde yer alan gerçek nesnelerin şeklini, rengini ve boyutunu niteliksel olarak tanımlayan ve bu tanımlar arasında benzerlik ölçüsünü hesaplayan bir yaklaşımla seramik endüstrisinde paledyen seramik parçalarının mozaik montajını gerçekleştiren bir sistem geliştirmişlerdir.

King ve diğ. [15], Endüstriyel robotun inşaat sektöründe kullanımını ve uygulanabilirliğini test etmek için, robot ile karo dizerek mozaik oluşturma konusunda iki entegre sayısal iş akışı geliştirmiştir. Bunların birincisi değişik boyutlarda standart karolar ile rastlantısal desen oluşturma, diğeri çeşitli boyutlarda karoları, derz ile kontrast oluşturacak şekilde yerleştirerek sayısal görüntü oluşturulmasıdır. (Şekil 2.6). Her iki iş akışında da mozaik görüntüsünün oluşturulması ve robotik hareket komutları dizisine dönüştürülmesi için Rhino yazılımı için görsel programlama arayüzü Grasshopper kullanılmıştır. Çıktı, ABB marka robotlara doğrudan yüklenebilecek RAPID dilinde kodları içeren metin dosyasıdır.

Uygulamalar ABB 4400 endüstriyel robotu pnömatik tutucu kullanılarak gerçekleştirilmiştir. Finansal uygunluk çalışmasında, ABD’de halen el emeği ile gerçekleştirilen mozaik dizme işleminin bir işçinin maliyeti ile robotik dizme işleminin maliyeti karşılaştırılmış, mozaik deseninin özelleştirilebilmesi açısından robot ile mozaik dizme uygulanabilir olarak kabul edilmiştir.



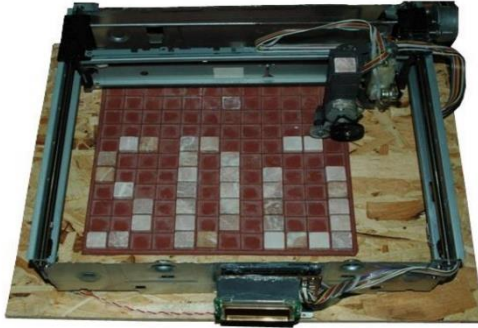
**Şekil 2.6:** ABB 4400 robotu, seramik karolar ile Marilyn Monroe'nun mozağini dizerken [15].

Jung, ve diğ. [16], NAO insansı robota vakum tutucu ekleyerek ve temel bilgisayarlı görü yöntemlerinden yararlanarak ticari robotlar ile kare karo zemin döşemenin gerçekleştirilebilirliğini araştırmışlardır. Şekil 2.7’de görüldüğü gibi NAO robot 23 saniyede bir karoyu yerden alabilmektedir.

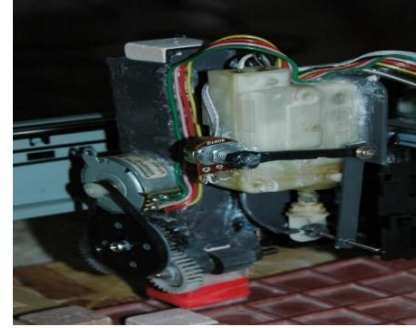


**Şekil 2.7:** NAO robot ile zemin döşeme [16].

İ.S. Üncü ve diğ. [17] “Görüntü işleme metoduyla mermer mozaik otomasyonu” çalışmalarında mermer ocaklarından elde edilen atık parçaları değerlendirmek amacıyla mozaik kalıplara uygun desen üreten otonom bir sistem üretilmiştir. Geliştirilen sistem bilgisayar kontrollü, üç eksendeki hareketleri adım motorları ile sağlanmaktadır. Sistem bilgisayar arayüzünden seçilen 100dpi çözünürlükteki resimleri, mozaik şablonlara 32x32'lik alana yerleştirebilmektedir. Fotoğrafi işleyebilmek için yazılım ile fotoğraf, 1cmx1cm kareler halinde 32x32 boyutlarında parçalanır. Oluşturulan 1cmx1cm alanların renk yoğunluğu bulunur. Görüntü, alanın renk yoğunluğuna göre eşiklenerek siyah veya beyaz renge dönüştürülür. Sistem taşları koyu ve açık renk olarak algılayarak, önce açık renk taşlardan sırayla koyu renge doğru sıralayarak yerleştirir (Şekil 2.8). Bilgisayarda işlenmiş olan fotoğraf RS232 seri haberleşme protokolü kullanılarak elektronik karta aktarılır. Veriler tasarlanan sistemde elektronik karta gönderilerek adım motorlar konumlandırma işlemi yapılır. Adım motorları PIC 16F877 tabanlı mikrodenetleyici vasıtasıyla sürülmüştür. Sistemin görüntü işleme alt yapısı C# programlama dili kullanılarak oluşturulmuştur. Geliştirilen sistem insan gücü ile yapılan bir işin otonomlaşmasını sağlayarak hızlı ve seri üretim imkanı vermektedir.



a)



b)

**Şekil 2.8:** a) Görüntüdeki koyu bölgelere yerleştirilen karolar. b) Taş bırakma kapsülü [17].

O. Önal [18] breşli kayalar kullanılarak mozaik üretimi çalışmasında, bir mermer kaynağından elde edilen breşli mermer karolar kullanarak iki görüntünün mozaikini sunmuştur. Mermer parçalarını görüntüye yerleştirmek için görüntü parçası ve karo arasında, görüntü piksellerinin R, G, B değerlerinin Öklid uzaklığını hesaplamıştır. Aynı kaynaktan gelen mermer parçaları genelde monokrom olmaktadır. Monokrom çalışmak, çözüm kümesini 16,7 milyon renkten, 255 farklı değere indirger ki bu çözümün başarısını aynı oranda arttıracaktır. Şekil2.9'da O. Önal'ın [18] 5 cm mermer karolar ile mozaik çalışmaları

görülmektedir. Solda; Botticelli'nin Venüs'ün doğuşu tablosundan Venüs'ün başı 81'e 62 karodan 12.56m<sup>2</sup> mozaik, sağda; William Bouguereau'nun taslak çiziminden oluşturulmuş 46'ya 83 karodan oluşan 9.55m<sup>2</sup> mozaik görülmektedir.



Şekil 4: Botticelli'nin Venüs'ün doğuşu tablosundan alınmış hedef resimle oluşturulmuş 81'e 62 elemanlı 12.56 m'lik mozaik.



Şekil 5: William Bouguereau'nun taslak çiziminden oluşturulmuş 46'ya 83 elemandan oluşan 9.55 m'lik mozaik.

### Şekil 2.9: O. Önal'ın mozaik çalışmaları [18].

P. Nuntaporn ve K. Pisut., [19], aynı ölçüde kare karolar kullanılarak, sayısal bir görüntüden, kişiselleştirilebilir mozaik tasarımı oluşturmak için, bir dizme otomasyonu geliştirmiştir. Sistem, yavaş hareket eden bir konveyör üzerine, yan yana sıralanmış karolardan oluşan satırlar yerleştiren bir mozaik üretim konseptini uygulamaktadır.

P. Panitcha, ve K. Pisut. [20], Nuntaporn ve Pisut'un ürün akış tabanlı karo dizme otomasyonu çalışmasında, satır oluşumunu iyileştirmek için yeni bir karo dağıtımı karar verme mekanizmasını LabView ile gerçekleştirmiştir.

Li X. ve diğ. [21], robotik dizme için robotun karoyu otomatik olarak seçmesi ve kalibre edilmemiş kameralar ve sınırlı görüş alanına rağmen istenen konuma yerleştirmesini sağlayan yeni, uyarlanabilir, görme tabanlı denetleme şeması önermiştir. Önerilen denetleyici robotlara iki aşamalı dizme yeteneği sağlamaktadır. İlk olarak, kalibre edilmemiş kameralar nedeniyle bilinmeyen konum bilgileri, manuel kalibrasyon gerekli olmayacak şekilde çevrimiçi olarak tahmin edilir. İkinci olarak, sınırlı görüşten kaynaklanan özellik kaybı, Kartezyen-uzay bölgesel geri besleme ile manuel yardıma gerek kalmadan giderilir.



Kapalı döngü sisteminin kararlılığı, Lyapunov yöntemleri ile titizlikle kanıtlanmıştır ve önerilen kontrol şemasının performansını göstermek için deneysel sonuçlar sunulmuştur.

### **2.3 Dijital Mozaik Teknikleri ile ilgili Çalışmalar**

Görüntünün yapay olarak mozağının oluşturulması, “bilgisayar grafikleri” alanında, “foto gerçekçi olmayan kaplamalar” başlığı altında incelenmektedir. Sanatsal kalitede dijital mozaik oluşturmak, bilgisayar grafikleri alanında en zorlu konularından biridir.

#### **2.3.1 Görüntünün Mozaik Görüntüsü Haline Getirilmesi**

Bilgisayar grafikleri alanında 2006’da en yeni araştırma alanlarından biri fotogerçekçi olmayan kaplamalardır. Dijital mozaikler, “karo” olarak adlandırılan küçük görüntülerin yan yana döşenmesiyle elde edilirler. Aynı kaynak görüntü ile başlayarak, karo veri tabanının seçimine ve uygulanan kurallara bağlı olarak, farklı sayısal mozaikler yaratılması mümkündür. Sayısal mozaik üretilmesi problemini çözmek için ilk adım problemi matematiksel çerçevede yeniden formüle etmektir Görüntünün mozağını oluşturma problemi, matematiksel optimizasyon problemi olarak ele alınabilir [22].

#### **2.3.2 Foto Mozaikler**

Foto mozaik bir giriş görüntüsünü küçük resimlerden oluşan dikdörtgen bir ızgaraya dönüştürür. Bu yöntemde algoritma genellikle büyük bir görüntü veri tabanında, giriş görüntüsüne uygulanan ızgaranın hücrelerinin görüntülerine benzeyen görüntüler arar.  $\mathbb{R}^2$  düzleminde verilen  $I^2$  görüntüsü, küçük dörtgen görüntüler (karo) veri tabanı verilmekte ve  $N$  hücreden oluşan düzenli dörtgen ızgara,  $I^2$  görüntüsüne uygulanmaktadır. Izgaradaki her bir hücre için, veri tabanında, hücre görüntülerine benzeyen  $N$  adet karo görüntüsü bul ve ızgaraya yerleştir [22].

Di Blasi ve Petralia, hızlı foto-mozaik oluşturmak için “Zıt kutup kümeleme yöntemi” kullanan bir yöntem sunmuştur [23].

D'Souza ve diğ. insan yüzü görüntüsünün minyatürlerini, karo olarak kullanarak, yüzün fotomozağını oluşturan bir sistem tasarlamışlardır. En iyi karoyu bulmak için evrimsel arama ve genetik algoritma kullanmışlardır [24].

Narasimhan ve Sathesh Fotomozaik oluşturmak için rastgele yinelemeli iyileştirme algoritması önermişlerdir. Ara sonuçları animasyonlar üretmek için kullanmışlardır. 400x600, 600x800 ve 800x1000 ebatlarında görüntülerde 10x10 karolar kullanmışlardır. Önerilen tekniğin genetik algorithmadan daha verimli olduğunu göstermişlerdir [25].

Mikamo ve diğ. fotomozaik eşleştirme aşaması için üç yaklaşım geliştirmiştir: Açgözlü tabanlı prosedürel bir algoritma, benzetilmiş tavlama ve SoftAssign. Son iki yaklaşım, yalnızca sınırlı sayıda karo bulunduğu durumlarda etkili olmaktadır [26].

Yang ve diğ. Macar yöntemi ya da Blossom Algoritması ile fotomozaik probleminin çözülebileceğini belirtmişler ve denemelerinde Blossom V kullanmışlar ve GPU grafik işlemcide paralel çalışacak bir metot geliştirmişlerdir [27].

Zhang ve diğ. sundukları fotomozaik algoritma, global hatayı Macar Yöntemi ile minimize ederken, yerel hataları azaltmak için atanan karo görüntüsünün, atandığı bloğa 0.0 ile 1.0 arasında değer alan  $\alpha$  oranında benzetme yoluna gitmişlerdir [28].

## **2.4 Konunun Literatürdeki Önemi**

Problem geniş kapsamlı ve disiplinler arasıdır. Sanatsal mozaik ile ilgili bilimsel çalışmalar bilgisayarlı grafik alanındadır. Sanatsal mozaik üretimine dönük çalışma yok denecek kadar azdır. Literatürde, mermer mozaik dizmek için doğal taş tessera kullanırken benzerlik ölçüsünde rengi göz önünde bulunduran ve mozaik'in toplam hatasını azaltmak için bir optimizasyon yöntemi kullanan çalışmaya rastlanmamıştır.

Görüntünün mozaikini mermer karo görüntüleri ile dizme konusundaki yapılan literatür taramasında, görüntü işleme, özellik çıkarma, bilgisayarlı görü ve optimizasyon yöntemlerinden yeterince yararlanılmadığı görülmüştür.

Sınırlı sayıda karolarla çalışıldığında, görüntüdeki nesnelere algılayarak bu nesnelere ait görüntü parçalarına, arka plana göre daha uygun karolar atayarak sonucu iyileştiren bir çalışmaya rastlanmamıştır.

### 3. MATERYAL VE YÖNTEM

#### 3.1 Giriş

Desen, nesnelere ve mozaik oluşturmada kullanılacak doğal taş, cam ve seramik parçaları, bilgisayar ortamında sayısal görüntüleri ile temsil edilirler. Görüntü işleme yöntemleri ile işlenirler. Bu çalışmada, doğal taş, seramik ve cam tesseraların yüzey görüntüleri “karo” olarak adlandırılmıştır. Mozaiği dizilmek istenen -desen veya nesne içeren- görüntü, dizmede kullanılacak karoların ölçülerinde parçalara ayrılır. Her bir desen “görüntü parçası” ile yerine yerleştirilen karo arasında bir hata vardır. Çalışmada bu hata görüntü vektörlerinin Öklid uzaklığı olarak ölçülmüştür. Çalışmada görüntünün tamamına yerleştirilen karolardan kaynaklanan toplam hatanın en az olması için bir optimizasyon yöntemi önerilmiştir.

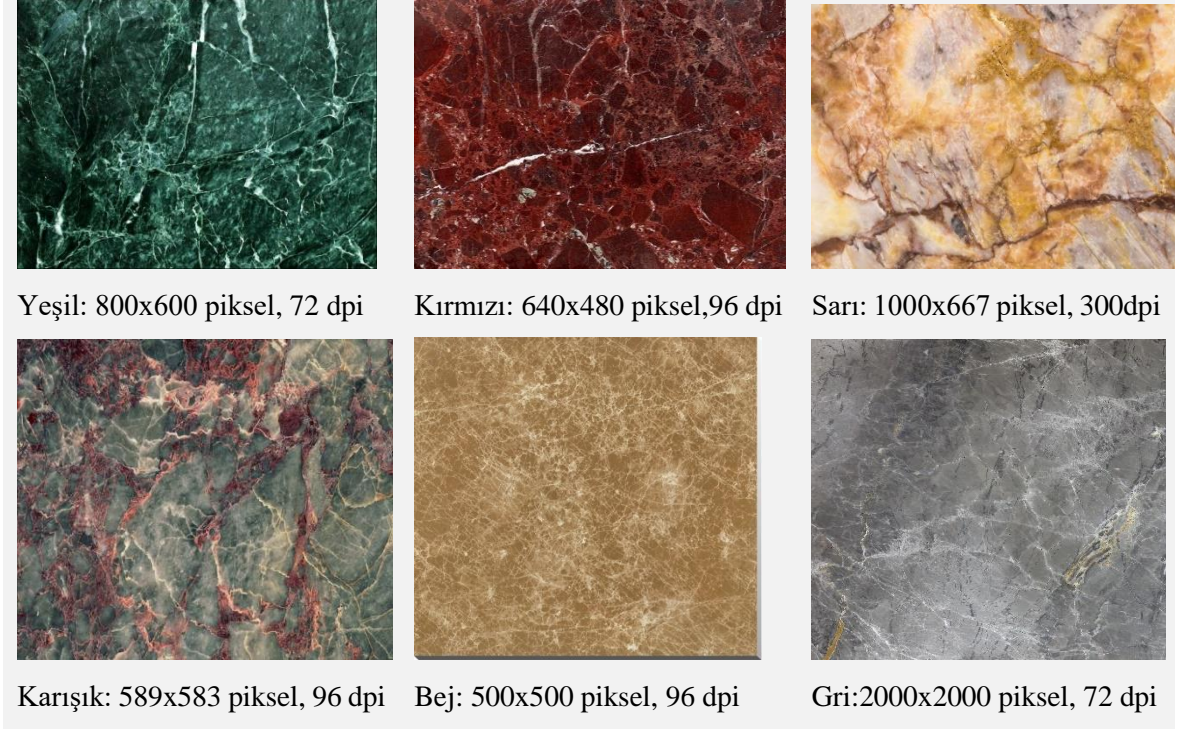
Nesne ya da desen içeren görüntülerin mozaiğini dizmek amacıyla esnek üretim yapan otomasyon sistemi, çalışma düzlemindeki mevcut karoları kullanarak, desen kalıbı kullanmaksızın, üretilmek istenen mozaik için, karoların en uygun dizilimini sağlamalıdır. Mermer, doğal taş, cam, metal vb. değişik malzeme çeşitlerinden, değişik şekil ve ölçülerdeki parçalar kullanılabilir. Bu sistemin gerçekleştirilebilmesi için gerekli yöntemler ve teknikler şunlardır:

1. Karoların Algılanması: Çalışma düzleminin kamera görüntüsü kullanılarak, karoların algılanması, konum, şekil, ölçü, parlaklık ve renk özelliklerinin elde edilmesi için görüntü işleme yöntemleri.
2. Mozaiği dizilecek görüntünün, karolara uygun büyüklükte görüntü parçalarına bölünmesi.
3. Renkli iki görüntü parçası arasında benzerliğin ölçülmesi için matematiksel yöntemlerin belirlenmesi.
4. Görüntü parçalarına, sınırlı sayıdaki karolardan en uygun olanlarının atanması ve toplam hatanın minimize edilmesi için uygun eniyileme (optimizasyon) yönteminin belirlenmesi.
5. Farklı ölçülerde karoların bulunması durumunda, görüntü parçalarına karo atanması için bir yöntem geliştirilmesi.

6. Görüntüde nesnelere tespit ederek, nesne içeren görüntü parçalarına, arka plana göre daha uygun karolar atanması için bir yöntem geliştirilmesi. Bu yol ile sınırlı sayıda mermer parça ile dizilecek mozaik kalitesinin artırılması.
7. Karoları çalışma düzleminde bulunduğu konumdan alınıp, mozaikte bulunması gereken konuma yerleştirilmesi için tut ve yerleştir robot mekanizmasının tamamlanması.

Hem karonun hem de görüntü parçasının RGB renk uzayında 24bitlik piksellerden oluşan, eni, boyu belirli görüntü olmalarının ötesinde, çalışma düzlemindeki yeri, görüntüdeki yeri, renk kanallarının ortalaması, standart sapması, kovaryans matrisi, entropisi, belirgin olup olmadığı, HOG doku tanımlayıcısı, CIELab renk uzayındaki karşılığı ve 90°,180°,270° döndürülmüş durumları, kendi indisi ve eşleştiği karonun indisi gibi birçok özelliği depolamak amacı ile “**karo**” adında bir yazılım nesnesi (sınıf) oluşturulmuştur. Çok sayıda karo nesnesi, liste yapısında (Liste<Karo>) tutulmaktadır. Eşleşecek karolar için bir liste, görüntü parçaları için diğer bir liste oluşturulmuştur.

Mermer mozaik karoları dizmek için tut ve yerleştir işlemlerinde yaygın olarak kullanılan Delta Robot prototipi tasarlanmıştır. Çalışma düzlemine rastgele saçılmış karoların kamera ile tespit edilmesi, robot uç işlevcisi ile bulunduğu konumdan alınıp yerleştirilmesi prosesinin düzgün çalışması için birçok deneme ve düzeltme yapılması gerekmektedir. Sistem çalışır duruma gelene kadar, görüntü işleme, nesne algılama, matematiksel yöntemlerin ve yazılımın geliştirilebilmesi için, gerçek koro görüntüleri yerine, mermer levha görüntülerinin bilgisayar ortamında 8x8, 16x16 ve 32x32 piksel ölçülerinde bölünerek elde edilen karo görüntüleri kullanılmıştır. Renklerine göre gruplandırılan mermer levhaların her grup için bir örnek görüntü Şekil 3.1’de verilmiştir.



**Şekil 3.1:** Mermer levha görüntüleri

Tablo 3.1’de Bilgisayar ortamında oluşturulmuş, renklerine göre gruplanmış mermer levha görüntülerinin dosya sayıları, görüntülerin toplam alanı ve bu gruplardan oluşturulmuş 8 piksel, 16 piksel ve 32 piksel kenar ölçülerine sahip kare karo görüntü dosyalarının sayısı görülmektedir.

**Tablo 3.1:** Bilgisayar ortamında oluşturulmuş karo bilgileri.

Grup	Grup Dosya Sayısı	Dosyaların Toplam Alanı (piksel <sup>2</sup> )	8 piksel Karo Sayısı	16 piksel Karo Sayısı	32 piksel Karo Sayısı
Yeşil	3	3.260.000	3.205	3.193	3.113
Kırmızı	6	1.870.801	1.878	1.862	1.708
Sarı	1	667.000	672	651	620
Karışık	3	4.329.307	4.336	4.201	4.147
Bej	15	3.155.173	3.044	2.950	2.778
Gri	5	8.320.160	8.244	8.184	8.003

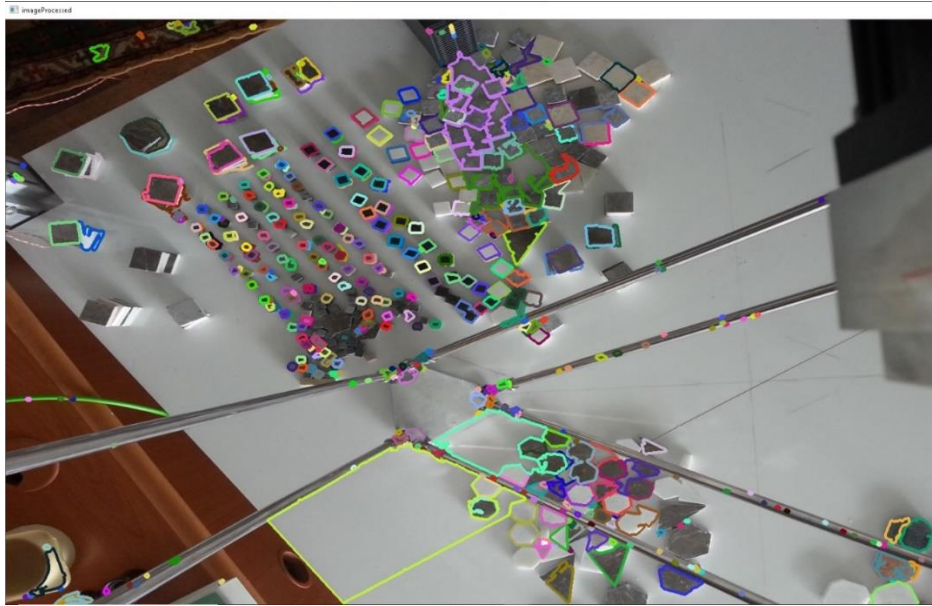
Delta robot ile mermer mozaik dizme denemelerinde kullanılmak üzere Balıkesir’de bir mermer mozaik üreticisinden Tablo 3.2 de görülen mermer karo örnekleri alınmıştır.

**Tablo 3.2:** Bir mozaik üreticisinden alınan çeşitli karolar

Karo Şekli	Ölçüler(mm)	Adet
Küp	10x10x10	470
Kare prizma	14x14x10	310
Kare prizma	20x20x10	150
Kare prizma	23x23x10	315
Kare prizma	30x30x10	48
Paralelkenar	57x32, h=10	5
Dik üçgen	50x50, h=10	20
Altıgen	28, h=10	20
Köşeleri pahlı kare	48x48x10	50
Kare prizma	48x48x10	23

### 3.2 Karoların Algılanması

Çalışma düzlemine gelişi güzel dağılmış mermer karo parçalarının görüntülerinin bilgisayar ortamına aktarılması için bir IP kamera düzeneği kurulmuştur. Görüntü işleme yöntemleri kullanılarak alınan görüntü bölütlenir, her bir mermer parçaya ait görüntü izole edilir, numaralandırılır, bellekte ve görüntü dosyalarında depolanarak dizme işleminde kullanılmak üzere karo görüntüleri veri kümesi oluşturulur. Karoları algılamak için Canny Kenar Bulma Algoritması temelli nesne algılama yöntemi kullanılmıştır. Şekil 3.2’de çalışma düzlemi görüntüsüne uygulanan algoritmanın tespit ettiği karolar, kenar pikselleri yapay renklendirilmiş olarak görülmektedir.



**Şekil 3.2:** Çalışma düzleminde algılanan karolar.

Algoritmanın adımları aşağıdaki gibidir:

1. OpenCV GaussianBlur(5,5) fonksiyonu ile 5x5 ölçülerinde Gaussian filtre ile gürültü uzaklaştırılarak görüntü yumuşatılır.
2. Ardından Canny kenar bulma algoritması Canny(altEşik, üstEşik) uygulanır. Sobel operatörü ile görüntüde gradyanların genlikleri ve açıları hesaplanır. Gradyanların genliği *üstEşik*'ten büyükse, güçlü kenardır, *altEşik*'ten küçükse ihmal edilir. Eğer gradyan genliği *altEşik* ile *üstEşik* arasında ise, sadece güçlü kenar pikselleri ile bir bağı varsa kenar pikseli olarak kabul edilir. (Uygulamada *altEşik*=50, *üstEşik*=200 alınmıştır.)
3. 3x3 yapılandırma elemanı ile morfolojik kapama (closing) uygulanır. Bu, morfolojik kabartma işlemi sonrası aşındırma işlemidir. Kopuk nesnelere birleştirip, nesne üzerindeki delikleri kapatarak gürültüyü gidermek amacı ile uygulanır.
4. Kenarlar dolaşılır, EmguCV CvInvoke.FindContours() metoduyla kenar noktaları dolaşarak konturlar elde edilir.
5. Eğer konturun uzunluğu 6 noktadan küçükse bu noktalara EmguCV CvInvoke.BoundingRectangle(contour) metoduyla bir dikdörtgen uydurulur.
6. Aksi takdirde (konturun uzunluğu 6 nokta veya daha fazla) EmguCV CvInvoke.FitEllipse(contour) metoduyla kontur noktalarından geçen elips uydurulur.
7. Konturun en-boy oranı Eşitlik (3.1) ile hesaplanır:

$$en\ boy\ oranı = \frac{maks(en, boy)}{min(en, boy)} \quad (3.1)$$

8. Konturun Convex Hull'u yani bu noktaları içine alan en küçük konveks noktalar kümesi CvInvoke.ConvexHull() metoduyla bulunur ve alanı: *konturConvexHullAlanı* hesaplanır. Bu değerler Eşitlik (3.2)'de yerine konarak *katılık* hesaplanır.

$$Katılık = \frac{konturAlanı}{konturConvexHullAlanı} \quad (3.2)$$

9. Kontur momentleri Eşitlik (3.3) ile hesaplanır, CvInvoke.Moments() metodu 3. Dereceye kadar moment hesaplar.

$$m_{ji} = \sum_{x,y} (konturNoktaDizi(x, y) \cdot x^j \cdot y^i) \quad (3.3)$$

Konturun kütle merkezi Eşitlik (3.4) yardımıyla hesaplanır.

$$Cx = \frac{m_{10}}{m_{00}}, Cy = \frac{m_{01}}{m_{00}} \quad (3.4)$$

10. Kontur alanı 25'den büyükse ve görüntünün alanının dörtte birinden küçükse ve en-boy oranı 2.1'den küçükse ve kütle merkezi Eşitlik (3.4) ile hesaplanabiliyorsa ve *katılık* 0.5 den büyükse bir karo tespit edilmiştir, sol üst köşesinin koordinatına yerleştirilmiş eni  $w$ , boyu  $h$  olan dikdörtgen  $(x, y, w, h)$  ile kutulanır. Kontrol edilen mantıksal ifade: Eğer(  $(\text{konturAlanı}>25) \wedge (\text{konturAlanı}<\text{GörüntüAlan}/4) \wedge (\text{enboyoranı}<2.1) \wedge (Cx>0) \wedge (Cy>0) \wedge (\text{katılık}>0.5)$  )  $\equiv$  Doğru
11. Bulunan karonun görüntüsü ve konum bilgisinden, bir “Karo” nesnesi yaratılarak Liste<Karo> listesine eklenir.

Çalışmada otomasyon sisteminin görüntüyü, sanayi mozaiklerinde kullanılan kare taşlar kullanarak dizmesi hedeflenmiştir. Sanayi mozaiklerinde kullanılan karolar temel geometrik şekillerde üretilmektedir. Görüntü işleme ve algılamada, nesnelerin görüntüleri, bilgisayar ortamında genellikle nesneyi içeren bir kare ya da bir dikdörtgen çerçeve ile sınırlandırılarak işaretlenir. Bu nedenle dikdörtgen olmayan karoların işaretlenmesinde bir miktar karoya ait olmayan alan, görüntüye dahil olacaktır.

Dizme sistemi için, sanayi mozaiklerinde kullanılan, yüksekliği 10 mm, kenar ölçüleri birbirinin iki katı olan, alanı: 10x10 mm<sup>2</sup>, 20x20 mm<sup>2</sup> ve 40x40 mm<sup>2</sup> mermer parçaları esas alınmıştır (Şekil 3.3). Sanayi mozaiklerinde izin verilen 1-2 mm'lik derz boşlukları, çalışmada en fazla 1mm olacak şekilde sınırlandırılmıştır. Ebatları 10 mm, 20 mm ve 40 mm olan kare karo görüntülerinin, yazılımla 8 piksel, 16 piksel ve 32 piksel boyutlarında görüntülere karşılık gelecek şekilde kullanıcı tarafından ölçeklenmesi sağlanmıştır. Bunun nedeni, kameranın çalışma düzlemine uzaklığının uygulamadan uygulamaya farklı olması durumunda, taşların piksel cinsinden ölçülerinin de farklı olacağı durumudur. 10 mm, 20 mm ve 40 mm olan kare karo görüntülerinin 8 piksel, 16 piksel ve 32 piksele ölçeklenmesi özellik çıkarma olarak da ele alınabilir.





**Şekil 3.3:** Soldan sağa 60 mm, 40 mm, 20 mm, 10 mm ölçülerinde karolar [4].

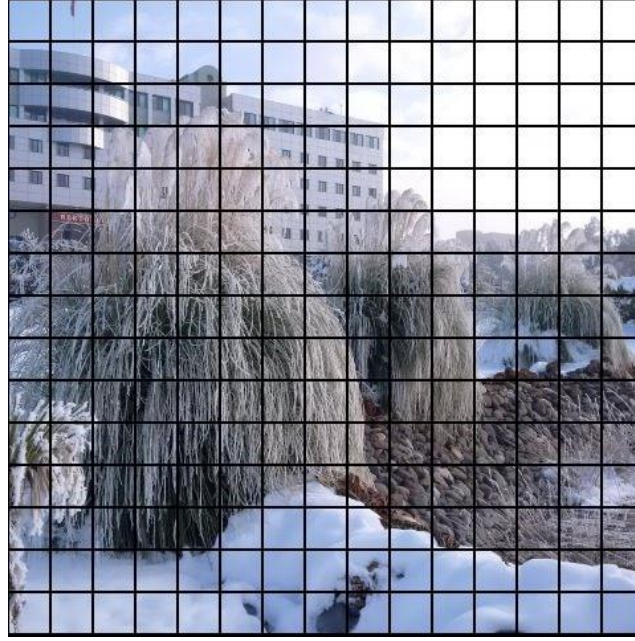
Kullanıcı karoları ölçekledikten sonra, veri kümesindeki her bir karo görüntüsünün özellikleri; ortalaması, standart sapması, bağıl standart sapması ( $\frac{\sigma}{\mu}$ ), CIELab renk uzayına dönüştürülmüş ve 90°, 180°, 270° döndürülmüş görüntüleri vb. hesaplanır. Karolar ölçülerine göre (8, 16, 32 piksel) gruplanır, gruptaki karo sayıları histogramı hesaplanır. En büyük ve en küçük karoların ölçüleri tespit edilir. Karo görüntüleri veri kümesi güncellenir.

### 3.3 Görüntünün Parçalara Bölünmesi

Sayısal ortamda görüntüler genellikle JPEG formatında kullanılmakta ve gerek duyulan görüntü kalitesini sağlayan, en yüksek oranda sıkıştırılmaktadır. JPEG formatında görüntüler, 8x8 piksellik kutulara bölünerek, her bir kutuya kosinüs dönüşümü uygulanarak sıkıştırmaktadır. Bu nedenle, mozaği dizilecek desen görüntüsünün en küçük görüntü parçası alanı 8x8 piksel olarak sınırlandırılmıştır.

Mozaği oluşturulacak görüntü, dizilecek karoların şekil ve ölçülerine uygun parçalara ayrılmalıdır. Görüntü parçalarının ölçülerinin belirlenmesi için, önce dizmede kullanılacak mermer karo kümesinden ilgili bilgilerin alınması gerekmektedir.

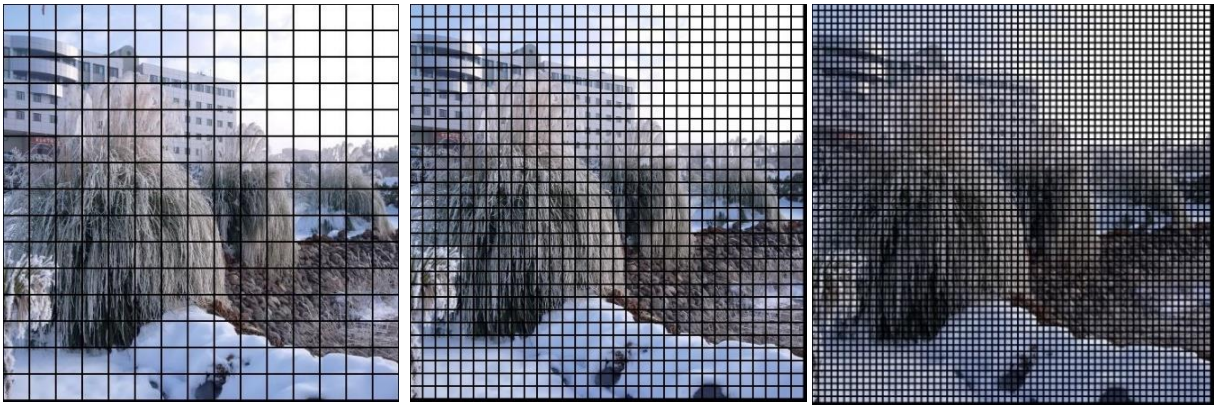
Eğer karo görüntüleri veri kümesindeki karoların tamamı aynı ölçüde ise, hücre ölçüleri karolarla aynı olan düzenli ızgara ile görüntü parçalara bölünerek, desen “görüntü parçaları veri kümesi” oluşturulur. Şekil 3.4’de 32x32 piksel karolar için uygulanan ızgara görülmektedir.



**Şekil 3.4:** Desen görüntüsüne uygulanmış 32x32 piksel düzenli ızgara.

### 3.3.1 Farklı Ölçülerde Karolar için Görüntünün Parçalara Bölünmesi

Eğer çalışma düzleminde çeşitli ölçülerde, örneğin bir kenarı 10 mm, 20 mm, 40 mm karolar olması durumunda, bunların karşılaştırılabilmesi için desen görüntüsünün de uygun hücre ölçülerinde ızgaralar kullanılarak bölünmesi, her ölçü için görüntü parçaları veri kümelerinin oluşturulması gerekir. Şekil 3.5 da hücresinin bir kenarı 32, 16, 8 piksel olan ızgaralar ile desen görüntüsünün bölünmesi görülmektedir.



**Şekil 3.5:** 32, 16, 8 piksel hücrelere sahip ızgara ile bölünmüş görüntü.

Her 32x32 piksellik görüntü bölgesine, 4 adet 16x16 piksel ve 16 adet 8x8 piksel görüntü parçası denk düşmektedir. Bu görüntü bölgesine 32, 16 ve 8 piksel karolardan hangilerinin atanacağını hesaplamak için bu görüntü bölme işlemleri yapılmış ve veri kümeleri oluşturulmuştur.

### 3.4 İki Görüntü Parçası Arasında Benzerliğin Ölçülmesi

Di Blasi ve Petralia [23], hızlı foto-mozaiik oluşturmak için kolay bir yöntem sunmuşlardır. Karo olarak kullanılacak görüntülerin veri tabanını oluşturmak için her bir karo görüntüsü, 9 eşit dörtgene bölünerek 3x3 grid oluşturulur. Her bölge kendi içinde kırmızı(R), yeşil(G) ve mavi(B) kanalları için ortalama değerleri hesaplanır. Üç renk kanalı için üç adet ortalama değer bize 27 elemanlı özellik vektörünü  $x$ 'i verir. Veri tabanındaki her bir karo görüntüsü 27 elemanlı özellik vektörü  $x$  hesaplanır. Foto-mozaiği oluşturulmak istenen görüntü yatayda ve düşeyde parçalara bölünerek düzenli bir grid oluşturulur. Sonra her parça kendi içinde yeniden 3x3 alt bölgeye ayrılır. Her alt bölgenin RGB renk kanallarının ortalaması hesaplanarak 27 elemanlı özellik vektörü  $y$  hesaplanır.  $y$ , görüntünün tüm parçaları için hesaplanır. Son olarak her bir  $y$  vektörü ile en iyi eşleşen  $x$  vektörüne ait karo görüntüsü ölçeklenerek, görüntüde ilgili alana yerleştirilir. En iyi eşleşme, birbirine uzaklığı minimum olan  $x$  ve  $y$  vektörleridir. Yöntem veri tabanında karo ararken zıt-kutup(antipole) ağaç yapısı kullanır. Zıt-kutup ağaç yapısı metrik uzayda büyük veri kümelerinde arama yapmak için uygundur. Zıt-kutup ağaç kümeleme algoritması, en uzak kayıtların farklı kümelere yer alması gerektiğinden hareketle verileri yinelemeli olarak alt kümelere böler. Sonuç yaprakları son kümeler olan ikili ağaç yapısıdır. Daha sonra kökten başlayarak en yakın küme merkezini bulmayı garanti eden  $k$  en yakın komşu (k-NN) arama ağaçtaki yolları takip ederek kümeleri oluşturur [23].

#### 3.4.1 Benzerlik ve Uzaklık

Veri madenciliği bağlamında benzerlik; Veri kümesindeki kayıtları temsil eden boyutlarda hesaplanan, bir kayıtın diğer kayıta uzaklığı ile ters orantılıdır [29].  $V$  olarak adlandırılacak bir veri kümesindeki  $m$  adet kayıt  $V = \{X_1, X_2, X_3, \dots, X_m\}$  ve  $i$ . kayıtın elemanları:  $X_i = \{x_{i1}, x_{i2}, x_{i3}, \dots, x_{in}\}$  olmak üzere  $R^n$ ;  $n$  boyutlu vektör uzayında  $X_i$  ile  $X_j$  arasındaki Öklid uzaklığı Eşitlik (3.5) ile hesaplanır:

$$d(X_i, X_j) = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \quad (3.5)$$

Eşitlik (3.5)'de RGB renk uzayı için  $n=3$  dür. Benzerlik kavramı ise uzaklığın tersi bir anlam içerir ve iki veri arasındaki yakınlığı gösterir. Genel olarak benzerlik:

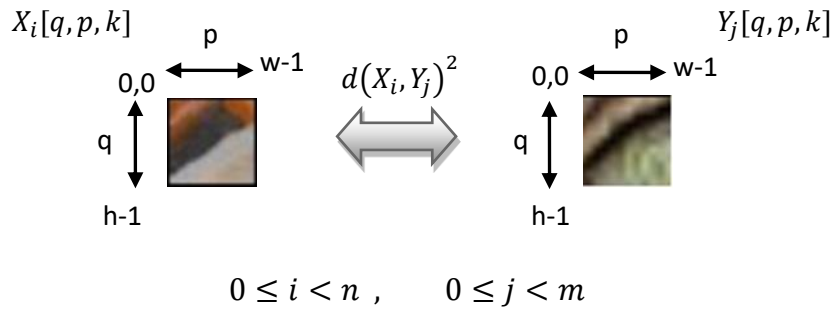
$$S(X_i, X_j) = \frac{1}{1 + d(X_i, X_j)} \quad (3.6)$$

Eşitlik (3.6) deki gibi ifade edilir. Bununla beraber benzerlik ölçümü için çeşitli yöntemler vardır. Bunlardan birisi yarı metrik olan Dice benzerlik ölçümü Eşitlik (3.7)'deki gibidir.

$$S(X_i, X_j)_{Dice} = \frac{2 \sum_{k=1}^n x_{ik} x_{jk}}{\sum_{k=1}^n x_{ik}^2 + \sum_{k=1}^n x_{jk}^2} \quad (3.7)$$

Diğer benzerlik ölçütleri Jaccard, Cosine ve Overlap örnek olarak verilebilir.

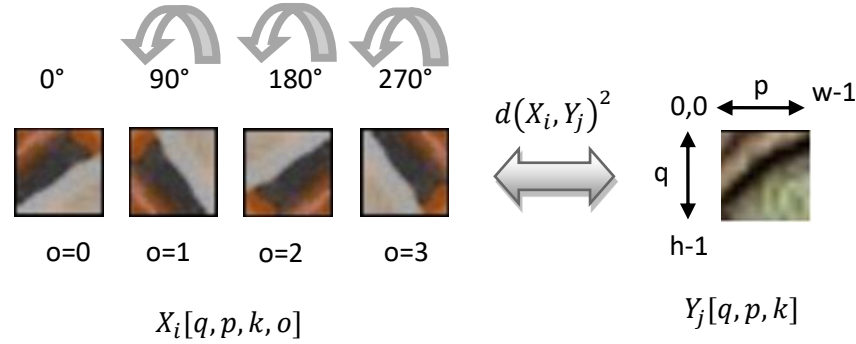
Nesnelerin en uygun yapı taşları ile dizilmesi probleminde, karo sayısı:  $n$  adet mermer karo görüntüsünün oluşturduğu veri kümesi  $\mathbf{X}$  de  $i$ . karo görüntüsü  $X_i$  olsun. Görüntü parçası sayısı:  $m$  adet görüntü parçasının oluşturduğu veri kümesi  $\mathbf{Y}$  de  $j$ . görüntü parçası  $Y_j$  olsun. Görüntüler üç kanal ( $c=3$ ) CIELab görüntü uzayında, genişlikleri  $w$  ve yükseklikleri  $h$  olsun.  $X_i$  ve  $Y_j$  görüntüler arasındaki uzaklık Eşitlik (3.8)'de verilmektedir.:



**Şekil 3.6:** Birbirine benzerliği hesaplanan solda karo, sağda görüntü parçası.

$$d(X_i, Y_j)^2 = \sum_{q=0}^{h-1} \sum_{p=0}^{w-1} \sum_{k=0}^{c-1} (X_i[q, p, k] - Y_j[q, p, k])^2 \quad (3.8)$$

Bir karo  $90^\circ, 180^\circ$  ve  $270^\circ$  döndürülerek dört farklı şekilde yerleştirilebilir. Bu dört durum için uzaklığı hesaplayıp en yakın karoyu ve dönme açısını bulmak için uzaklık formülü Eşitlik (3.9) ve Eşitlik (3.10)'daki gibi güncellenir.



**Şekil 3.7:** Görüntü parçasına benzerliği hesaplanmak üzere üç kez  $90^\circ$  derece döndürülmüş karo.

$$d(X_i, Y_j)^2 = \min_o \left[ \sum_{q=0}^{h-1} \sum_{p=0}^{w-1} \sum_{k=0}^{c-1} (X_i[q, p, k, o] - Y_j[q, p, k])^2 \right] \quad o = 0, 1, 2, 3 \quad (3.9)$$

$$Açıl[i, j] = o \quad (3.10)$$

$$o = 0, 1, 2, 3, \quad 0 \leq i < n, \quad 0 \leq j < m$$

Her  $X_i$  karosunun içerisinde CIELab renk uzayına dönüştürülmüş ve  $0^\circ, 90^\circ, 180^\circ, 270^\circ$  döndürülmüş 4 karo görüntüsü indisleri  $o=0, 1, 2, 3$  olacak şekilde belirlenmiştir.

Görüntü parçası  $Y_j$  'nin sadece  $0^\circ$  yani döndürülmemiş CIELab renk uzayındaki görüntüsü işleme girmektedir.  $p$  ve  $q$  sırasıyla yatay ve düşey görüntü pikselleri için indislerdir. Lab renk uzayının kanal sayısı  $c=3$  ve indisi  $k=0, 1, 2$  dir.

CIELab renk uzayı, tıpkı RGB gibi üç kanallıdır. İnsan renk algısının ve gamının tamamını kapsar. Renk üç değer ile ifade edilir.  $L^*$  algısal aydınlık için 0 siyah ile 100 beyaz arasında değer alır.  $a^*$  yeşil ve kırmızı karşıt renkleri arasında değer alır.  $b^*$  mavi ve sarı karşıt

renkleri arasında deęer alır. CIELAB,  $a^*$  ve  $b^*$  eksenleri sınırsızdır ve  $L$  deęerine baęlı olarak, insan gamını kapsayacak şekilde kolayca  $\pm 150$ 'yi ařabilirler. Bununla birlikte, yazılım uygulamaları genellikle bu deęerleri pratik nedenlerle sıkıřtırır. Örneęin, 8bit tamsayılar kullanılıyorsa,  $a^*$  ve  $b^*$  -128 ile 127 aralıęına ölçeklenir. Belirli bir sayısal deęişim, renkte benzer algılanan bir deęişime neden olur ve küçük renk farklılıklarını tespit etmek için kullanılıřtır. CIELab renk uzayı donanımdan baęımsızdır. Tanımladıęı renkler, bir bilgisayar monitörü veya bir yazıcı gibi belirli bir cihazla iliřkili deęildir, bunun yerine laboratuvar kořullarında renk eřleřtirme deneyleri sonuçlarının ortalaması ilgilidir [30].

Her  $i$  ve her  $j$  için,  $Y_j$  görüntü parçası ile  $X_i$  mermer karosunun döndürölmüş durumları arasında en küçük Öklid uzaklık deęeri  $d$ , Eřitlik (3.11) deki uzaklık matrisi  $D$  'ye aktarılır:

$$D = \begin{bmatrix} d[1,1] & \cdots & d[1,m] \\ \vdots & \ddots & \vdots \\ \vdots & & \vdots \\ d[n,1] & \cdots & d[n,m] \end{bmatrix} \quad (3.11)$$

Her  $i$  ve her  $j$  için,  $Y_j$  görüntü parçası ile  $X_i$  mermer karosu arasındaki uzaklıęı en küçük yapan mermer karosu dönme açısının indisi  $o$ , Eřitlik (3.12)'deki  $A_{\mathcal{C}i}$  matrisine aktarılır.

$$A_{\mathcal{C}i} = \begin{bmatrix} A_{\mathcal{C}i}[1,1] & \cdots & A_{\mathcal{C}i}[1,m] \\ \vdots & \ddots & \vdots \\ \vdots & & \vdots \\ A_{\mathcal{C}i}[n,1] & \cdots & A_{\mathcal{C}i}[n,m] \end{bmatrix} \quad (3.12)$$

Mermer karoların yerleřtirme açılarını tutan  $A_{\mathcal{C}i}$  matrisinin bir elemanı  $A_{\mathcal{C}i}[i,j]$ ,  $i$ . mermer karosunun  $0^\circ, 90^\circ, 180^\circ, 270^\circ$  ( $o=0,1,2,3$ ) döndürölmüş durumlarından  $j$ . görüntü parçasına en yakın olanının  $o$  indisini içerir.

Uzaklık matrisinin kişisel bilgisayar ile hesaplanması oldukça fazla işlem gerektirmekte ve tamamlanması uzun zaman almaktadır. Günümüzde kişisel bilgisayar işlemcileri çok çekirdekli ve aynı anda birden fazla işlem gerçekleřtiren yapıdadır. Ancak geleneksel programlama yöntemleri ile yazılan programlar çok fazla hesap yüküne sahip olsa da

işlemcinin ancak %17-%20 iş yükü ile çalıştırabilmekte ve sürecin tamamlanması uzun zaman almaktadır. Çözüm, hesap yükünü eş zamanlı olarak çalışacak iş parçalarına bölerek, işlemcinin çekirdeklerinde paralel olarak çalıştırılmasını sağlamaktır. Microsoft .net'in System.Threading.Tasks sınıfı kütüphanesi bu amaçla yazılmış sınıflar barındırmaktadır. Bu kütüphaneden Parallel.ForEach sınıfı, iş parçalarını paralel olarak işlemciye vermek için kullanılmıştır. Öncelikle Parallel.ForEach döngüsü içinde gerçekleştirilecek görevin, birbirinden bağımsız ve paralel olarak hesaplanabilir olması gerekmektedir. Partitioner sınıfının Create metodu ile karolar ve görüntü parçaları 48 elemanlı bölütlere ayrılır. Bir bölütün ilk elemanı item1, son elemanı item2 dir. Her bölüt bir iş parçasına karşılık gelmektedir. İş parçası içinde item1. elemandan, item2. elemana kadar hesaplamalar normal for döngüsü ile gerçekleştirilmektedir. Parallel.ForEach, işlemci çekirdeklerini meşgul edecek şekilde birden fazla iş parçasını işlemciye gönderir, tüm iş parçaları tamamlanana kadar, işi biten iş parçası yerine yenisini gönderir. Şekil 3.8'de uzaklık matrisinin hesaplanması için iç içe geçmiş iki paralel foreach döngüsü görülmektedir. Görüntü parçası sayısı:  $m$ , karo sayısı:  $n$  olmak üzere  $i$ . karonun  $j$ . görüntüye Öklid uzaklığını hesaplayan  $SSD(i, j)$  fonksiyonudur. İşlemci yükü %99'dur.

```

var parca_n = System.Collections.Concurrent.Partitioner.Create(0, n, 48)
var parca_m = System.Collections.Concurrent.Partitioner.Create(0, m, 48)
Parallel.ForEach(parca_n, aralik_n =>
{
    Parallel.ForEach(parca_m, aralik_m =>
    {
        for (int i = aralik_n.Item1; i < aralik_n.Item2; i++)
            for (int j = aralik_m.Item1; j < aralik_m.Item2; j++)
                SSD(i, j);
    });
});

```

**Şekil 3.8:** İç içe geçmiş iki paralel foreach döngüsü.

### 3.5 Atama

$n$  adet mermer karonun,  $m$  adet görüntü parçasına, birebir ve toplamda en az hata ile eşleştirilmesi problemi, bir Atama Problemidir (Assignment Problem AP).  $i \in \{1, 2, \dots, n\}$  ve  $j \in \{1, 2, \dots, m\}$  olmak üzere  $i$ . mermer karonun,  $j$ . görüntü parçasına uzaklığı  $d_{ij}$  ve  $d_{ij}$ 'lerin tümü  $D$  uzaklık matrisi olarak verilmektedir. Atamada amaç, her görüntü parçasını

mümkün olan en küçük  $d_{ij}$  değerine sahip karo ile eşleştirmek ve atamaların tamamının toplam hatası için global minimumu bulmaktır. Böylelikle en uygun yapı taşları ile dizme hedefine ulaşılmış, optimum çözüm bulunmuş olunacaktır. Uygulamada, mermer karo ve görüntü parçası sayılarının birbirine eşit olmadığı durumlar ile karşılaşmak sıklıkla mümkündür. Bu duruma; “dengesiz atama problemi” denir ve modele, uzaklıkları  $d_{ij}=0$  sıfır olan sözde mermer karo ya da görüntü parçaları eklenerek problem dengeli hale getirilir. Böylelikle eşleşecek mermer karo sayıları ile görüntü parça sayıları birbirine eşitlenerek ( $n=m$ ), uzaklık matrisi, kare matris haline getirilir. Atama probleminin doğrusal programlama modelinde  $x_{ij}$  karar değişkenleri Eşitlik (3.14)’te verildiği gibidir [31]:

$$x_{ij} = \begin{cases} 1 & \text{Eğer } i. \text{ mermer karo, } j. \text{ görüntü parçasına atanacak ise} \\ 0 & \text{Eğer } i. \text{ mermer karo, } j. \text{ görüntü parçasına atanmayacak ise} \end{cases} \quad (3.14)$$

Modelin minimize edilmeye çalışılan Amaç Fonksiyonu Eşitlik (3.15) [31]:

$$z = \min \left( \sum_{j=1}^m \sum_{i=1}^n d_{ij} x_{ij} \right) \quad (3.15)$$

Kısıt: i. mermer karo, bir görüntü parçasına atanmalıdır Eşitlik (3.16) [31]:

$$\sum_{j=1}^n x_{ij} = 1 \quad 1 \leq i \leq n \quad (3.16)$$

Kısıt: j. görüntü parçasına bir mermer karo atanmalıdır Eşitlik (3.17) [31]:

$$\sum_{i=1}^n x_{ij} = 1 \quad 1 \leq j \leq n \quad (3.17)$$

### 3.5.1 Macar Yöntemi

Macar yöntemi, atama problemini çözen kombinatoryal optimizasyon algoritmasıdır. 1955 yılında “Macar yöntemi” adını veren Harold Kuhn tarafından geliştirilmiş ve yayınlanmıştır. Algoritma büyük ölçüde iki Macar matematikçi Dénes Könyig ve Jenő Egerváry’nin eski çalışmalarına dayandığı için bu şekilde adlandırılmıştır. Sade, kolay anlaşılabilen etkili bir çözüm yöntemidir. Daha sonra James Munkres 1957’de algoritmayı gözden geçirerek,



iyileştirmiştir ve hesaplama zaman karmaşıklığının bir polinom ile ifade edilebileceğini (polinom zamanlı algoritma) göstermiştir [32]. Bu nedenle Kuhn – Munkres algoritması veya Munkres atama algoritması olarak da bilinir. Orijinal algoritmanın zaman karmaşıklığı  $O(n^4)$  olmasına karşın, Edmonds ve Karp ve bağımsız olarak Tomizawa,  $O(n^3)$  çalışma süresine ulaşabileceğini bulmuşlardır.  $O(n^3)$  varyantlarından biri Jonker-Volgenant (LAPJV) algoritmasıdır. Ford ve Fulkerson çözümü, Ford-Fulkerson Algoritması şeklinde, genel maksimum akış problemlerine genişletmişlerdir [33].

Atama probleminin çözümünde Macar yönteminden on kat hızlı çalışan LAPJV algoritması [34], iki parçalı olmayan graflara da uygulanabilen Blossom Algoritması kullanılmaktadır. Çalışmada Macar yöntemi, atama probleminin en çok bilinen çözümü olduğu ve mozaik dizme otomasyon sistemi için geliştirilen programa kolayca eklenebildiği için tercih edilmiştir. Bunun yanında atama probleminin çözümünde sezgisel ve meta sezgisel yöntemler optimuma yakın sonuçlar üretmelerine karşın, sonucun optimuma ne kadar yakın olduğunun bir ölçüsü olmadığı için kullanılmamışlardır.

Macar yönteminin, maliyet matrisi üzerinde tekrarlı olarak bir dizi matris işlemleri ile gerçekleştirilen bir çözümü vardır. Graf Teorisini kullanan, biri  $O(n^4)$  karmaşıklığı ile gerçekleştirmesi kolay ve hızlı, diğeri  $O(n^3)$  karmaşıklığı ile gerçekleştirmesi zor iki uygulaması daha mevcuttur.

### 3.5.2 Maliyet Matrisi ile Macar Yöntemi

Eşitlik (3.18)'de görülen, atamaların maliyetlerinden oluşan matris, *orijinal maliyet matrisi* olarak adlandırılmaktadır. Orijinal maliyet matrisi  $n$  boyutlu kare bir matristir.

$$D = \begin{bmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{n1} & \cdots & d_{nn} \end{bmatrix} \quad (3.18)$$

Macar yöntemini, maliyet matrisi üzerinde çeşitli matris işlemleri yaparak gerçekleştiren algoritma aşağıdaki gibidir [31]:

**Adım 1:** Orijinal maliyet matrisinin her bir satırının minimum değerini bul ve bu değeri ilgili satırdaki tüm elemanlardan çıkart. Bu, satırdaki en düşük maliyeti 0 yapar.

**Adım 2:** Adım 1'in uygulanması ile elde edilen maliyet matrisinin her bir sütununun minimum değerini bul ve bu değeri ilgili sütundaki tüm elemanlardan çıkart. Bu her bir kolondaki en küçük maliyeti 0 yapar.

Orijinal maliyet matrisine Adım 1 ve Adım 2'nin uygulanması ile elde edilen, her satır ve sütununda en az bir tane sıfır değeri içeren yeni maliyet matrisi *indirgenmiş maliyet matrisi* olarak adlandırılır.

**Adım 3:** İndirgenmiş (veya son indirgenmiş) maliyet matrisinde yer alan sıfırların hepsini kapatacak şekilde minimum sayıda yatay veya düşey çizgiler çiz.

**Adım 3a:** Eğer sıfırların tümünü kapatan minimum çizgi sayısı, orijinal maliyet matrisinin satır (veya sütun) sayısı  $n$ 'e eşit ise minimum maliyetli atamaya ulaşılmıştır. Sıfırların bulunduğu hücrelere karşılık gelen iş/işçi eşleşmesi optimal atamadır. Dur.

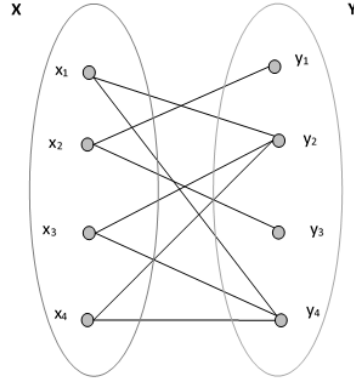
**Adım 3b:** Aksi taktirde, sıfırların tümünü kapatan minimum çizgi sayısı, orijinal maliyet matrisinin satır sayısı  $n$ 'den küçük ise minimum maliyetli atamaya ulaşılmamıştır. Adım 4'e git.

**Adım 4:** Üzeri çizilerek kapatılmamış elemanlar içinden en küçük maliyeti seç ve bu elemanı, üzeri çizilmemiş bütün elemanlardan çıkar. Aynı elemanı, iki kapatma çizgisinin kesiştiği yerdeki elemana/elemanlara ekle. Bu işlemler sonucunda oluşan *son indirgenmiş maliyet matrisi* olarak adlandırılan matrisi elde et ve Adım 3'e git.

Algoritma şu temel fikir üzerinde çalışmaktadır, orijinal maliyet matrisinin satır ve sütunlarından belirli kurallara göre bazı değerlerin çıkartılıp eklenmesiyle oluşan indirgenmiş maliyet matrisi için en uygun atama, aynı zamanda orijinal maliyet matrisi için de en uygun atamadır [31].

### 3.5.3 İki Parçalı Graf (Bipartite Graf) Kullanarak Macar Yöntemi

Macar yöntemi, iki parçalı grafta (Şekil 3.9), kararlı maksimum (veya minimum) ağırlık eşleşmesi bulmak için kullanılabilir, iki parçalı grafın ağırlıkları değiştirilerek Macar yöntemi çalıştırılabilir.



Şekil 3.9: İki parçalı Graf  $G_{4,4}$

Bu mükemmel eşleşen bir grafa “uygulanabilir etiketleme” yapılarak bulunabilir. Mükemmel eşleşme, her düğümün eşleşmenin bir bağlantısına sahip olduğu anlamına gelir. İki parçalı graflar, grafi oluşturan düğümler kendi aralarında bağlantı olmayan fakat karşısındaki kümedeki düğümlerle bağlantısı olan iki farklı düğüm kümesinden oluşur. Bu kümelerin eleman sayıları  $m$  ve  $n$  ise iki parçalı graf  $G_{m,n}$  şeklinde gösterilir [32].

İki parçalı grafi oluşturan iki kümeden birinin her bir düğümü ile diğer kümenin her bir düğümü bir hat ile birleştirilmiş ise bu grafa *iki parçalı tam graf* (complete bipartite graph) denir [35]. İki parçalı tam graf Şekil 3.10(b)’de görülmektedir.



Şekil 3.10: a) İki parçalı graf  $G_{4,3}$  b) İki parçalı tam graf  $K_{4,3}$

*Maksimum eşleşme* (Maximum Matching) bir grafta bulunabilecek en fazla eşi arar. *Mükemmel Eşleşme* (Perfect Matching) eğer bir graftaki bütün düğümler bir başka düğüm ile eşleşmiş ve eşleşmemiş düğüm kalmamış ise bu eşleşmeye mükemmel eşleşme denir [32].

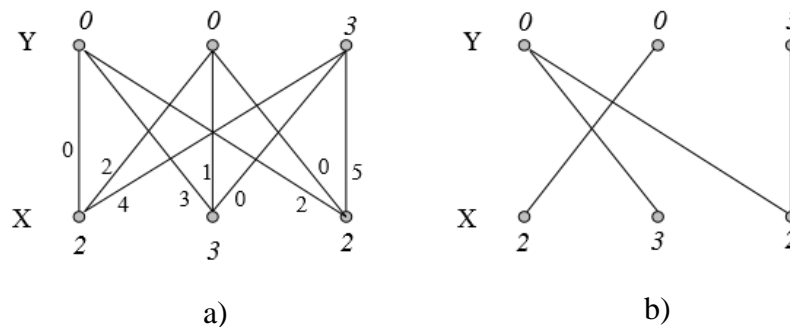
$G = (V, E)$  tam bir “iki parçalı graf” olarak göz önüne alındığında, sol taraftaki düğümler kümesi  $X$ , sağ taraftaki düğümler kümesi  $Y$ , tüm düğümler  $V = X \cup Y$ , bağlantılar  $E \in X \times Y$  ve iki tarafın kesişim kümesi  $X \cap Y = \emptyset$  boş kümedir.  $(x,y)$  bağlantısının ağırlığını  $w(x,y)$  olarak gösterilsin. Bir  $v$  düğümünün komşuluğu,  $v$  ile bir bağlantı paylaşan tüm düğümleri içeren  $J_G(v)$  kümesi ve  $S$  kümesinin komşuluğu,  $S$  de bir düğüm ile bağlantı paylaşan tüm düğümleri içeren küme  $J_G(S)$  dir. Bir  $M (M \subseteq E)$  eşleşmesi göz önüne alındığında,  $v$  düğümü eğer  $M$ 'ye aitse *eşleşmiş düğüm* denir, aksi takdirde *serbest düğüm* olarak adlandırılır [35].

Uygulanabilir etiketleme fonksiyonu  $l: V \rightarrow \mathbb{R}$  graftaki her düğüme bir sayı atar. Eşitlik (3.19)'da ifade edilen, iki taraftan birer düğüm  $x$  ve  $y$  'yi birleştiren bağlantının ağırlığı  $w(x,y)$ ,  $x$  ve  $y$ 'nin etiketleri toplamını geçemez. Bu üçgen eşitsizliği gibi düşünülebilir [36]:

$$l(x) + l(y) \geq w(x, y), \quad \forall x \in X, \forall y \in Y \quad (3.19)$$

$G$  'nin altgrafı olan  $G_l \subseteq G$  aynı zamanda eşitlik grafi:  $G_l = (V, E_l)$ , uygulanabilir etiketleme fonksiyonu  $l$  ile etiketlenmiş orijinal grafın “iki taraftan birer düğümün etiketleri toplamı, düğümleri birleştiren bağlantının ağırlığına tam olarak eşittir” Eşitlik (3.20) koşulunu sağlayan bağlantıları içerir [37]. Etiketlenmiş graf, Şekil3.11(a)' da, eşitlik grafi Şekil3.11(b) de görülmektedir.

$$E_l = \{ (x, y) \in E \mid l(x) + l(y) = w(x, y) \} \quad (3.20)$$



Şekil 3.11: a) Düğüm Etiketleme. b) Eşitlik grafi.

Teorik olarak, eğer  $G_l$  eşitlik grafında **mükemmel bir eşleşme**  $M^*$  ise, o zaman  $M^*$ ,  $G$  de (optimal) maksimum ağırlıklı bir eşleştirmedir. Şimdi,  $M^*$ 'dan daha büyük ağırlıkta

mükemmel bir eşleşme olmadığını Eşitlik (3.21)'de görülmektedir.  $M$ 'nin verilen herhangi bir mükemmel eşleşme olduğunu varsayalım, o zaman

$$\begin{aligned}
 w(M) &= \sum_{x,y \in M} w(x,y) \leq \sum_{x,y \in M} \{l(x) + l(y)\} \\
 &= \sum_x l(x) + \sum_y l(y) = \sum_{x,y \in M^*} \{l(x) + l(y)\} \\
 &= \sum_{x,y \in M^*} w(x,y) = w(M^*)
 \end{aligned} \tag{3.21}$$

$M^*$  maksimum mükemmel eşleşmedir. Bu nedenle Kuhn-Munkres algoritması global optimal değere ulaşmayı garanti eder [37].

Sayısal olarak, Algoritma  $l$  'nin zaman karmaşıklığı  $O(mn^2)$  'dir, burada  $m$ ,  $X$  'deki eleman sayısıdır ve  $n$ ,  $Y$  'deki eleman sayısıdır.

Eşitlik grafında mükemmel eşleşme, orijinal grafta maksimum eşleşmeye karşılık gelir. Optimal eşleşmeyi bulma problemi, eşitlik grafında mükemmel eşleşmeyi bulmaya indirgenmiştir. İki taraf  $X$  ve  $Y$  'nin düğüm sayıları eşit olmalıdır. Eksik bağlantıların ağırlığı sıfır olarak eklenmelidir.

Macar yöntemi minimum maliyet ile atamayı hesaplar. Burada anlatılan maksimum ağırlıklı eşleşme ise eşleşmeyi sağlayan bağlantıların ağırlıklarının toplamının maksimum olmasını sağlar. Bu nedenle problemi maksimum ağırlıklı eşleşme probleminden, minimuma dönüştürerek yeniden modellemek gerekir [35]. Bu amaçla Eşitlik (3.22) ile ağırlıklar değiştirilir:

$$w(x,y) = \max(w(x,y)) - w(x,y), \forall (x,y) \in E \tag{3.22}$$

Orijinal Kuhn-Munkres algoritması [37]:

Girdi: İki parçalı graf  $G = (V, E)$  ve ağırlık  $w(x, y)$ .

Çıktı: Optimal mükemmel eşleşme  $M$ .

**Adım 1:**  $G$ 'de bir başlangıç etiketlemesi  $l$  ve eşleşmesi  $M$  'yi oluştur.

**Adım 2:** Eğer  $M$  mükemmel ise, dur. Aksi takdirde  $u \in X$  serbest düğüm seç.  $S = \{u\}$ ,  $T = \emptyset$  ata.

**Adım 3:** Eğer  $J_l(S) = T$  ise, etiketleri güncelle ( $J_l(S) \neq T$ 'yi zorla)

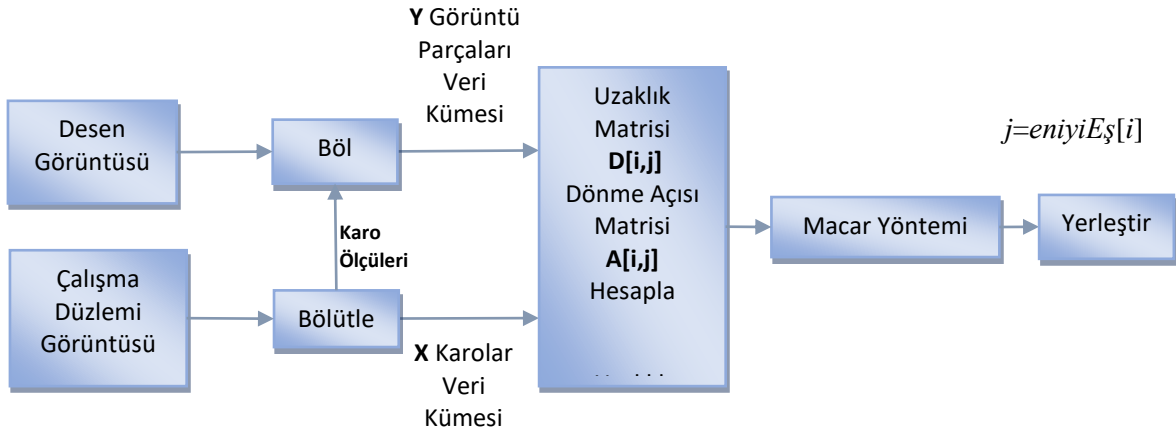
$$\alpha_l = \min_{s \in S, y \notin T} \{l(x) + l(y) - w(x, y)\}$$

$$\hat{l}(v) = \begin{cases} l(v) - \alpha_l & v \in S \\ l(v) + \alpha_l & v \in T \\ l(v) & \text{diğer} \end{cases}$$

**Adım 4:** Eğer  $J_l(S) \neq T$  ise,  $y \in J_l(S) - T$  seç:

- Eğer  $y$  serbest ise,  $u$ - $y$  uzatılmış yoldur.  $M$ 'yi uzat ve **Adım2**'ye git.
- Eğer  $y$  eşleşmiş ise örneğin  $z$  ile, alternatif ağacı genişlet:  $S = S \cup z$ ,  $T = T \cup y$  **Adım3**'e git.

Macar yöntemine uzaklık matrisi  $D[i,j]$  girdi olarak verilir. Macar yönteminin çıktısı en küçük toplam hatayı veren  $j=eniyeEş[i]$  dizisidir. Dizi,  $i$ . karonun eşleştiği  $j$ . görüntü parçasını verir. Problemin çözümüne ilişkin blok diyagram Şekil 3.12'de görünmektedir.



**Şekil 3.12:** Tüm karoların aynı ölçüde olduğu durumda görüntü parçalarına atama.

### 3.6 Farklı Ölçülerde Karoların (8, 16, 32 piksel), Görüntü Parçalarına Atanması

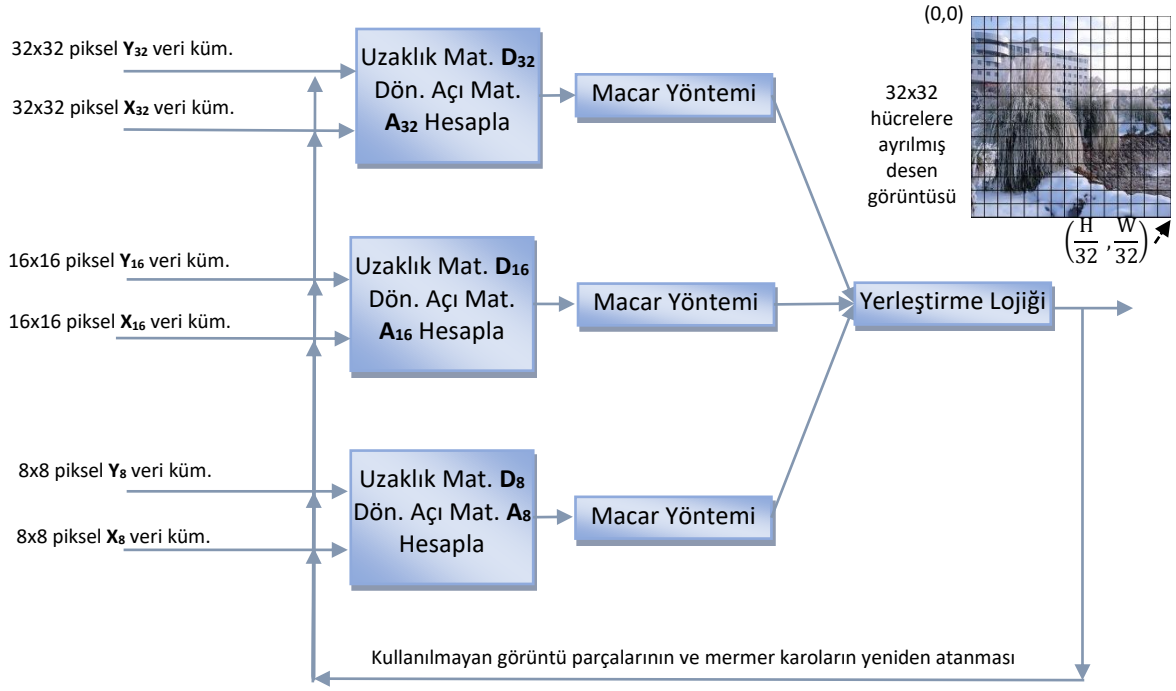
Mermer karolar veri kümesinde çeşitli ölçülerde; örneğin 8, 16 ya da 32 piksel karolar olduğunda, eşleşen uygun görüntü parçalarının bulunabilmesi için, mozaïği dizilecek görüntünün aynı ölçülerde hücrelere sahip ızgaralarla parçalanarak 8, 16, 32 piksel görüntü parçalarından oluşan veri kümelerinin oluşturulması gereklidir.

Atamadan önce mermer karoların, çalışma düzlemi üzerinde konum bilgileri vardır ancak desen görüntüsü oluşturmak için nereye yerleştirilmeleri gerektiğinin bilgisi yoktur. Bir mermer karo, bir görüntü parçasına atandığında, onun dizilecek görüntü üzerindeki konum bilgisine de sahip olur.

Hücreleri 8, 16 ve 32 piksel ölçülerinde olan üç katman (üç görüntü parçası veri kümesi) ile çalışıldığından şu problemlerin aşılması gerekmektedir:

1. Katmanların birinde ya da ikisinde hiç karo olmayabilir.
2. Üç farklı katmanda, dizilecek görüntüde en uygun konuma yerleştirilen karolar, aynı konumlara denk düşüp diğer katmanlardaki karolarla örtüşebilirler.
3. 8, 16 piksel küçük karolar en uygun konuma yerleştğinde, yüzeyi tamamen kaplamayabilirler, boşluk kalabilir.

Bu problemlerin üstesinden gelinmesi için önerilen yöntemin blok diyagramı Şekil 3.13'de görülmektedir. Şekil 3.13'de Görüntü parçalarının oluşturduğu kümeler **Y**, karoların oluşturduğu kümeler **X** olarak adlandırılmıştır. Yöntem, 8, 16 ve 32 piksel ölçülerinde üç katmanda uzaklık matrislerini ayrı ayrı hesaplayarak ( $D_8$ ,  $D_{16}$ ,  $D_{32}$ ), Macar yöntemi ile atamalar yapar.

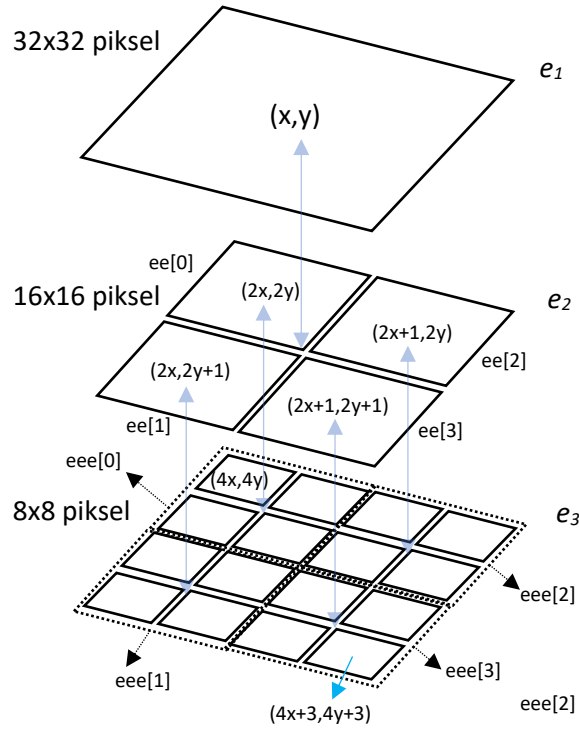


**Şekil 3.13:** 32x32 hücrelerin, 32,16 ve 8 piksel karoların atanması.

Daha sonra, yerleştirme lojîği, desen görüntüsünde 32x32 piksel hücrelere sahip ızgarayı temel alarak, bu ızgarada soldan sağa, yukardan aşağıya, sol üst köşeden sağ alt köşeye hücreleri dolaşır. Her bir hücre için, Macar yönteminin atadığı karolardan, varsa “1 tane 32” ya da “4 tane 16” ya da “16 tane 8 piksel” karo atanmışsa, hatası en az olan karo grubunu 32 piksellik hücreye yerleştirilmek üzere, kullanılmayan karolar silinmek üzere etiketler ve bir sonraki hücreye geçer. 32x32 piksellik hücrenin kaplanması şartıyla atanan karolar 8x8 ve 16x16 piksel karoların kombinasyonu da olabilir. Yerleştirilmek üzere etiketlenmiş karolar, görüntü parçaları ile eşlenerek “yerleşen karolar listesine aktarılır ve diğer katmanlarda kullanılmayan görüntü parçaları silinir. 32 piksellik hücreleri kaplamayı başaramayan geride kalan karolar, tekrar atanmak üzere uzaklık matrisi hesabına ve ardından Macar yöntemine girerler. Döngü, atanacak hiç görüntü parçası ya da mermer karo kalmayınca kadar devam eder.

32x32 piksel hücrelerle bölünen desen görüntüsünün her bir hücresinin, 32, 16 ve 8 piksel karo atamalarından en az hatalı olanlar ile kaplanması için karo ve karo gruplarının yerleştirme hatalarının adlandırılması Şekil 3.14 de görülmektedir.





**Şekil 3.14:** 32, 16 ve 8 piksel karo ve karo gruplarının yerleştirme hatalarının adlandırılması.

$e_1$ ,  $e_2$ ,  $e_3$  sırasıyla 32,16 ve 8 piksel katmanlarda, 32x32 piksel görüntü parçasına atanan mermer karoların toplam yerleştirme hatalarıdır.  $e_1$ : 32x32 piksel görüntü parçasına atanan mermer karosunun Öklid uzaklığıdır, Eşitlik (3.23) ile verilmiştir:

$$e_1 = D_{32}(x, y) \quad (3.23)$$

$e_2$ : Aynı alanı örten 16x16 piksel görüntü parçalarına atanan 4 adet mermer karonun Öklid uzaklıklarının toplamıdır, Eşitlik (3.24) ile verilmiştir:

$$e_2 = D_{16}(2x, 2y) + D_{16}(2x + 1, 2y) + D_{16}(2x, 2y + 1) + D_{16}(2x + 1, 2y + 1) \quad (3.24)$$

Algoritmada kolay okunması açısından 16x16 piksel karoların her birinin yerleştirme hatası Eşitlik (3.25), (3.26), (3.27), (3.28) ile verilmiştir:

$$ee[0] = D_{16}(2x, 2y) \quad (3.25)$$

$$ee[1] = D_{16}(2x, 2y + 1) \quad (3.26)$$

$$ee[2] = D_{16}(2x + 1, 2y) \quad (3.27)$$

$$ee[3] = D_{16}(2x + 1, 2y + 1) \quad (3.28)$$

$e_3$ : Aynı alanı örten 8x8 piksel görüntü parçalarına atanan 16 adet mermer karonun Öklid uzaklıklarının toplamıdır Eşitlik (3.29) ile verilmiştir:

$$e_3 = \sum_{i=0}^3 \sum_{j=0}^3 D_8(4x + i, 4y + j) \quad (3.29)$$

16x16 dört karo ile karşılaştırılmak üzere 8x8 karolardan dörderli dört grup oluşturulur ve grupların toplam yerleştirme hataları Eşitlik (3.30), (3.31), (3.32), (3.33) ile hesaplanır:

$$eee[0] = D_8(4x, 4y) + D_8(4x + 1, 4y) + D_8(4x, 4y + 1) + D_8(4x + 1, 4y + 1) \quad (3.30)$$

$$eee[1] = D_8(4x, 4y + 2) + D_8(4x + 1, 4y + 2) + D_8(4x, 4y + 3) + D_8(4x + 1, 4y + 3) \quad (3.31)$$

$$eee[2] = D_8(4x + 2, 4y) + D_8(4x + 3, 4y) + D_8(4x + 2, 4y + 1) + D_8(4x + 3, 4y + 1) \quad (3.32)$$

$$eee[3] = D_8(4x + 2, 4y + 2) + D_8(4x + 3, 4y + 2) + D_8(4x + 2, 4y + 3) + D_8(4x + 3, 4y + 3) \quad (3.33)$$

Macar yönteminin (x, y) konumuna karo atmadığı durumlarda, 32, 16 ve 8 piksellik karoların, görüntü parçalarına uzaklıklarına olası en büyük değerler  $D_{32}(x,y)=32^2 \times 255^2 \times 3$ ,  $D_{16}(x,y)=16^2 \times 255^2 \times 3$ ,  $D_8(x,y)=8^2 \times 255^2 \times 3$  atanır.

$saye_1$ ,  $saye_2$ ,  $saye_3$  sırasıyla 32,16,8 piksel katmanlarda, görüntü parçasına atanan mermer karoların sayısıdır. Yerleştirme Algoritması:

Mermer karo sayısı ya da görüntü parça sayısı sıfır olmadığı sürece ve tekrar sayısı 10'dan küçük olduğu sürece dön:

1. 8, 16 ve 32 piksel katmanlar için ayrı ayrı karoların uzaklık matrisi **D** ve yerleştirme açıları **A** matrislerini hesapla.
2. 8, 16 ve 32 piksel katmanların her biri için ayrı ayrı Macar yöntemi ile karo ataması yap.  $j=eniyiEş[i]$  kullanarak j. görüntü parçasına atanan i. karoya, j. görüntü parçasının ızgaradaki konumunu (x,y) aktar.
3.  $y:=0 \rightarrow H/32$ ,  $x:=0 \rightarrow W/32$ , 32 piksel hücrelere sahip ızgaranın desen görüntüsündeki her bir hücresi (y,x) için dön:
  - a.  $e_1$ ,  $e_2$ ,  $e_3$ ,  $ee$  ve  $eee$  hesapla. Sırasıyla 8, 16, 32 piksel katmanlarda birbirine karşılık gelen atama hatalarıdır.
  - b.  $saye_1$ ,  $saye_2$ ,  $saye_3$  hesapla. Sırasıyla 8, 16, 32 piksel katmanlarda, görüntü parçasına atanan mermer karoların sayısıdır.

**Yerleştirme Lojigi:**

- c. **Eğer  $e_1 < e_2$  ve  $e_1 < e_3$  ve  $saye_1 > 0$  ise:**  $e_1$  en küçük ve 1 tane 32 piksel karo atanmış ise, 32 piksel görüntü parçasını yerleştirilecek, 8 ve 16 piksel görüntü parçalarını silinecek olarak etiketle (32 piksel kaplanması durumu).
- d. **Değil ise eğer  $e_2 < e_1$  ve  $e_2 < e_3$  ve  $saye_2 > 3$  ise:**  $e_2$  en küçük ve 4 tane 16 piksel karo atanmış ise, 16 piksel 4 adet görüntü parçasını yerleştirilecek, 8 ve 32 piksel görüntü parçalarını silinecek olarak etiketle (16 piksel kaplanması durumu).
- e. **Değil ise eğer  $e_3 < e_1$  ve  $e_3 < e_2$  ve  $saye_3 > 15$  ise:**  $e_3$  en küçük ve 16 tane 8 piksel karo atanmış ise, 8 piksel 16 adet görüntü parçasını yerleştirilecek, 16 ve 32 piksel görüntü parçalarını silinecek olarak etiketle (8 piksel karo ile kaplanması durumu).
- f. **Değil ise eğer  $(saye_2 + saye_3/4) \geq 4$  ise** (8 ve 16 piksel koraların kombinasyonu ile kaplanması durumu):
- i. **Eğer  $ee[0] < eee[0]$  ise** 16 piksel görüntü parçalarının  $\{(4x, 4y), (4x, 4y+1), (4x+1, 4y), (4x+1, 4y+1)\}$  etiketi “**yerleştir**”, 8 piksel görüntü parçasının  $(2x+2y)$  etiketine “**sil**” ata. **Değil ise** 8 piksel görüntü parçasının etiketine “**yerleştir**”, 16 piksel görüntü parçalarının etiketine “**sil**” ata.
  - ii. **Eğer  $ee[1] < eee[1]$  ise** 16 piksel karoların  $\{(4x, 4y+2), (4x+1, 4y+2), (4x, 4y+3), (4x+1, 4y+3)\}$  etiketine “**yerleştir**”, 8 piksel karonun  $(2x, 2y+1)$  etiketine “**sil**” ata. **Değil ise** 8 piksel karonun etiketi “**yerleştir**” 16 piksel karoların etiketi “**sil**” ata.
  - iii. **Eğer  $ee[2] < eee[2]$  ise** 16 piksel karoların  $\{(4x+2, 4y), (4x+3, 4y), (4x+2, 4y+1), (4x+3, 4y+1)\}$  etiketine “**yerleştir**”, 8 piksel karonun  $(2x+1, 2y)$  etiketine “**sil**” ata. **Değil ise** 8 piksel karonun etiketine “**yerleştir**”, 16 piksel karoların etiketine “**sil**” ata.
  - iv. **Eğer  $ee[3] < eee[3]$  ise** 16 piksel karoların  $\{(4x+2, 4y+2), (4x+3, 4y+2), (4x+2, 4y+3), (4x+3, 4y+3)\}$  etiketine “**yerleştir**”, 8 piksel karonun  $(2x+1, 2y+1)$  etiketine “**sil**” ata.

**Değil ise** 8 piksel karonun etiketine “**yerleştir**” 16piksel karoların etiketi “**sil**” ata.

Yerleştirilecek olarak etiketlenen karoları yerleştir. Silinecek olarak etiketlenen görüntü parçalarını sil. Tekrar sayısını bir arttır. Döngünün başına dön.

### 3.7 Nesne İçeren Görüntü Parçalarının Etiketlenmesi

Görüntü işaretleri yüksek yapıdır. Görüntü parçaları komşularıyla güçlü bağımlılığa sahiptir ve bağımlılık görsel sahnedeki nesnelerin yapıları ile ilgili bilgi taşır. İnsan görme sistemi, görsel sahnelerden yapısal bilgi çıkarmak için yüksek derecede evrim geçirmiş ve uyum sağlamıştır. Görüntüde arka plandan nesnelere ayırt ederek, her bir nesneyi yorumlar ve izler [38].

Macar yöntemi, desen görüntü parçalarının tamamına atanan karoların uzaklıklarının toplamını minimize etmektedir. Görüntünün ilgi odağındaki nesne ile arka plan aynı önemde değerlendirilmekte ve daha çok yapısal bilgi içeren nesne kötü dizilebilmektedir. Eğer görüntüde nesne içeren bölgeler işaretlenir ve bu bölgelere karo atama işlemi sırasında arka plana göre öncelik verilirse, sınırlı sayıdaki karo stoku daha etkili kullanılabilir. Yöntem böyle bir dar boğazı aşmak için önerilmiştir.

Görüntüde insan algısının yöneldiği ilgi odaklarını tespit etmek ve mermer karo atama sırasında bu noktalara öncelik vermek için yapay zeka kullanılarak bu bölgeler etiketlenir ve ardından bir arayüz ile kullanıcıyla etkileşime girilerek, ihtiyaçları doğrultusunda son hali vermesi sağlanır.

Çalışmada yapay zeka kullanılarak, görüntüde algılanan nesnelere kesilen her görüntü parçası etiketlenir. Görüntü parça indeksi:  $0 \leq j < m$  olmak üzere  $j$ . görüntü parçası bir nesne ile kesişiyorsa, Eşitlik (3.34)'deki gibi  $p_j=1$  ile etiketlenir. Nesne içermeyen arka plan görüntü parçaları  $p_j=k$  ile etiketlenir.  $k=2000$  alınmıştır.  $k$  değeri büyüdükçe, arka plana atanan karoların benzerliği azalır, yerleştirme hatası artar.

$$p_j = \begin{cases} 1 & | \text{ } j. \text{görüntü parçası nesne içeriyorsa} \\ k & | \text{ } j. \text{görüntü parçası nesne içermiyorsa} \end{cases} \quad (3.34)$$

Mermer karoların, görüntü parçalarına Uzaklık Matrisi  $D$ , elemanları  $d_{ij}$  olmak üzere, matrisin  $j$ . kolonundaki tüm elemanlar,  $p_j$ 'ye bölünerek, görüntü parçalarının nesne içerip içermemelerine göre uzaklık matrisi Eşitlik (3.35)'de görüldüğü gibi yeniden yapılandırılır:

$$d'_{ij} = \frac{d_{ij}}{p_j} \quad 1 \leq i < n, \quad 1 \leq j < m \quad (3.35)$$

Böylelikle, güncellenmiş uzaklık matrisi  $D'$  elemanları  $d'_{ij}$  'ye, nesne içeren görüntü parçalarına atanan mermer karoların uzaklıkları olduğu gibi aktarılır. Arka plana ait nesne içermeyen görüntü parçalarına atanabilecek mermer karoların tümünün uzaklığı  $k$  kat azaltılmış olarak aktarılır. Diğer bir deyişle, nesne içermeyen arka plan görüntü parçaları, tüm mermer karolara yakınlaştırılmıştır.

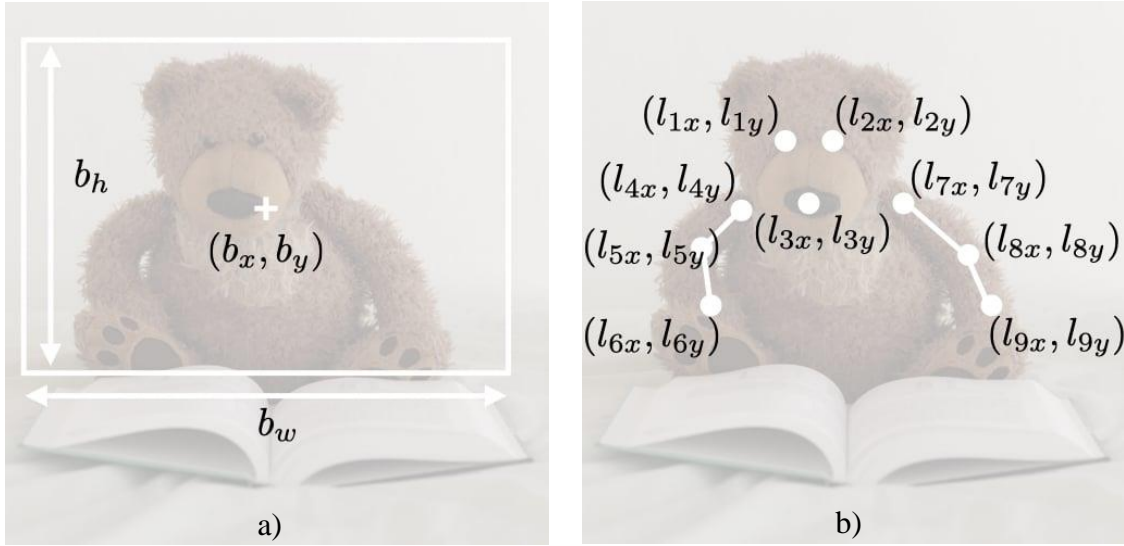
### 3.7.1 Derin Öğrenme Temelli Nesne Detektörleri

Nesne algılama, görüntüde veya videoda nesnelere tanıyan, yerini ve pozunu tespit edebilen bilgisayarlı görü yöntemleridir. Görüntü işleme ve makine öğrenmesi yöntemlerini kullanırlar. Işık şiddetindeki değişimler, nesnenin kameraya uzaklığı ve kameranın açısı gibi birçok faktör nesne tespitini olumsuz yönde etkiler. Başarımın artırılması için son yıllarda özellik çıkartma işlevinin de nöron ağı tarafından üstlenildiği derin öğrenen sistemler kullanılmaya başlanmıştır. Gerçek zamanlı nesne tespit eden sistemler, derin öğrenen evrişimli nöron ağları kullanılarak geliştirilmiştir. Bu gerçek zamanlı sistemlerin, otonom sürüş, güvenlik kamerası izleme, yüz tanıma, nesne takibi, hava görüntülerinin analizi, imalat ve robotik sistemler gibi birçok uygulama alanı mevcuttur.

Görüntüdeki nesnelere algılamak ve nesne içeren bölgeleri etiketlemek için YOLO V3 evrişimli nöron ağı kullanılmıştır. YOLO insandan sörf tahtasına 80 değişik nesneyi algılayabilmekte ve dikdörtgen bir çerçeve ile sınırlayabilmektedir. Görüntüdeki çok sayıda insan pozunu algılamak için OpenPose kullanılmıştır. Yakın çekim insan görüntülerinde yüz ve ellerin anahtar noktalarını algılamak için OpenPose yüz ve el kullanılmıştır. Tüm bunlara ek olarak görüntünün tamamında birden fazla insan yüzü algılama özelliğine sahip Dlib yüz detektörü uygulanmıştır.

Nesne algılama, nesnenin sadece konumunu bulmak ya da yönelişi, poz gibi daha karmaşık bir durumu tespit etmek üzere başlıca iki ana gruba ayrılır [39]. “Sınırlayıcı kutu ile

algılama”; Şekil 3.15(a)’daki gibi, görüntüde nesnenin bulunduğu konum, sınırlayıcı bir kutu ile ifade edilir. “Anahtar Noktalar ile algılama” Şekil 3.15(b)’de görüldüğü gibi, bir nesnenin şekli, yönelişi, pozu vb. özellikleri referans noktalar ile temsil edilir.

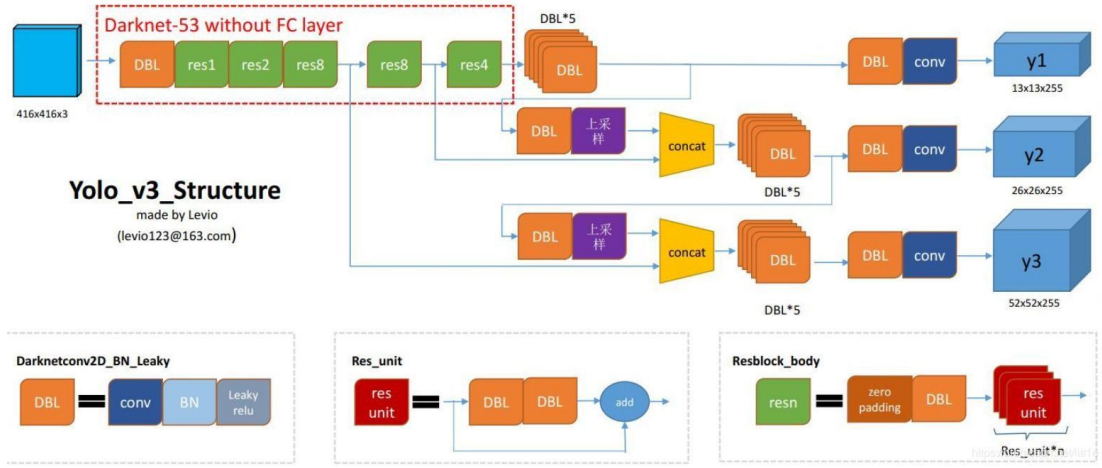


**Şekil 3.15:** Nesne detektörleri a) Sınırlayıcı kutu ile nesne algılama. b) Anahtar noktalar ile nesne algılama [39].

Aralık 2015 de, OpenCV 3.1 sürümü ile birlikte derin nöron ağlarını (DNN) desteklemeye başlamıştır. OpenCV’nin ayrıca Python dili için yazılım kütüphanesi, paketleri bulunmaktadır. Visual Studio Python ortamından OpenCV paketlerini kullanarak görüntü işleme, bilgisayarlı görü ve derin nöron ağları uygulamaları yazıp çalıştırmak çok daha kolay hale gelmiştir. Çalışmamızda .net’in “Process” sınıfını kullanarak C# Windows Form uygulamasından, Python ile yazılmış evrişimli nöron ağı (CNN) uygulamaları çalıştırılmış, çıktıları Form uygulamasına alınarak işlenmiştir.

### 3.7.2 YOLO, Nesne Detektörü

YOLO nesneleri gerçek zamanlı olarak tespit edebilen, rakiplerinden doğruluk ve hız bakımından daha avantajlı olan tek aşamalı nesne detektörüdür. “You Only Look Once: Sadece bir kez bakarsın” baş harflerinden oluşan bir kısaltmadır [40]. Çalışmada COCO veri kümesi ile eğitilmiş YOLO V3 kullanılmıştır (Şekil 3.16).



Şekil 3.16: YOLO V3 (You Only Look Once- Version3) nesne detektörü boru hattının basitleştirilmiş blok diyagramı [41].

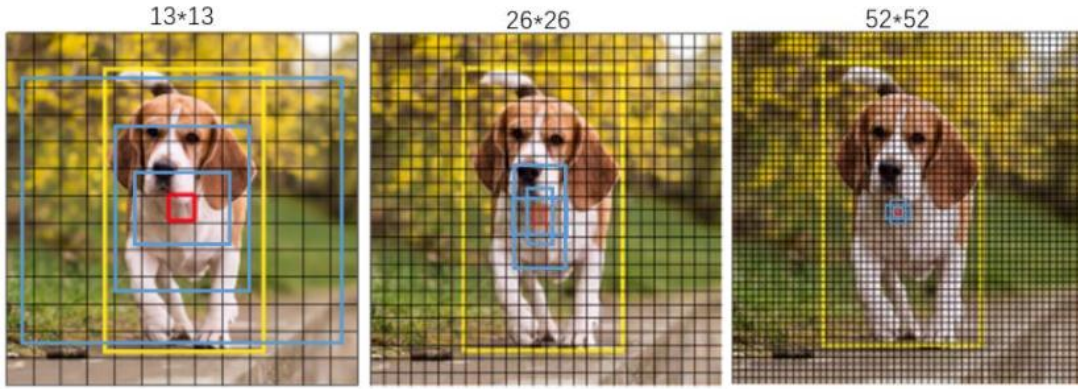
### 3.7.2.1 YOLO V3 Ağının Girişi

YOLO V3 ağının girişine RGB renk uzayında kare şeklinde bir kenarı 32'nin katı olacak görüntüler uygulanır. Uygulamamızda önceden eğitilmiş ağlar: 320x320, 416x416 veya 608x608 seçeneklerinden 320x320 ağırlık dosyası kullanılmıştır. Bu durumda YOLO V3 ağının girişine uygulanacak görüntü ölçeklenerek 320x320 piksel ölçülerine indirgenmelidir [42].

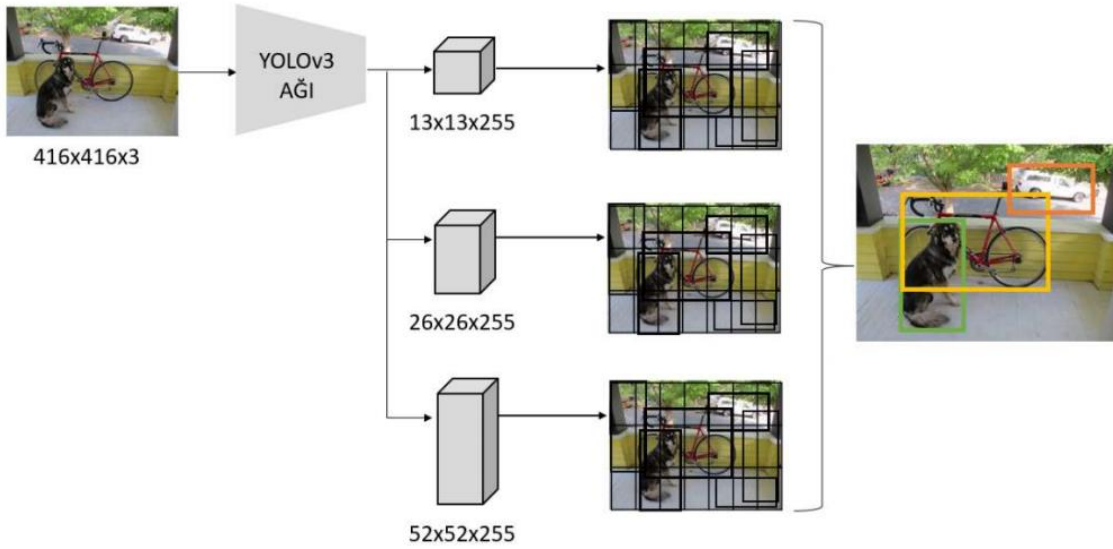
### 3.7.2.2 YOLO V3 Ağının Çıkışı

YOLO V3 ağının girişine uygulanan görüntünün, çıkışında Şekil 3.17(a)'da görüldüğü gibi 3 farklı ölçekte özellik haritası çıkarılmıştır. Büyük boyutlu nesnelere algılamak için 13x13x255, orta boyutlu nesnelere için 26x26x255 ve küçük boyutlu nesnelere algılamak için 52x52x255 ölçülerinde özellik haritaları işlenerek nesnelere konumları ve ait oldukları sınıflar elde edilir.

Şekil 3.17(b)'de gösterilen 13x13x255 çıkışında, görüntü yatayda 13 ve dikeyde 13 parçaya bölünmüştür. Hücrelerin her biri için 3 adet öneri (anchor) kutusu tahmin edilmektedir. Ağın aynı anda birden fazla nesneyi tahmin etmesine olanak verir.



a) Girişe uygulanan görüntü; 13x13, 26x26 ve 52x52 parçalara bölünür.

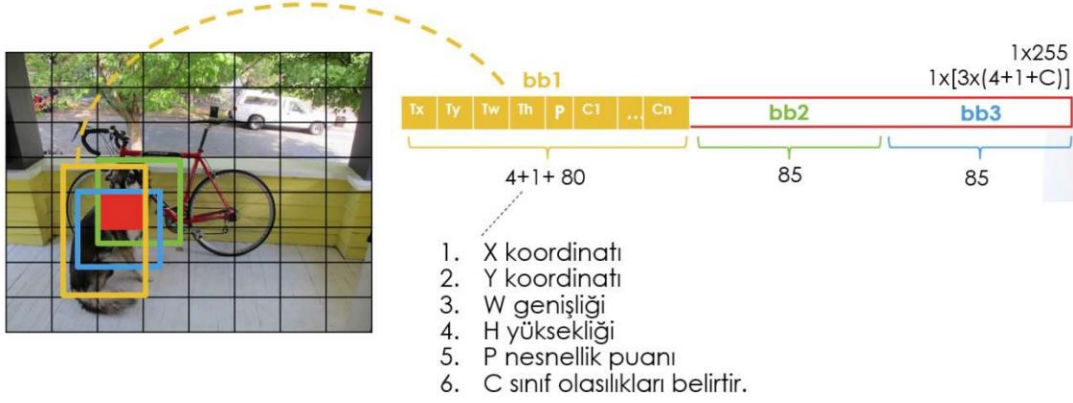


b) YOLOV3 Ağının Çıkışı

**Şekil 3.17:** YOLO V3 ağının giriş ve çıkışı a) Ağın girişine uygulanan görüntünün, üç farklı ölçekte özellik çıkarımı için parçalara ayrılması ve b) her bir görüntü parçası için çıkarılan 255 adet özellik [42].

Şekil 3.18'de kırmızı ile belirtilen hücre için sarı, yeşil ve mavi renkte 3 tane öneri kutusu tahmini yapılır. Ağın çıkışına bu öneri kutularının değerleri verilmektedir ve her bir öneri kutusu için çıkışa 85 adet değer verilir. Bu değerlerin ilk dördü (x, y, w, h) sırasıyla öneri kutusunun merkez koordinatları, genişliği ve yüksekliğidir. Ardından bu öneri kutusu için tutulan nesnellik puanı (P) gelir. Bunu COCO veri setindeki 80 tane nesne sınıfı için tahmin edilen 80 tane sınıf olasılığı (C) takip eder. Aynı şekilde 2.ve 3.öneri kutuları için de 85'er değer tutulur. Böylelikle her bir hücrenin derinliği 255 (85 x 3) olmaktadır [42].

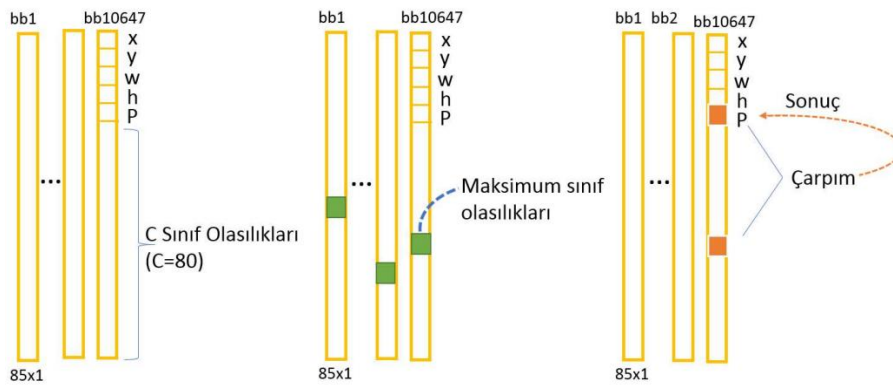




Şekil 3.18: YOLO V3 ağının çıkışındaki her bir hücrenin derinliği [42].

### 3.7.2.3 YOLO V3 Ağının Çıkışının Kullanılması

YOLO V3 ağında 3 farklı ölçekte, her bir hücre için 3 tane öneri kutusu tahmini yapılarak ağın çıkışına toplam  $3x(13x13 + 26x26 + 52x52)=10.647$  tane öneri kutusu tahmini verilir. Şekil 3.19'da gösterildiği gibi 80 tane C sınıf olasılık değerlerinden maksimum olanlar her bir öneri kutusu için ayrı ayrı belirlenir. Bulunan bu maksimum değerlerle sınır kutularının nesnellik puanı çarpılır ve sonuç nesnellik puanının yerine yazılır. Bulunan bu yeni nesnellik puanı ile öneri kutuları belli bir değer ile (0,6) eşiklenir. Eşiği aşanlar, nesnellik puanına göre büyükten küçüğe sıralanır ve maksimum olmayı bastırma (NMS: Non-Maximal Suppression) uygulanır.

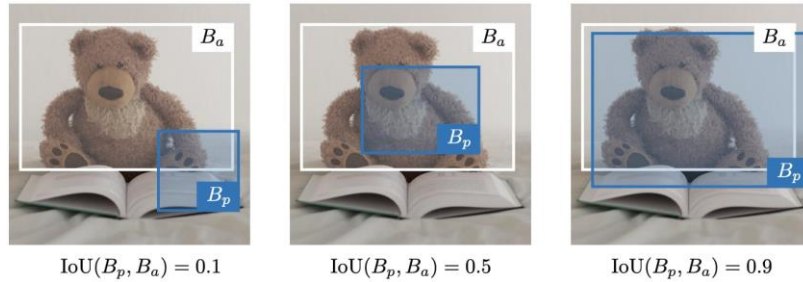


Şekil 3.19: YOLO V3 çıkış işlem adımları [42].

Maksimum olmayanı bastırma tekniği, nesne için yinelenen ve örtüşen öneri kutuları içinde nesneyi en iyi kapsayanı seçip diğerlerini elemek için uygulanır. Örneğin N tane öneri kutusu için nesnellik puanları (0,95, 0,85, 0,82, 0,75, ...) şeklinde olsun. En büyük nesnellik puanı olan kutu seçilerek diğerleri arasında karşılaştırma yapılır. Şekil 3.20’de seçilen ilk dört değer ((0,95, 0,85), (0,95, 0,82), (0,95, 0,75)) için öneri kutuları gösterilmiştir. Bu değerler arasında Şekil 3.20’de gösterildiği gibi kesişimin birleşime oranı IoU (Intersection Over Union) değeri Eşitlik (3.36) ile hesaplanır. Öngörülen sınırlama kutusu  $B_p$ ’nin gerçek sınırlama kutusu  $B_a$  üzerinde ne kadar doğru konumlandırıldığını ölçen bir fonksiyondur.  $IoU \in [0,1]$  olmak üzere:

$$IoU(B_p, B_a) = \frac{B_p \cap B_a}{B_p \cup B_a} \quad (3.36)$$

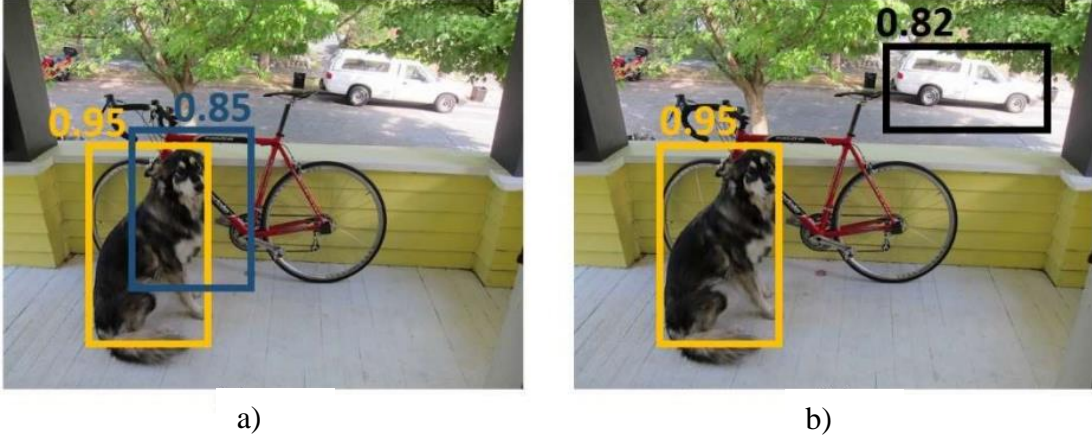
olarak tanımlanır. Bu değer öneri kutularının kesişimlerinin alanının, birleşimlerinin alanına bölünmesiyle hesaplanmaktadır.  $IoU(B_p, B_a) \geq 0.5$  olması durumunda, öngörülen sınırlama kutusu  $B_p$ , kabul edilir derecededir, nesnellik puanı düşük olan öneri kutusu  $B_p$  elenebilir.



**Şekil 3.20:** IOU değerinin hesaplanması [39].

Örneğin Şekil 3.21(a) görüntüsündeki iki öneri kutusunun IOU değeri eşik değerinden büyük çıkarsa 0,85 değerini ifade eden sınır kutusu elenir. Bu durumda bir nesne için bir öneri kutusu atanmış olur. Böylece elimizde (0,95, 0,82, 0,75, ...) değerleri kalmaktadır. Bir sonraki adımda 0,95 ile 0,85 nesnellik puanlarının belirttiği sınır kutuları arasında IoU değeri hesaplanır. Bu sınır kutuları Şekil 3.21(b)’deki görüntüde gösterilmiştir. Bunlar arasındaki IoU değeri eşik değerinden küçük çıkarsa eleme yapılmaz. Bu şekilde ilk sıradaki nesnellik puanı olan 0,95’in belirttiği öneri kutusu ile diğer nesnellik puanlarının belirttiği tüm öneri kutuları karşılaştırılır. Bir sonraki adımda ikinci sıradaki nesnellik puanına geçilir ve ondan

sonra gelen nesnellik puanları arasında karşılaştırma yapılır. Bu işlem dizinin sonuna gelene kadar tekrar edilir. Böylece kalan öneri kutuları ile sonuç elde edilir.



**Şekil 3.21:** YOLO V3 Nesnellik Puanlarının ifade ettiği sınır kutularına örnek [42].

#### 3.7.2.4 Yolo İle Nesne İçeren Görüntü Parçalarının Etiketlenmesi

Uygulamada YOLO V3 ağına girişine, mozaığı dizilmek istenen görüntü uygulanır. YOLO, algılanan nesnelere içeren çerçevelerin en, boy ve konum bilgilerinin listesini, ağına çıkışına verir. Algılanan nesnelere, nesneyi içeren en küçük dörtgen ile ifade edildiklerinde, nesne içermeyen arka plan görüntü parçaları da nesne olarak işaretlenmektedir. Bu durumdan kurtulmak için, bir kullanıcı ara yüzü açılarak, etkileşimli olarak kullanıcıdan, dikdörtgen çerçeveyi, nesnenin silüetine uygun olarak düzenlenmesi istenir.

**Tablo 3.3:** YOLO'nun algıladığı nesnelere listesi

Sıra No	Nesne (İng.)	Nesne adı	Sıra No	Nesne (İng.)	Nesne adı
1	Person	Kişi	41	Pizza	Pizza
2	Tie	Kravat	42	Donut	Çörek
3	Cell phone	Cep telefonu	43	Cake	Kek
4	Remote	Uzaktan kumanda	44	Chair	Sandalye
5	Clock	Saat	45	Potted plant	Saksı bitkisi
6	Toothbrush	Diş fırçası	46	Toaster	Tost makinası
7	Book	Kitap	47	Dining table	Yemek masası
8	Handbag	el çantası	48	Tv monitor	Tv monitor
9	Backpack	sırt çantası	49	Sofa	Kanepe
10	Suitcase	Bavul	50	Bed	Yatak
11	Umbrella	Şemsiye	51	Microwave	Mikrodalga
12	Bird	Kuş	52	Oven	Fırın
13	Cat	Kedi	53	Refrigerator	Buzdolabı
14	Dog	Köpek	54	Scissors	Makas
15	Horse	At	55	Hair drier	Saç kurutucu
16	Sheep	Koyun	56	Toilet	Tuvalet
17	Cow	İnek	57	Sink	Lavabo
18	Elephant	Fil	58	Tennis racket	Tenis raketi
19	Bear	Ayı	59	Kite	Uçurtma
20	Zebra	Zebra	60	Sports ball	Spor top
21	Giraffe	Zürafa	61	Frisbee	Frizbi
22	Teddy bear	Oyuncak ayı	62	Baseball glove	Beyzbol eldiveni
23	Mouse	Fare	63	Baseball bat	Beysbol sopası
24	Keyboard	Tuş takımı	64	Skis	Kayaklar
25	Laptop	Dizüstü	65	Snowboard	Kar tahtası
26	Vase	Vazo	66	Skateboard	Kaykay
27	Bottle	Şişe	67	Surfboard	Sörf tahtası

**Tablo 3.3** (devam)

Sıra No	Nesne (İng.)	Nesne adı	Sıra No	Nesne (İng.)	Nesne adı
28	Wine glass	Şarap kadehi	68	Bench	Bank
29	Cup	Bardak	69	Stop sign	Dur işareti
30	Fork	Çatal	70	Parking meter	Parkmetre
31	Knife	Bıçak	71	Traffic light	Trafik ışığı
32	Spoon	Kaşık	72	Fire hydrant	Yangın musluğu
33	Bowl	Kase	73	Bicycle	Bisiklet
34	Banana	Muz	74	Motorbike	Motosiklet
35	Apple	Elma	75	Car	Araba
36	Sandwich	Sandviç	76	Boat	Tekne
37	Orange	Portakal	77	Bus	Otobüs
38	Broccoli	Brokoli	78	Truck	Kamyon
39	Carrot	Havuç	79	Train	Tren
40	Hot dog	Sosisli	80	Aeroplane	Uçak

### 3.7.3 Dlib İle Yüz Anahtar Noktası İçeren Görüntü Parçalarının Etiketlenmesi

Dlib, gerçek dünya problemlerini çözmek için C++ 'da karmaşık yazılımlar geliştirilebilecek içinde makine öğrenme algoritmaları ve araçları barındıran modern bir C++ yazılım kütüphanesidir. Hem akademide hem de endüstride; robotik, gömülü cihazlar, cep telefonları ve büyük yüksek performanslı bilgi işlem ortamları da dahil olmak üzere çok çeşitli alanlarda kullanılmaktadır. Dlib'in açık kaynak lisansı, herhangi bir uygulamada ücretsiz olarak kullanılmasına olanak sağlar. ISO Standart C++ ile yazıldığından OS X, MS-Windows, Linux, Solaris, BSD ve HP-UX işletim sistemlerine taşınabilir. Python API si mevcuttur [43].

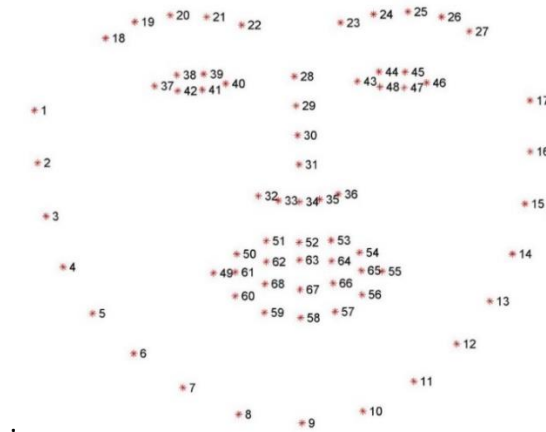
Dlib ile yüz anahtar noktalarının tespit edilmesi için kullanılan *get\_frontal\_face\_detector()* fonksiyonu, *scan\_fhog\_pyramid()* fonksiyonunu çağırarak, görüntüde kameraya bakan yüzlerin konumunu ve sayısını tespit eder. *Scan\_fhog\_pyramid()* fonksiyonu görüntünün çeşitli ölçeklenmiş sürümleri üzerinde kayan pencere dolaştırarak, gradyanların histogramı “HOG” özelliklerini çıkartır ve bu özellikleri kullanarak yüzleri tespit eder. Yüz çok değişik

açılarda görüntülenmiş olabilir ve referans noktalarını konumlamak başlı başına bir iştir ve bu amaçla tespit edilen yüz görüntüleri *dlib.shape\_predictor()* metoduna aktarılır. Bu metot çeşitli açılarda görüntülenmiş insan yüzlerini içeren “doğal ortamda 300 yüz yarışmasının” yüz anahtar noktaları veri tabanı iBUG 300-W ile eğitilmiş modelin ağırlık dosyasını kullanır. Dosya *dlib.net*’den indirilebilir. iBUG 300-W veri tabanının ticari kullanımı desteklenmemektedir [44].

Dlib yüz hizalama yöntemi V. Kazemi ve J. Sullivan ‘ın 2014’deki çalışması [45] temel alınarak yazılmıştır. Bu bildiriye, Şekil 3.22’de görülen yüzdeki referans noktaların (landmark) konumlarının, doğrudan piksel yoğunluğunun seyrek bir alt kümesinden tahmin etmek için bir regresyon ağacı dizisinin nasıl kullanılabileceği ve yüksek kaliteli tahminlere gerçek zaman performansıyla nasıl ulaşılacağı sunulmaktadır. Karesele hata kayıplarının toplamını optimize eden, eksik veya kısmen etiketli verileri işleyen, gradyan destekli öğrenen, kademeli regresyon fonksiyonları kullanan bir yöntem sunulmuştur.

Dlib yüzdeki göz, kaş, burun, çene çizgisi gibi belirgin bölgeleri konumlamak ve temsil etmek için kullanılır. Yüz çizgileri, yüz hizalama, baş poz tahmini, yüz değiştirme, göz kırpma algılama ve uyuklayan sürücü tespiti gibi birçok uygulamada başarıyla kullanılmıştır [45].

Çalışmada *dlib*, *OpenCV* kütüphanesi ve *Python* dili kullanarak uygulanmıştır. Görüntüde birden fazla insan yüzünü, sayısını, konumunu ve her bir yüzün 68 anahtar noktasını tespit edebilir. Gözbebeklerini algılamaması bir dezavantajdır.



Şekil 3.22: Dlib yüz anahtar noktaları [44].

Dlib'in girişine verilen mozaği dizilmek istenen görüntü için Dlib'in çıktısı, algılanan her bir yüz için 68 adet anahtar noktanın koordinatlarıdır.

Mozaik dizme işleminde, görüntüde insan ve nesne içeren görüntü parçalarına, arka plana göre daha uygun karolar yerleştirmek için, Tablo 3.4'de listelenen öğeler, anahtar noktaları birer doğru parçası ile birleştirilerek kalıp görüntüye çizilir. Birbirine yakın anahtar noktaları ve doğruları birleştirmek için morfolojik kapama uygulanır. Çizilen öğeler ile kesişen görüntü parçaları  $p_j=1$ , nesne içermeyen görüntü parçaları  $p_j=k$  olarak etiketlenir.

**Tablo 3.4:** Dlib yüz anahtar noktaları ve öğeleri.

Öge Numarası	Anahtar Nokta İndeksi	Açıklama
1	37-40	Sağ göz çevresi, (çokgen), en öncelikli
2	43-48	Sol göz çevresi, (çokgen)
3	18-22	Sağ kaş
4	23-27	Sol kaş
5	49-60	Dudak dış çevresi, (çokgen)
6	61-68	Dudak iç çevresi, (çokgen)
7	28-31	Burun düşey
8	32-36	Burun alt yatay
9	1-17	Yüz çevresi, en az öncelikli

### 3.7.4 OpenPose Beden Anahtar Noktalarının Bulunması

Poz tahmini, bilgisayarlı görüde kişi ya da nesnelerin öğelerinin konum ve yönelişlerinin tespit edildiği genel bir problemdir. Çözüm, nesneyi tanımlayan anahtar noktaların yerlerinin belirlenmesi ile sağlanır. Şekil 3.23'de çok sayıda insan pozu tahmini görülmektedir. Ayak anahtar noktaları (ayak parmakları ve topuklar) dahil, aynı kişiye ait beden parçaları birbirine bağlıdır [46].



**Şekil 3.23:** Çok sayıda insan pozunu tahmini [46].

İnsan görüntüsündeki anahtar noktalarının tespiti onlarca yıldır çalışılan bir problemdir. İlk çalışmalar insan yüzünü hizalamak için yüz anahtar noktalarının tespitine odaklıdır. Sonra probleme gövde, ayak, el noktaları dahil edilmiş, daha sonra ortamda bir ya da daha fazla insanın poz tahminine dönüşmüştür. Daha yeni ve zorlu bir problem, el anahtar noktalarının tespitidir. Bir sonraki adım ise, tüm bu anahtar nokta saptama görevlerinin, aynı algorithmada birleştirilerek “Tam beden (gövde, yüz, el ve ayak) poz tahmininin gerçekleştirilmesidir. Oyun ve animasyon endüstrilerinde; Hedefleme, üç boyutlu ağ (mesh) rekonstrüksiyonu gibi tam beden anahtar noktası tespitinden hemen yararlanabilecek çeşitli uygulama alanları vardır. Genel olarak, beden bilgisi kullanan her yöntem, kişinin tanınması, takibi veya eylem tanıma gibi yüz, el ve ayak tespitinden yararlanabilir. Bu ihtiyaçlara rağmen, tam beden poz tahmini sağlayan mevcut tek yöntem, OpenPose'dir [46]. OpenPose, çok ağırlı bir yaklaşım olarak, beden, yüz ve el anahtar noktası algılama algoritmalarını doğrudan kullanmaktadır. Tek bir çıkarımda birden fazla insanın tam beden kısımlarını tahmin eden tek aşamalı yöntem, sahnedeki insan sayısından bağımsız olarak, sabit bir çıkarım süresine sahiptir. İlk olarak görüntüdeki tüm bedenler tespit edilir. Beden kısımlarının hangi bedene ait olduklarının hesaplanma süresi, görüntüdeki kişi sayısı ile orantılıdır. Beden tespit edildikten sonra yüz ve el detektörleri uygulanabilir.

Yakın zamana kadar poz tahmininde pek az gelişme olmuştur. Bunun nedeni kaliteli veri kümelerinin eksikliğidir. Bu günlerde araştırmacılar, problemlerin çözümünün, sadece iyi bir veri kümesi kadar uzak olduğunu düşünmektedirler. Son yıllarda araştırmacıların problemin üzerine tüm güçleri ile gitmelerinin nedeni, cazip veri kümelerinin yayınlanmış olmasıdır. Bu veri kümelerinden bazıları [45]:

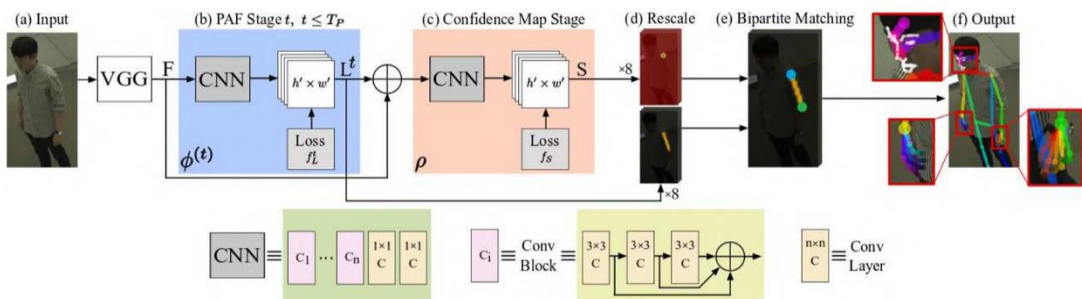


- i. COCO Anahtar Noktaları
- ii. MPII İnsan Poz Veri Kümesi
- iii. VGG Poz Veri Kümesidir.

OpenPose'nin insan poz kestirimi için COCO ve MPII ile eğitilmiş iki farklı modeli bulunmaktadır. İnsan anahtar noktaları olarak, COCO modeli 18 nokta tespit ederken, MPII modeli 15 nokta tespit etmektedir. Bu mimari, 2016 yılında COCO anahtar noktaları yarışmasını kazanmıştır.

### 3.7.4.1 Çok Sayıda İnsan Pozu Tahmin Modeli

OpenPose, Carnegie Mellon Üniversitesinin Algısal Hesaplama Laboratuvarında görüntüdeki insanların pozlarını tahmin etmek amacı ile yapılan çalışmaların bir parçasıdır. Araştırmacılar bu amaçla bir derin nöron ağı tasarlamışlar ve yukarıda adı geçen veri kümeleri ile eğitmişlerdir. Ağın genel mimarisi Şekil 3.24'de görülmektedir. Her C (Conv) bloğu bir Evrişim-PReLU dizisine karşılık gelmektedir. Şekil 3.24(a)'da modele giriş olarak  $w \times h$  boyutunda renkli bir görüntü verilmiştir. İleri beslemeli ağ, eş zamanlı olarak parçalar arası ilişkiyi (PAF) gösteren iki boyutlu vektör alanları kümesi  $L^t$ 'yi (Şekil 3.24(b)) ve beden parçalarının konumlarının iki boyutlu güven haritası  $S^t$ 'yi (Şekil 3.24(c)) öngörür. Test zamanında, doğruluğu arttırmak için en iyi PAF'lar ve güven haritaları yeniden boyutlandırılmaktadır (Şekil 3.24(d)). İkili eşleşme, aynı kişiye ait tüm beden parçalarını PAF'ları kullanarak birleştirmekte, kişileri ayrıştırmaktadır (Şekil 3.24(e)). Çıkışa, görüntüdeki her kişi için anatomik anahtar noktaların iki boyutlu konumları verilmektedir (Şekil 3.29(f)) [46].



Şekil 3.24: Çok sayıda insanın, tüm beden poz tahmini ağının mimarisi [45].

Bu yaklaşım, beden parçaları arası ilişkilendirmeleri kodlayan parça ilişki alanlarını (PAF) ve algılama güven haritalarını yinelemeli olarak öngörmektedir. Her PAF, bir anahtar noktastan diğerine işaret eden iki boyutlu bir yönlendirme vektörü olarak tanımlanmaktadır.

Şekil 3.25’de görüntünün tamamı ağa girdi olarak verilmektedir, iki dallı CNN Şekil 3.25(b)’deki parça güvenlik haritasını ve Şekil 3.25(c)’deki parça ilişki alanlarını birlikte öngörmektedir. Şekil 3.25(d)’de ayrıştırma adımında beden parça adaylarını birleştirmek için bir dizi ikili eşleşme uygulanmaktadır. Şekil 3.25(e)’de son olarak görüntüde herkesin beden pozları bir araya getirilmiştir [46].



a ) Giriş görüntüsü      c) Parça ilişki Alanları      d) İkili eşleşme      e) Ayrıştırılmış Sonuçlar

**Şekil 3.25:** OpenPose tam beden girişine verilen iki insan görüntüsünün beden parçalarının gövdeye aitliklerinin belirlenmesi [45].

$\mathbf{S}$  kümesi  $\mathbf{S} = (\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_J)$  her beden parçası için  $\mathbf{S}_j \in \mathbb{R}^{w \times h}$ ,  $j \in \{1 \dots J\}$  olan  $J$  adet güvenlik haritasına sahiptir.  $\mathbf{L} = (\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_C)$  kümesi, her bir uzuv için  $C$  adet iki boyutlu vektör alanına  $\mathbf{L}_c \in \mathbb{R}^{w \times h \times 2}$ ,  $c \in \{1, \dots, C\}$  sahiptir. Son olarak, güvenlik haritaları ve ilişki alanları, görüntüdeki insanların iki boyutlu anahtar noktalarının çıktısını vermek için açgözlü çıkarımla ayrıştırılmaktadırlar [46].

Giriş görüntüsü ilk olarak önceden eğitilmiş evrişimli nöron ağı VGG-19 ağının ilk 10 katmanı tarafından analiz edilerek özellik haritaları kümesi  $\mathbf{F}$  oluşturulmaktadır. İlk aşamada ağ tespit güvenliği haritaları kümesi  $\mathbf{S}^1 = \rho^1(\mathbf{F})$  ve parça ilişki alanı  $\mathbf{L}^{(1)} = \phi^{(1)}(\mathbf{F})$  üretilmektedir.  $\rho^1$  ve  $\phi^1$  1. aşamadaki evrişimli nöron ağlarının çıkarımlarıdır. Takip eden her aşamada, her iki dalın bir önceki aşamasından tahminler, orijinal görüntü özellikleri  $\mathbf{F}$  ile birleştirilerek, iyileştirilmiş tahmin üretmek için kullanılmaktadır [45]. Nöron ağının  $t$ . aşamasındaki ağ tespit güvenliği Eşitlik (3.37) ve parça ilişki alanı Eşitlik (3.38)’de verilmektedir:

$$\mathbf{S}^{(t)} = \rho^{(t)}(\mathbf{F}, \mathbf{S}^{(t-1)}, \mathbf{L}^{(t-1)}), \forall t \geq 2 \quad (3.37)$$

$$\mathbf{L}^{(t)} = \phi^{(t)}(\mathbf{F}, \mathbf{S}^{(t-1)}, \mathbf{L}^{(t-1)}), \forall t \geq 2 \quad (3.38)$$

$\rho^t$  ve  $\phi^t$  evrişimli nöron ağının t. aşamadaki çıkarımlarıdır.

Gerçek doğru parçaların güvenlik haritası  $S_j^*$  ve gerçek doğru parça ilişki vektör alanı  $L_c^*$  olmak üzere, Eşitlik (3.39) ve Eşitlik (3.40) da görüldüğü gibi, ağın eğitim sırasında kullanılan bazı veri kümelerinde kişiler tamamen etiketlenmediği için kestirilen öngörüler ile haritaların ve alanların gerçek doğruları ( $S_j^*, L_c^*$ ) arasına  $L_2$  normunda kayıp fonksiyonu uygulanır [46]:

$$f_S^t = \sum_{j=1}^J \sum_p \left( W(p) \|S_j^t(p) - S_j^*(p)\|_2^2 \right) \quad (3.39)$$

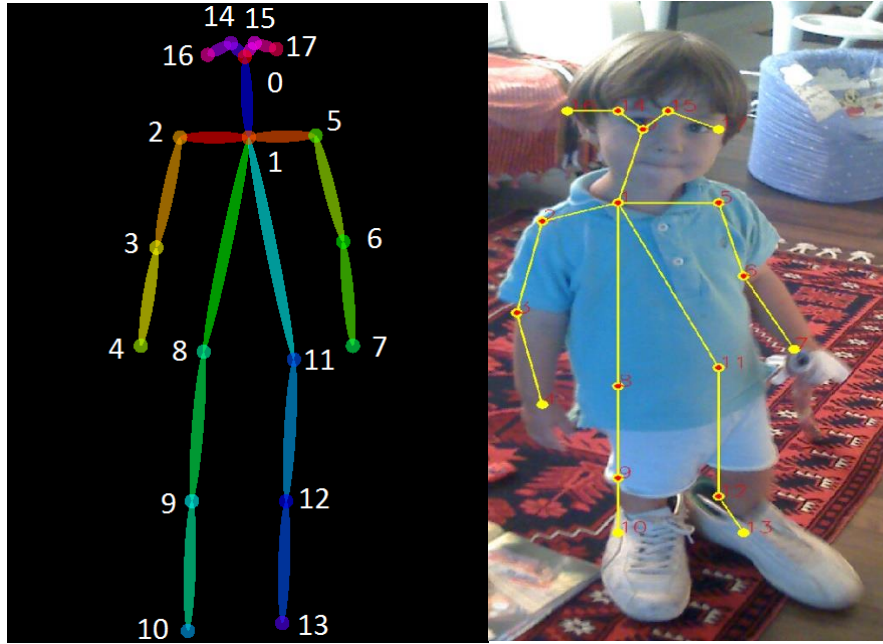
$$f_L^t = \sum_{c=1}^C \sum_p \left( W(p) \|L_f^t(p) - L_f^*(p)\|_2^2 \right) \quad (3.40)$$

$W(p)$ , veri kümesindeki bir görüntüde, bir  $p$  pikselinde etiket/açıklama eksik olduğunda  $W(p) = 0$  olan, ikili değer alan  $(0,1)$  bir maskedir. Aynı aday beden parçasının, birden fazla algılanmasını engellemek için güven haritalarında maksimum olmayan bastırma uygulanmaktadır. Son olarak, iki parçalı grafik eşleştirme, görüntüdeki her bir kişi için aynı parçayı paylaşan bağlantı adaylarını birleştirerek, tam bedeni oluşturmak için kullanılmaktadır [46].

Çalışmada, learnopencv.com sitesinde sunulan, OpenCV ve Python dili ile derin öğrenme temelli insan pozu kestirimi öğreticisinde [47] yer alan kodlar geliştirilmiştir. Evrişimli nöron ağının model dosyası için CMU Algısal Hesaplama Laboratuvarının OpenPose projesinin Github sayfası [48] ziyaret edilebilir. Eğitilmiş ağın ağırlık dosyasını, Carnegie Mellon Üniversitesi Bilgisayar Bilimleri Okulunun Algısal Hesaplama Laboratuvarının internet sitesinden indirmek için Github proje sayfasında [48] betik dosyası mevcuttur.

Tüm beden pozu tahmin ağının çıktısı, bir insan vücudu için 18 anahtar noktanın koordinatlarıdır (Şekil 3.26). Uygulamada noktaların tamamı algılanmayabilir. Görüntüde

birden fazla insan bulunması durumunda, anahtar noktalar eşleştirilerek birbirine yakınlıklarına göre hangi gövdeye ait oldukları belirlendikten sonra çıkışa verilir.



Şekil 3.26: COCO veri kümesi ile eğitilmiş OpenPose'nin 18 anahtar noktası.

#### 3.7.4.2 Beden Anahtar Noktası İçeren Görüntü Parçalarının Etiketlenmesi

OpenPose, çok sayıda insan pozü tahmin aşımın girişine, mozaiki dizilmek istenen görüntü verilir. Ağı çıktısı, görüntüdeki her bir insan vücudu için 18 anahtar noktadır. İnsanların birbirlerine çok yakın durmalarından kaynaklanan, algılanamayan noktalar olabilmektedir.

Tablo 3.5'de listelenen öğeler, anahtar noktaları birer doğru parçası çizilerek birleştirilir. Kalıp görüntüye öğeler çizildikten sonra birbirine yakın anahtar noktaları ve doğruları birleştirmek, çizgileri kalınlaştırmak için morfolojik kabartma ve kapama işlemleri uygulanır. Mozaik dizme işleminde, görüntüdeki insanlara, arka plana göre daha uygun karolar yerleştirmek için, oluşturulan çizim ile kesişen görüntü parçaları etiketlenir. Çizilen öğeler ile kesişen görüntü parçaları  $p_j=1$ , nesne içermeyen görüntü parçaları  $p_j=k$  olarak etiketlenir.

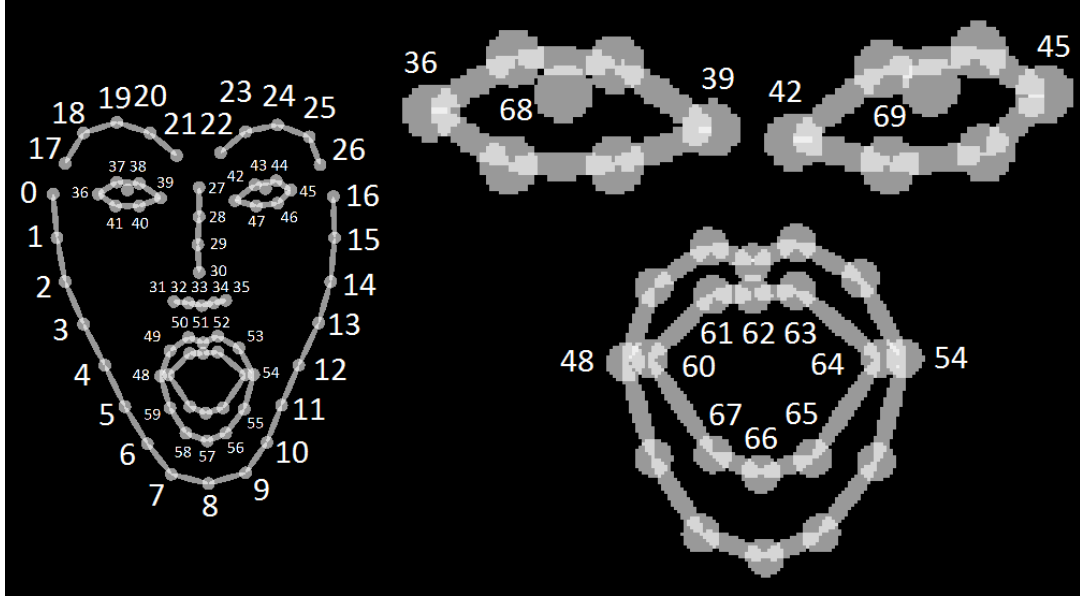
**Tablo 3.5:** İnsan bedeni anahtar noktalarının etiketlenmesi.

COCO veri kümesi insan anahtar noktaları			Beden öğeleri		
Anahtar Nokta İndeksi	Kısa Kodu İngilizce	Açıklama	Öğe Numarası	Anahtar Nokta İndeksi	Açıklama
0	Nose	Burun	1	14,15	Gözler
1	Neck	Boyun	2	16,17	Kulaklar
2	RShoulder	Sağ Omuz	3	0-1	Burun-Boyun
3	RElbow	Sağ Dirsek	4	2-3-4	Sağ Kol
4	RWrist	Sağ bilek	5	5-6-7	Sol Kol
5	LShoulder	Sol Omuz	6	8-9-10	Sağ Bacak
6	LElbow	Sol Dirsek	7	11-12-13	Sol Bacak
7	LWrist	Sol Bilek			
8	RHip	Sağ Kalça			
9	RKnee	Sağ Diz			
10	RAnkle	Sağ Ayak Bileği			
11	LHip	Sol Kalça			
12	LKnee	Sol Diz			
13	LAnkle	Sol Ayak Bileği			
14	Reye	Sağ Göz			
15	Leye	Sol Göz			
16	REar	Sağ Kulak			
17	LEar	Sol Kulak			

### 3.7.4.3 Yüz Anahtar Noktası İçeren Görüntü Parçalarının Etiketlenmesi

Resim ve heykelde olduğu gibi mozaikte de konu insan olabilmektedir. İnsan söz konusu olduğunda, yüzdeki detaylar, göz, burun, ağız bölgelerine yerleştirilecek karolar, arka plana göre çok daha dikkatli seçilmeli ve hatası daha az olmalıdır. OpenPose beden ile bulunan 18 noktadan sadece 6 tanesi (0 burun, 1 boyun, 14 sağ göz, 15 sol göz, 16 sağ kulak, 17 sol kulak) yüz ile ilgilidir. Oysa OpenPose yüz yöntemi, Şekil 3.27’de görüldüğü gibi, yüz için 70 tane anahtar nokta algılamaktadır [48]. Görüntüdeki yüzün kamera bakış açısına göre anahtar noktaların tamamı bulunamayabilir. Mozaik dizme işleminde, görüntüdeki insan

yüzüne, arka plana göre daha uygun karolar yerleştirmek amacıyla, OpenPose yüz anahtar noktalarını içeren ve Şekil 3.27'deki gibi bu noktalardan geçen doğru parçaları ile kesişen görüntü parçaları etiketlenir.



Şekil 3.27: OpenPose yüzün anahtar noktaları [47].

OpenPose yüz, dlib gibi görüntüdeki yüzlerin yerini ve sayısını tespit edememektedir. OpenPose yüzün, insan yüzünü algılayabilmesi için girişine uygulanan görüntünün sadece yüzü çerçevelemiş olması gerekmektedir. Görüntüde yüzleri tespit edip, OpenPose yüzün girişine verebilmek için, OpenPose tüm beden pozunu tahmin modeli kullanılmıştır. Görüntüde her bir beden için, sırasıyla sağ göz, sol göz, sağ kulak, sol kulak anahtar noktalarının ( 14, 15, 16, 17) koordinatlarının ortalaması alınarak, bu nokta merkezli 200x200 piksel görüntü bölgesi, OpenPose yüz evrişimli nöron ağının girişine uygulanır. Ağın çıkışından yüzün 70 adet anahtar noktasının koordinatları alınır [48].

Tablo 3.6'da listelenen öğeler, anahtar noktaları birer doğru parçası ile birleştirilerek kalıp görüntüye çizilir. Çizimden sonra birbirine yakın anahtar noktaları ve doğruları birleştirmek ve çizgileri kalınlaştırmak için morfolojik kabartma ve kapama işlemleri uygulanır. Mozaik dizme işleminde, görüntüdeki insanlara, arka plana göre daha uygun karolar yerleştirmek amacıyla, oluşturulan çizim ile kesişen görüntü parçaları etiketlenir. Çizilen öğeler ile kesişen görüntü parçaları  $p_j=1$ , nesne içermeyen görüntü parçaları  $p_j=k$  olarak etiketlenir.

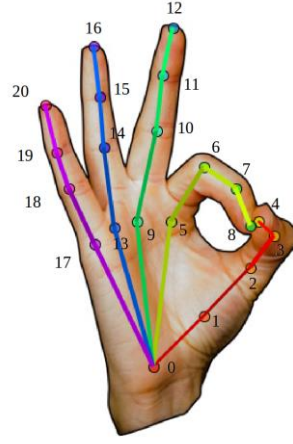
**Tablo 3.6:** Openpose yüz anahtar noktaları ve yüz öğeleri.

Öge No	Anahtar Nokta İndeksi	Açıklama
1	68	Sağ göz bebeği (en öncelikli)
2	69	Sol göz bebeği
3	36-41	Sağ göz çevresi (çokgen)
4	42-47	Sol göz çevresi (çokgen)
5	17-21	Sağ kaş
6	22-26	Sol kaş
7	48-59	Dudak dış çevresi (çokgen)
8	60-67	Dudak iç çevresi (çokgen)
9	27-30	Burun düşey
10	31-35	Burun alt yatay
11	0-16	Çene çevresi (en az öncelikli)

#### **3.7.4.4 El Anahtar Noktası İçeren Görüntü Parçalarının Etiketlenmesi**

Konu insan olduğunda, görüntüye zaman zaman insan eli de dahil olmakta ve ihmal edilemeyecek bir anlam katabilmektedir. Eldeki detaylar söz konusu olduğunda, bu bölgeye yerleştirilecek karoların, nesne içermeyen diğer bölgelere göre daha uygun karolardan seçilmesi ve yerleştirme hatalarının daha az olması sağlanabilir. OpenPose beden için bulduğu 18 noktadan sadece 2 tanesi (4 sağ bilek, 7 sol bilek) el ile ilgilidir. Yüz ve beden kestirimindeki heyecan verici gelişmelerle birlikte, son araştırmalar el anahtar noktalarının tespitine yönelmiştir.

Görüntüde el içeren bölgeleri belirlemek için OpenPose tüm beden pozu tahmin modeli kullanılır. 4 numaralı anahtar nokta sağ bilek, 7 numaralı anahtar nokta sol bilektir. 4 ve 7 numaralı anahtar noktalar merkezli 200x200 piksel görüntü bölgeleri ayrı ayrı OpenPose el evrişimli nöron ağının girişine uygulanır. Ağın çıkışından bir elin 21 adet anahtar noktasının koordinatları alınır [48].



**Şekil 3.28:** OpenPose el anahtar noktaları [48].

Tablo 3.7’de listelenen parmaklar, anahtar noktaları birer doğru parçası ile birleştirilerek kalıp görüntüye çizilir. Çizimden sonra birbirine yakın anahtar noktaları, doğruları birleştirmek ve çizgileri kalınlaştırmak için morfolojik kabartma ve kapama işlemleri uygulanır. Çizilen öğeler ile kesişen görüntü parçaları  $p_j=1$ , el içermeyen görüntü parçaları  $p_j=k$  olarak etiketlenir.

**Tablo 3.7:** OpenPose el anahtar noktaları ve öğeleri.

Öge Numarası	Anahtar Nokta İndeksi	Açıklama
1	1,2,3,4	Baş parmak
2	5,6,7,8	İşaret parmak
3	9,10,11,12,	Orta parmak
4	13,14,15,16	Yüzük parmak
5	17,18,19,20	Serçe parmak
6	0	Bilek

### 3.7.5 Nesne İçeren Görüntü Parçalarının Etiketlenmesi Yöntemi

Görüntüdeki nesnelerin en uygun yapı taşları ile dizilmesi için, evrişimli nöron ağları ile nesnelere algılayan ve nesne içeren görüntü parçalarını etiketleyen yöntemin algoritması aşağıdadır. Yöntem blok diyagram olarak Şekil 3.29’da sunulmuştur.

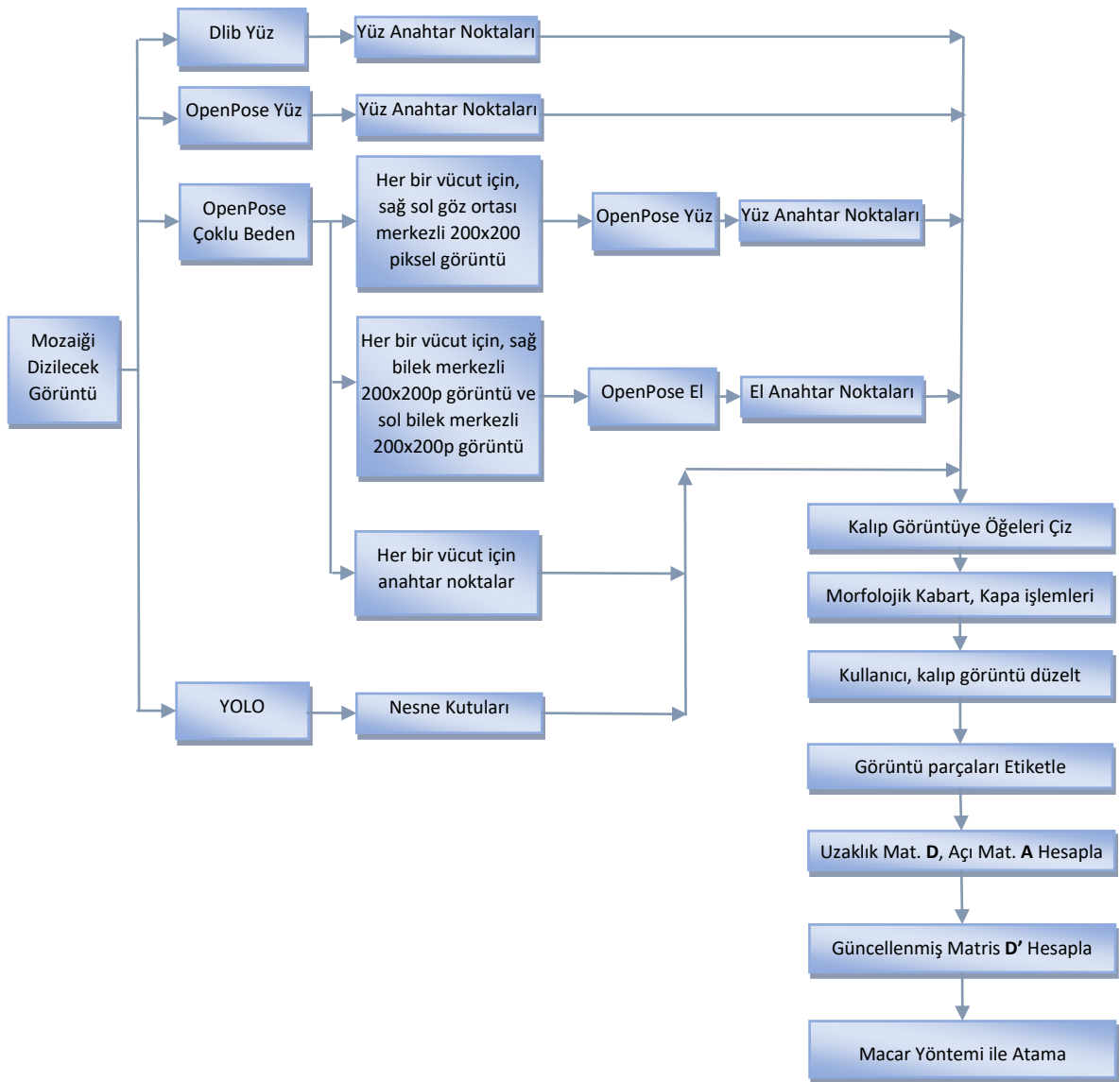
Görüntünün tamamına:



1. YOLO: Görüntünün tamamına YOLO ile nesne tespiti uygulanır. Bulunan nesnelere içeren en küçük dikdörtgenler bir listede tutulur.
2. Dlib: Görüntünün tamamına Dlib ile yüz tespiti uygulanır.
3. OpenPose Yüz: Görüntünün tamamına OpenPose Yüz ile yüz anahtar noktaları tespit edilir.
4. OpenPose Çoklu beden tespiti, görüntünün tamamına uygulanır. Bulunan her bir beden için:
  - A. Yüz bul: Bedenin sağ göz, sol göz, sağ kulak, sol kulak (14,15,16,17) noktalarının ortalaması olan nokta merkezli 200 piksel genişlik ve yükseklikte görüntü çerçeveleyerek elde edilen görüntü parçasına OpenPose yüz detektörü uygulanır. Bulunan yüz noktaları, görüntüsü ile birlikte *yüz görüntüleri* listesine eklenir.
  - B. El bul: Bedene ait 4. nokta sağ bilek veya 7. Nokta sol bilek bulunduysa bu noktalar merkezli 200piksel genişlik ve yüksekliğe sahip görüntü çerçevelenerek elde edilen görüntü parçasına OpenPose el detektörü uygulayarak parmaklara ait anahtar noktalar bulunur. Görüntü parçası ile birlikte *el görüntüleri* listesine eklenir.
5. Dlib yüz anahtar noktaları kullanarak Tablo 3.4'deki öğeleri kalıp görüntüye çiz.
6. OpenPose yüzün, görüntünün genelinde tespit ettiği yüz varsa anahtar noktalarını ve çoklu beden tespitinde **4.A.**'da elde edilen *yüz görüntüleri* listesindeki her bir yüzün anahtar noktalarını kullanarak Tablo 3.6'daki öğeleri kalıp görüntüye çiz.
7. Çoklu beden tespitinde **4.B.** de elde edilen *el görüntüleri* listesindeki her bir elin anahtar noktalarını kullanarak Tablo 3.7'deki öğeleri kalıp görüntüye çiz.
8. Çoklu beden tespiti ile elde edilen her bir gövdenin anahtar noktalarını Tablo 3.5'deki öğeleri kalıp görüntüye çiz.
9. YOLO nesne detektörünün tespit ettiği nesnelere içeren çerçevelerin alanı, eğer görüntünün alanının yarısından küçükse, nesne çerçevelerini, içi dolu dikdörtgenler olarak kalıp görüntüye çiz.
10. Kalıp görüntüde morfolojik kabartma ve kapama işlemleri uygulayarak çizgileri kalınlaştır, birbirine yakın noktaları, çizgileri birleştir ve küçük boşlukları kapat.
11. Bir arayüz açarak, kalıp görüntüyü kullanıcıya sun, ihtiyaçlar doğrultusunda kalıp görüntüye son hali vermesi sağlansın.

12. Görüntü parçaları listesini dolaşarak, konumu, kalıp görüntüde, nesne ile kesişen görüntü parçalarını  $p_j=1$ , kesişmeyen görüntü parçalarını  $p_j=k$  ile etiketle.
13. Eşitlik (3.9) ile uzaklık matrisini  $D$  hesapla
14. Eşitlik (3.35) ile güncellenmiş uzaklık Matrisi  $D'$  hesapla
15. Macar yöntemi ile atamaları gerçekleştir.

Dlib görüntünün tamamına yayılmış birden fazla yüzü bulabilirken, OpenPose yüz, portre görüntülerde başarılı olmaktadır.



**Şekil 3.29:** Nesne içeren görüntü parçalarının etiketlenmesi.

Algoritmada kullanılan morfolojik görüntü işleme yöntemlerinden kabartma, siyah-beyaz görüntülerde, nesnenin kenarlarındaki arka plan piksellerinin nesneye katılarak büyümesine neden olur.  $A$  görüntüsü bir piksel kümesi ve  $B$  bir yapılandırma elemanıdır.  $B$ 'nin orijini etrafında yansıması  $\hat{B}$ 'nin  $x$  kadar kaydırılması  $(\hat{B})_x$  dir. Kabartma Eşitlik (3.41) ile verilmiştir [49]:

$$A \oplus B = \{x \mid (\hat{B})_x \cap A \neq \emptyset\} \quad (3.41)$$

Kabartma,  $\hat{B}$ 'nin tüm  $x$  kaydırma durumları ile kesişen  $A$ 'nın boş küme olmadığı  $x$  kaydırma durumlarıdır.

Aşındırma, nesnenin kenarlarındaki nesne piksellerinin koparılarak küçülmesine neden olur. Eşitlik (3.42) ile verilmiştir [49]:

$$A \ominus B = \{x \mid (\hat{B})_x \subseteq A\} \quad (3.42)$$

$A$ 'nın,  $\hat{B}$ 'nin tüm noktalarını kapsadığı tüm  $x$  kaydırma durumlarıdır. Aşındırma ardından kabartma uygulanmasına “Açma” denir [49]. İnce çizgilerin, küçük noktaların silinmesine neden olur. Eşitlik (3.43) ile verilmiştir.

$$A \circ B = (A \ominus B) \oplus B \quad (3.43)$$

Kabartma ardından aşındırma uygulanmasına “Kapama” adı verilir [49]. Eşitlik (3.44)'de verilmiştir:

$$A \bullet B = (A \oplus B) \ominus B \quad (3.44)$$

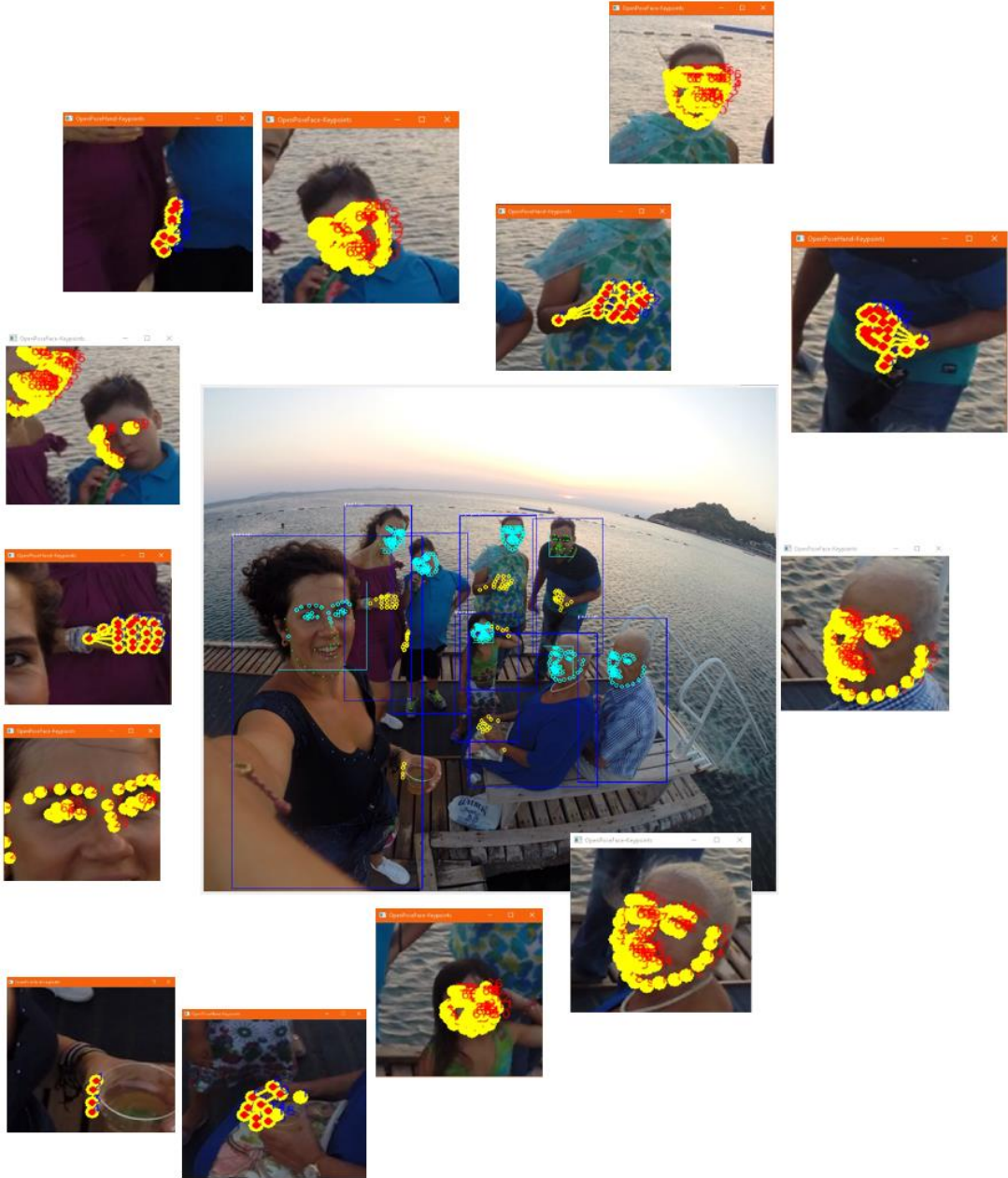
Birbirine yakın çizgilerin birleşmesine ve küçük deliklerin kapanmasına neden olur.

Algoritmanın uygulandığı görüntüde, OpenPose çoklu beden ile tespit edilen gövdeler Şekil 3.30'de yapay renklendirilmiş olarak görülmektedir.



**Şekil 3.30:** Görüntüye uygulanan OpenPose Çoklu Beden sonuçları.

Her bir bedene ait gözlerin orta noktası merkezli 200x200 piksel görüntülere uygulanan OpenPose yüz, bilek noktaları merkezli 200x200 piksel görüntülere uygulanan OpenPose el yöntemlerinin algılama sonuçları Şekil 3.31’de görülmektedir. Şekilde aynı zamanda YOLO ile algılanan kişi nesnelere mavi kutular ile işaretlenmiştir. OpenPose çoklu beden ile tespit edilen göz ve bilek noktalarına uygulanan OpenPose yüz ve el yöntemlerinin bulunduğu anahtar noktalar, kenarlardaki küçük görüntülerde sarı daireler ile işaretlenmiştir. Ortadaki görüntüye, OpenPose yüzün bulunduğu noktalar turkuaz, dlib yüzün bulunduğu noktalar yeşil, OpenPose elin bulunduğu noktalar sarı renkte aktarılmıştır.



**Şekil 3.31:** YOLO'nun algıladığı nesnelere sınırlayıcı kutuları ve OpenPose nin algıladığı anahtar noktalar.

Bulunan insanların kutularına, anahtar noktalara ve bu noktalardan geçen doğrulara morfolojik kabartma ve kapatma işlemleri uygulanarak nesnelere kalıp görüntüsü elde edilmeye çalışılır. Elde edilen görüntüye son şekli vermesi için bir arayüz açılarak kullanıcıya sunulur. Kullanıcının son şeklini verdiği kalıp görüntü Şekil 3.32(a)'da görülmektedir. Şekil 3.32 (b)'de kalıp görüntü ile kesişen görüntü parçaları görülmektedir. 8 piksel ölçülerinde 21.380 adet kardan 19.200 adet yerleştirilerek oluşturulan görüntü Şekil 3.32(c)'de görülmektedir. Karolardan 6.094 adeti insan olarak işaretlenmiştir.



a)



b)



c)

**Şekil 3.32:** a) Nesnelerin kalıp görüntüsü b) Kalıp görüntü ile kesişen 8 piksel görüntü parçaları. c) 8x8 piksel boyutlarında 19.200 adet görüntü parçasının yerleşimi.

### 3.8 Delta Robot

Robot Teknolojisi, seri kinematik zincir üzerine gelişmiştir. Birbiri ardına eklenen eksenler büyük mekanizmalara ve ağır eklemlere neden olmaktadır. Paralel Robot yapısında,

eklemler birbirine deęil, hepsi bir gövdeye mafsallanmıřtır. Böylelikle kolların hibiri, bir dięerine yük olmamaktadır [50].

Seri kinematik zincirde, eksenlerin ardıřık baęlantısı, hataların toplanarak büyümesine neden olmaktadır. Paralel robotlarda hatalar birbiri üzerine eklenmedięi için daha hassas konumlama olanaęı saęlanabilmektedir.

Paralel robotlar, seri robotlara göre daha hızlı hareket eder ve daha az enerji tüketir. Bunu en önemli nedeni hareketi aktaran mekanizmaların daha hafif olmasıdır.

Endüstride kullanılan 3 eksen Kartezyen sistemler, tasarımlarının ve kontrollerinin kolay olmasından dolayı seri doğrusal eyleyiciler yönünde gelişim göstermiştir. Doğrusal sistemlerde, adım motoru veya servo motordan gelen dönme hareketi, vidalı mil veya triger kayışı kullanılarak doğrusal harekete dönüřtürülür. Bilyalı rulman veya bilyalı araba gibi ekipmanlar, yataklama amacı ile kullanılmaktadır [50].

Redüktörler, yüke ve tahrik sistemine baęlı olarak motorlara eklenebilmektedir. Kendi aęırlıęı nedeni ile yüksek ataletlere neden olduęundan bu tasarım yapısı, makine üreticilerinin yüksek ivmeli tasarımlarının önünde bir engel oluşturmaktadır.

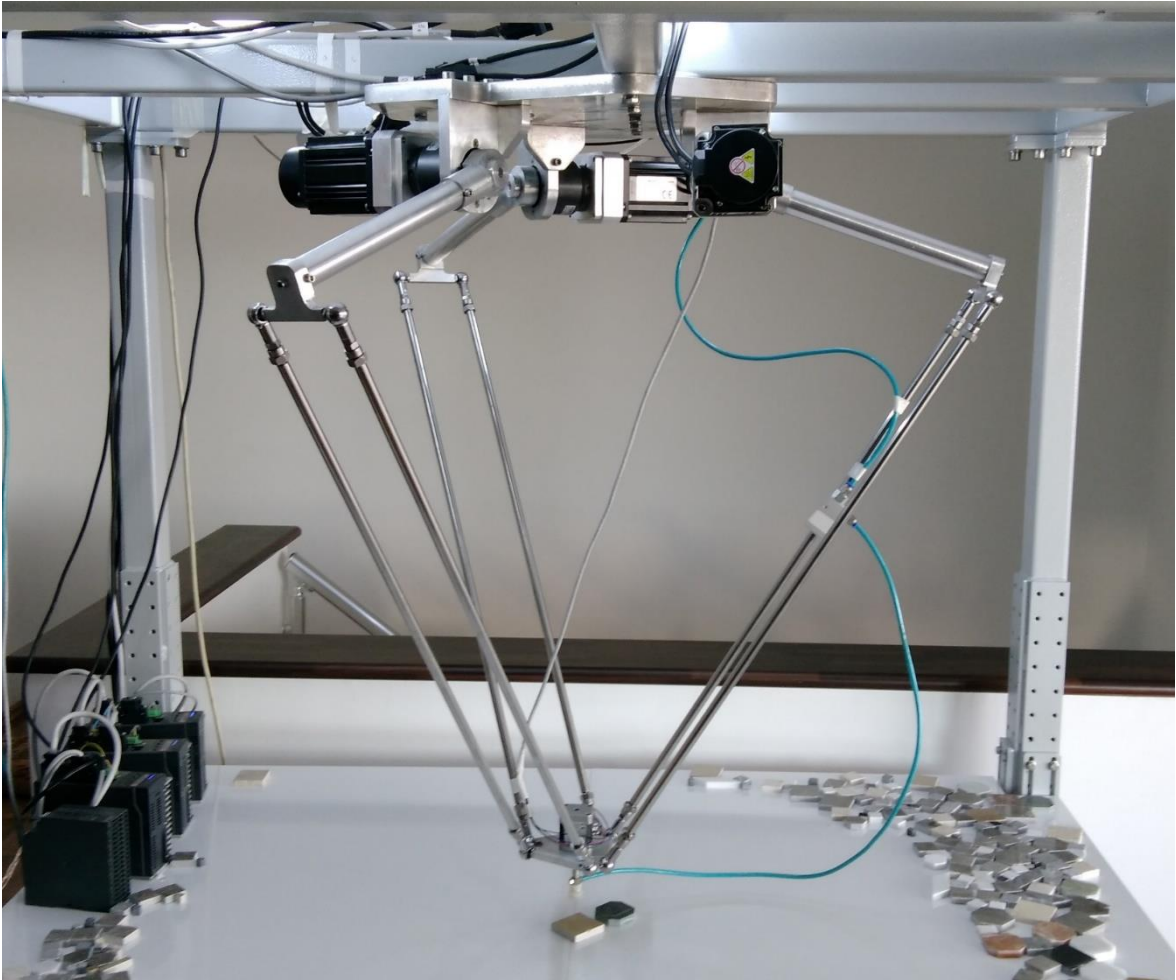
1980'lerde Prof.Dr. Reymond Clavel, Lozan şehrinde bulunan EPFL üniversitesinde; paralel eksen temelli yüksek hız ve ivmeye uygulanabilen, düşük enerji harcayan, özel üç eksen dereceli delta robot olarak bilinen bir robot icat etmiştir. Bu icadı patent ile koruma altına alınarak robotun farklı tipleri birden fazla firmaya üretim için aktarılmıştır [50].

Kanada'da yapılan bir çalışmada delta robotun ileri ve ters kinematik denklemleri, merkez koordinatı sabit olan bir daire ve merkez koordinatı hareketli olan bir kürenin kesişim noktası analitik olarak modellenerek elde edilmiş ve bir yazılım oluşturulmuştur [51].

Delta robotlar, paralel robotlar içinde en işlevsel ve uygulamaya açık bir ürün olarak kabul görse de, patent korumasından dolayı, uygulama sayısındaki artış ancak patent koruma süresinin tamamlanmasından (2010) sonra gerçekleşmiştir.

Delta robotlar üç kola sahip paralel robotlar olarak sınıflandırılmaktadır (Şekil 3.33). Endüstride yaygın olarak hızlı paketleme işlemlerini yerine getirmek için, sınıflandırma, sıralama, dizme işlemlerinde kullanılmaktadırlar [52].

Mermer parçalarının dizilmesi için dört serbestlik dereceli tut ve yerleştir robotu, delta robot için bir prototip gerçekleştirilmiştir. Bir desen oluşturan dairesel mermer mozaik madalyonlar genellikle 1m çapa sahip olmaları nedeniyle, delta robotun x-y düzleminde 1.2 m çapında çalışma alanına sahip olması hedeflenmiştir.



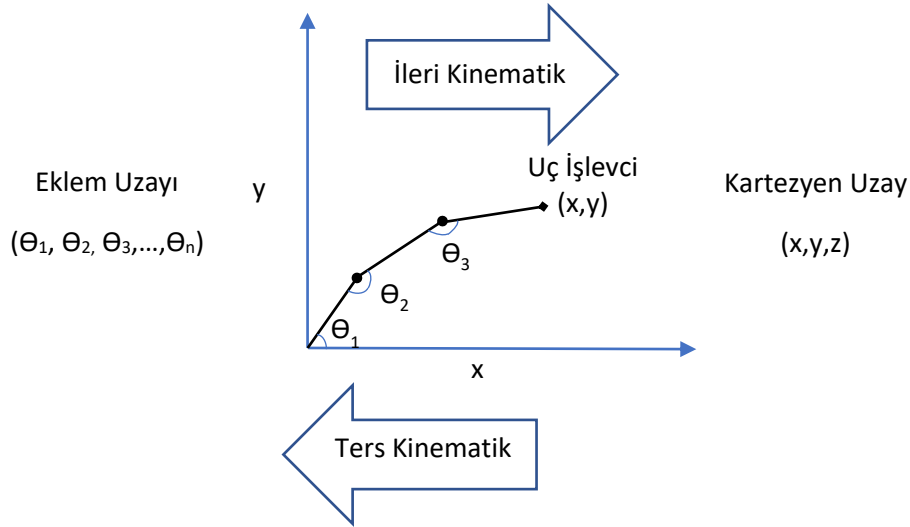
**Şekil 3.33:** Delta Robot Eksen sayısı 4, Çalışma Alanı  $\phi 1200 \times 250$  mm (silindir) yük 3 kg, dönel tork 0.5 Nm.

### 3.8.1 Delta Robot İleri ve Ters Kinematik Denklemleri

İleri kinematik problem, eklem açıları verildiğinde, Kartezyen uzayda bir robot manipülatörünün uç işlevcisinin konumunu ve yönelimini bulma işlemidir. Ters kinematik



probleme göre çok kolay olan ileri kinematik problem için her zaman bir çözüm vardır. Ters kinematik problem, Kartezyen uzayda, uç işlevcisi ana çerçeveye göre verilen konum ve yönelim verileri yardımıyla, eklem değişkenlerinin (açıların) bulunmasıdır [52]. Ters kinematik problemin çözümü, gerçek zamanlı kontrol, eyleyicilerin eklem torklarının hesaplanması, kaynak ve boyama için yörünge planlaması gibi birçok uygulama için önemlidir. Şekil3.34’de ileri ve ters kinematik problemleri şematik olarak gösterilmiştir.



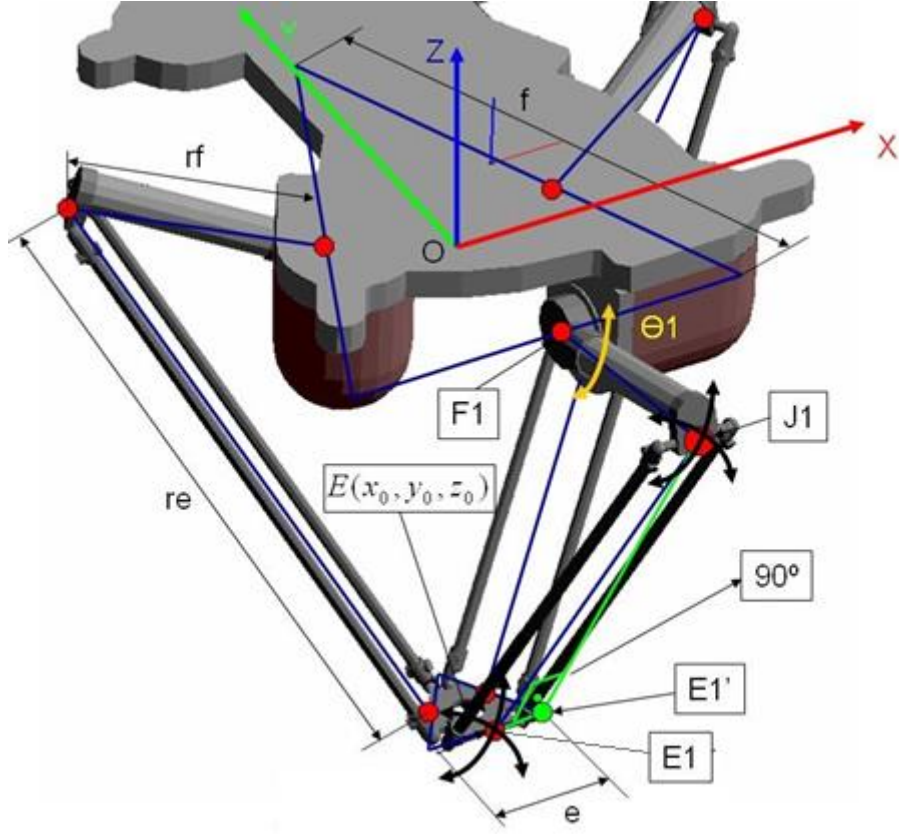
**Şekil 3.34:** İleri ve Ters Kinematik problemin şematik gösterimi.

### 3.8.2 Ters Kinematik

Delta Robotun ters kinematik denklem çözümü, ileri kinematik denklem çözümünden daha kolaydır [53]. 1. motorun üst kola bağlantısı  $F1$  ve açısı  $\theta_1$  dir. Üst kol ile alt kol arasındaki küresel eklem  $J1$  dir. Uç işlevcinin koordinatlı  $E$  dir. Uç işlevcide iki küresel ekleminin orta noktası  $E1$  dir. Kinematik modelin çıkarılmasında kullanılacak parametreler:

- Taşıyıcı görev yapan üst eşkenar üçgenin bir kenarı:  $f$
- Uç işlevciyi oluşturan alt eşkenar üçgenin bir kenarı:  $e$
- Üst kol uzunluğu, (üst kürenin yarıçapı) sabit  $F1$  noktasından küresel eklem noktası  $J1$ 'e uzaklık:  $rf$
- Alt kol uzunluğu, (alt kürenin yarıçapı) küresel eklem  $J1$ 'den  $E1$ 'e uzaklık:  $re$

Parametreler, robotun koordinat sistemine yerleşimi, Şekil 3.35 da görülmektedir.



Şekil 3.35: Delta Robot Parametreleri [53].

Taşıyıcı üçgende, 1 numaralı motor milinin üst kola bağlandığı nokta Eşitlik (3.45)'de verilmiştir:

$$F1 = (x_f, y_f, z_f) = \left( 0, \frac{-f}{2\sqrt{3}}, 0 \right) \quad (3.45)$$

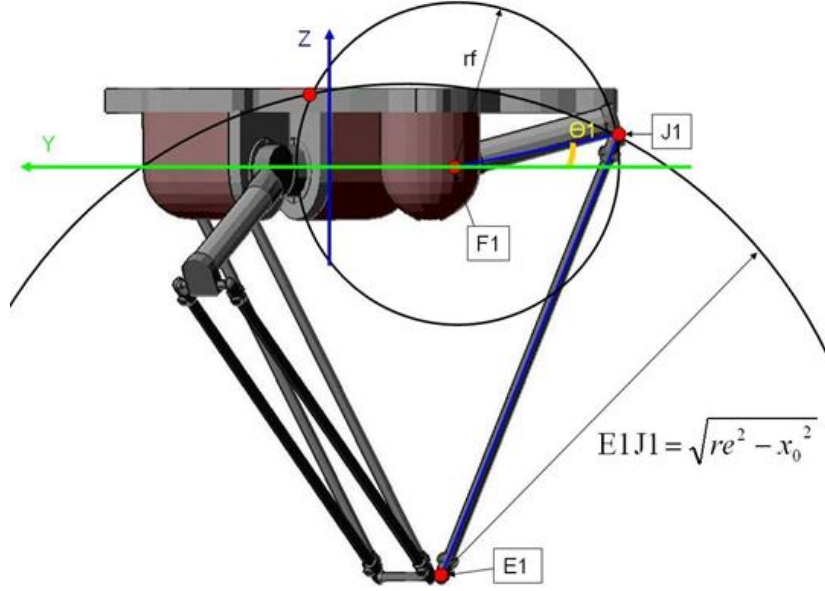
2 numaralı motor milinin üst kola bağlantı noktası Eşitlik (3.46) de verilmiştir:

$$F2 = \left( \frac{f}{4}, \frac{f}{4\sqrt{3}}, 0 \right) \quad (3.46)$$

3 numaralı motor milinin üst kola bağlantı noktası Eşitlik (3.47) de verilmiştir:

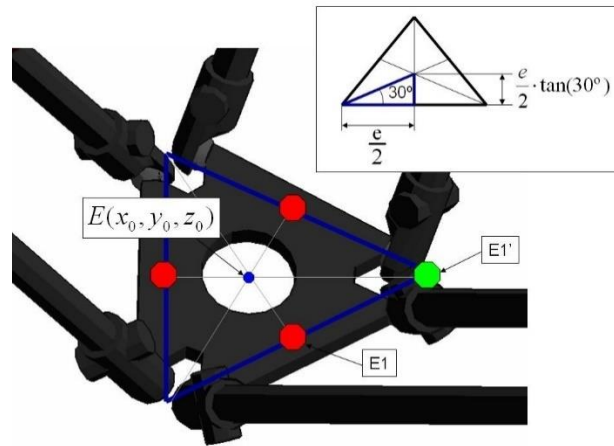
$$F3 = \left( -\frac{f}{4}, \frac{f}{4\sqrt{3}}, 0 \right) \quad (3.47)$$

Şekil 3.36’de görülen J1 noktası, üst kolun çizdiği merkezi F1, yarıçapı  $rf$  olan çember ile alt kolun çizdiği merkezi E1, yarıçapı  $\sqrt{re^2 - x_0^2}$  olan çemberin kesişim noktasıdır. Kesişim noktalarından sadece biri, Y koordinatı değeri küçük olan ters kinematik çözümde motor açısı  $\Theta$  ’nın hesaplanması için kullanılmaktadır [53].



Şekil 3.36: Üst kol ve alt kolun çizdiği çemberler ve kesişim noktaları[53].

Şekil 3.37’de Delta robot uç işlevcisi görülmektedir. Uç işlevcinin koordinatı;  $E = (x_0, y_0, z_0)$  dir. Şekil 3.38’de görülen 1 numaralı kola ait E1 noktasının koordinatları Eşitlik (3.48)’de verilmiştir:



Şekil 3.37: Uç işlevci koordinatları [53].

$$E1 = (x_e, y_e, z_e) = \left( x_0, \quad y_0 - \frac{e}{2\sqrt{3}}, \quad z_0 \right) \quad (3.48)$$

J1'in koordinatlarını, robot mekanizmasının ölçüleri ve  $E$ 'nin koordinatları kullanılarak hesaplanabilir [53]. Üst kolun çizdiği çemberin denklemi Eşitlik (3.49)'de verilmiştir:

$$(y - y_f)^2 + z^2 = rf^2 \quad (3.49)$$

Alt kolun çizdiği çember Eşitlik (3.50)'de verilmiştir.

$$(y - y_e)^2 + (z - z_0)^2 = re^2 - x_0^2 \quad (3.50)$$

$y_e = y_0 - \frac{e}{2\sqrt{3}}$  dır. Eşitlik (3.49)'dan, Eşitlik (3.50) çıkartılarak Eşitlik (3.51) elde edilir [53].

$$2(y_e - y_f)y + 2 \cdot z \cdot z_0 = rf^2 - re^2 + x_0^2 + z_0^2 - y_f^2 + y_e^2 \quad (3.51)$$

Buradan  $z$  çekilerek Eşitlik (3.52) elde edilir [53]:

$$z = \underbrace{\frac{rf^2 - re^2 + x_0^2 + z_0^2 + y_e^2 - y_f^2}{2 \cdot z_0}}_a + \underbrace{\frac{(y_f - y_e)}{z_0}}_b \cdot y \quad (3.52)$$

$$a = \frac{rf^2 - re^2 + x_0^2 + z_0^2 + y_e^2 - y_f^2}{2 \cdot z_0} \quad (3.53)$$

$$b = \frac{(y_f - y_e)}{z_0} \quad (3.54)$$

Eşitlik (3.52)'de sabit değer yerine Eşitlik (3.53) ile verilen  $a$  ve  $y$ 'nin katsayısı yerine Eşitlik (3.54) ile verilen  $b$  konulduğunda Eşitlik (3.55) elde edilir [53].

$$z = a + b \cdot y \quad (3.55)$$

Eşitlik (3.55), Eşitlik (3.49)'da  $z$  yerine konularak Eşitlik (3.56) elde edilir:

$$(1 + b^2) \cdot y^2 + 2(ab - y_f) \cdot y + (a^2 + y_f^2 - rf^2) = 0 \quad (3.56)$$

İkinci dereceden Eşitlik (3.56) denkleminin kökleri: Eşitlik (3.57)'de verilmiştir.

$$y_{J1} = \frac{y_f - ab \pm \sqrt{-(a + b \cdot y_f)^2 + rf^2(1 + b^2)}}{(1 + b^2)} \quad (3.57)$$

Eşitlik (3.57) de, uzaktaki kök (karekökün eksi işaretlisi) kullanılarak  $y$  bulunur, Eşitlik (3.55) de yerine konularak  $z$  hesaplanır [53].

$J1 = (x_j, y_j, z_j)$  noktası bulunduğunda, 1. motorun açısı  $\Theta_1$  Eşitlik (3.58) de verilmiştir:

$$\theta_1 = \frac{180^\circ}{\pi} \tan^{-1}\left(\frac{z_j - z_f}{y_j - y_f}\right) \quad (3.58)$$

Eşitlik (3.58)'de  $F1$  noktası için  $z_f = 0$  ve  $y_f = \frac{-f}{2\sqrt{3}}$  dır.

$\Theta_2$  ve  $\Theta_3$  açıları için  $E = (x_0, y_0, z_0)$  noktası,  $z$  eksenine göre  $\pm 120$  derece döndürülerek, yine Eşitlik (3.58) ile açılar hesaplanır [53]. 2. motor için  $E_2$  noktası Eşitlik (3.59)'da verilmiştir:

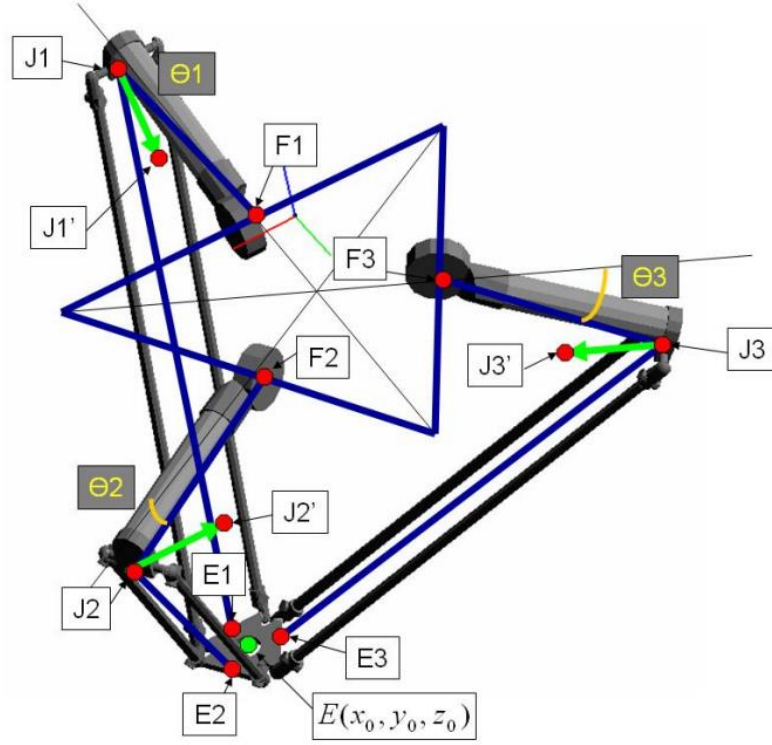
$$E_2 = (x_0 \cos(120) + y_0 \sin(120), \quad y_0 \cos(120) - x_0 \sin(120), \quad z_0) \quad (3.59)$$

3. motor için  $E_3$  noktası Eşitlik (3.60)'da verilmiştir:

$$E_3 = (x_0 \cos(120) - y_0 \sin(120), \quad y_0 \cos(120) + x_0 \sin(120), \quad z_0) \quad (3.60)$$

### 3.8.3 İleri Kinematik

Şekil (3.39)'da  $F1, F2, F3$  noktalarından üst kollara bağlı motorlar, üst kolları  $\Theta_1, \Theta_2, \Theta_3$  açılarına getirildiğinde, üst kollar ile alt kolların bağlantı noktaları  $J1, J2, J3$ , aynı zamanda alt kolların çizdiği yarıçapı  $re$  olan çemberlerin merkez noktalarıdır. İleri kinematik problemin çözümü, alt kolların çizdiği yarıçapı  $re$  olan çemberlerin kesişim noktasının bulunmasıyla sağlanır. Alt kolların uçları bir noktada birleşmeyip, uç işlevci üçgenin kenarlarında,  $E1, E2, E3$  noktalarında yer aldığı için,  $E1E, E2E, E3E$  geçiş vektörleri kullanılarak kürelerin merkezleri  $J1, J2, J3$  noktalarından  $J1', J2', J3'$  noktalarına taşınır (Şekil 3.38 ve Şekil 3.39).  $J1', J2', J3'$  noktaları, sırasıyla Eşitlik (3.61), (3.62) ve (3.63)'de verilmiştir [53]:



Şekil 3.38: Bağlantı noktaları [53].

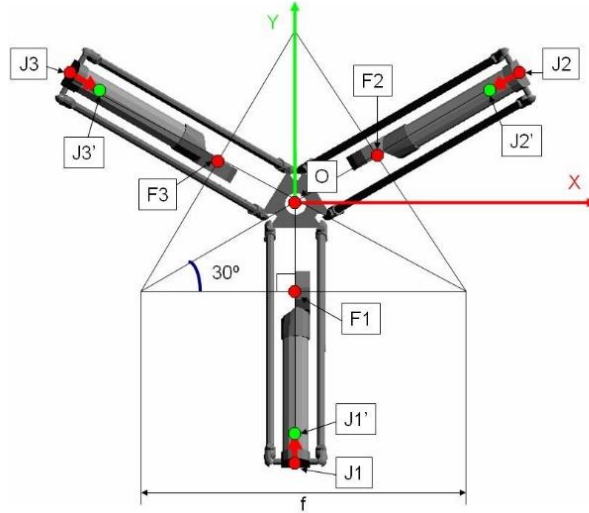
$$J1' = (x_1, y_1, z_1) = \left( 0, -\frac{(f-e)}{2\sqrt{3}} - rf \cdot \cos(\theta_1), -rf \sin(\theta_1) \right) \quad (3.61)$$

$$J2' = (x_2, y_2, z_2) = \quad (3.62)$$

$$\left( \left[ \frac{(f-e)}{2\sqrt{3}} + r_f \cdot \cos(\theta_2) \right] \cos(30), \left[ \frac{(f-e)}{2\sqrt{3}} + rf \cdot \cos(\theta_2) \right] \sin(30), -rf \sin(\theta_2) \right)$$

$$J3' = (x_3, y_3, z_3) = \quad (3.63)$$

$$\left( -\left[ \frac{(f-e)}{2\sqrt{3}} + r_f \cdot \cos(\theta_3) \right] \cos(30), \left[ \frac{(f-e)}{2\sqrt{3}} + r \cdot \cos(\theta_3) \right] \sin(30), -rf \sin(\theta_3) \right)$$



Şekil 3.39: J1', J2', J3' noktalarının x-y düzlemine izdüşümü [53].

1, 2 ve 3 numaralı motorların alt kollarının çizdiği, yarıçapı  $re$  olan çemberlerin denklemleri Eşitlik (3.64), (3.65) ve (3.66) ile verilmiştir [53]:

$$(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = re^2 \quad (3.64)$$

$$(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = re^2 \quad (3.65)$$

$$(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = re^2 \quad (3.66)$$

Şekil 3.39'da 1 numaralı motorda  $x_1=0$  olduğu görülmektedir. Eşitlik (3.64), (3.65), (3.66)'da parantezlerin karesi alınarak Eşitlik (3.67), (3.68), (3.69) elde edilir:

$$x^2 + y^2 + z^2 - 2y_1y - 2z_1z = re^2 - y_1^2 - z_1^2 \quad (3.67)$$

$$x^2 + y^2 + z^2 - 2x_2x - 2y_2y - 2z_2z = re^2 - x_2^2 - y_2^2 - z_2^2 \quad (3.68)$$

$$x^2 + y^2 + z^2 - 2x_3x - 2y_3y - 2z_3z = re^2 - x_3^2 - y_3^2 - z_3^2 \quad (3.69)$$

Eşitlik (3.67), (3.68), (3.69)'u sadeleştirmek için Eşitlik (3.70) ile verilen  $w$ , denklemlerde yerine konur.

$$w_i = x_i^2 + y_i^2 + z_i^2 \quad (3.70)$$

Eşitlik (3.67)'den Eşitlik (3.68) çıkartılarak Eşitlik (3.71) elde edilir:

$$x_2x + (y_1 - y_2)y + (z_1 - z_2)z = \frac{w_1 - w_2}{2} \quad (3.71)$$

Eşitlik (3.67)'den Eşitlik (3.69) çıkartılarak Eşitlik (3.72) elde edilir:

$$x_3x + (y_1 - y_3)y + (z_1 - z_3)z = \frac{w_1 - w_3}{2} \quad (3.72)$$

Eşitlik (3.68)'den Eşitlik (3.69) çıkartılarak Eşitlik (3.73) elde edilir:

$$(x_2 - x_3)x + (y_2 - y_3)y + (z_2 - z_3)z = \frac{w_2 - w_3}{2} \quad (3.73)$$

Eşitlik (3.71)'den  $y$ 'yi çeker ve Eşitlik (3.72)'de yerine koyarsak, Eşitlik (3.74) elde edilir [53]:

$$x = \frac{[(y_1 - y_2)(z_1 - z_3) - (y_1 - y_3)(z_1 - z_2)]z + \frac{1}{2}[(w_1 - w_2)(y_1 - y_3) - (w_1 - w_3)(y_1 - y_2)]}{(y_3 - y_1)x_2 - (y_2 - y_1)x_3} \quad (3.74)$$

Eşitlik (3.74)'de  $z$ 'nin katsayısı yerine Eşitlik (3.76) ile verilen  $a_1$ , sabit değer yerine Eşitlik (3.77) ile verilen  $b_1$  konursa, Eşitlik (3.75) elde edilir. Eşitlik (3.78)'de  $d$ ,  $a_1$  ve  $b_1$ 'in paydalarında yer almaktadır.

$$x = a_1z + b_1 \quad (3.75)$$

$$a_1 = \frac{1}{d}[(y_1 - y_2)(z_1 - z_3) - (y_1 - y_3)(z_1 - z_2)] \quad (3.76)$$

$$b_1 = \frac{1}{2d}[(w_1 - w_2)(y_1 - y_3) - (w_1 - w_3)(y_1 - y_2)] \quad (3.77)$$

$$d = (y_1 - y_2)x_3 - (y_1 - y_3)x_2 \quad (3.78)$$

Eşitlik (3.72)'den  $x$  çekilir ve Eşitlik (3.71)'de yerine konulursa, Eşitlik (3.79) elde edilir [53]:

$$y = \frac{[(z_1 - z_3)x_2 - (z_1 - z_2)x_3]z + \frac{1}{2}[(w_1 - w_2)x_3 - (w_1 - w_3)x_2]}{(y_1 - y_2)x_3 - (y_1 - y_3)x_2} \quad (3.79)$$

Eşitlik (3.79)'da  $z$ 'nin katsayısı yerine Eşitlik (3.81) ile verilen  $a_2$ , sabit değer yerine Eşitlik (3.82) ile verilen  $b_2$  konulursa, Eşitlik (3.80) elde edilir. Eşitlik (3.83)'da  $d$ ,  $a_2$  ve  $b_2$ 'nin paydalarında yer almaktadır.



$$y = a_2 z + b_2 \quad (3.80)$$

$$a_2 = \frac{1}{d} [(z_1 - z_3)x_2 - (z_1 - z_2)x_3] \quad (3.81)$$

$$b_2 = \frac{1}{2d} [(w_1 - w_2)x_3 - (w_1 - w_3)x_2] \quad (3.82)$$

$$d = (y_1 - y_2)x_3 - (y_1 - y_3)x_2 \quad (3.83)$$

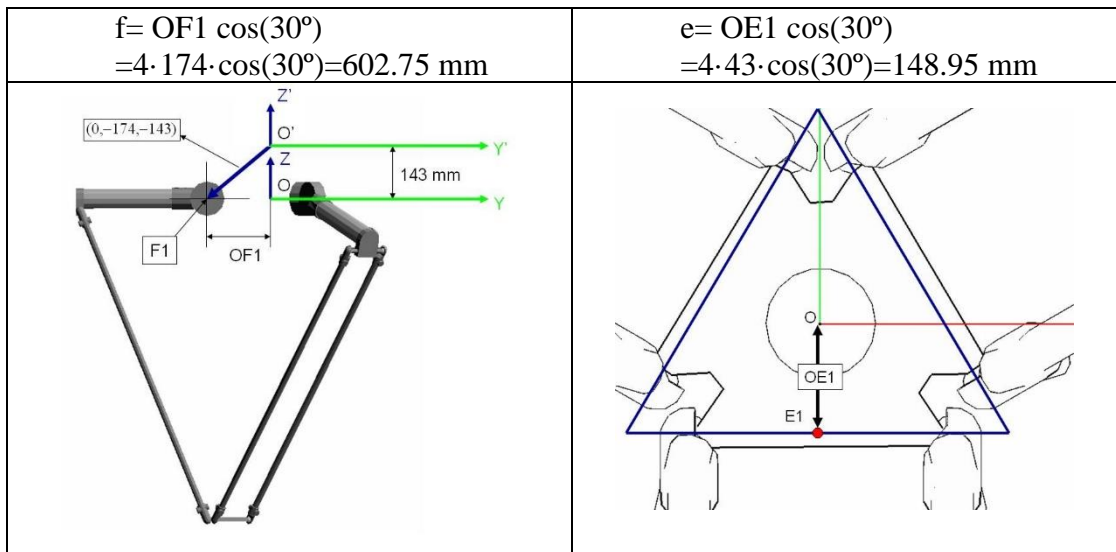
Eşitlik (3.75) ve Eşitlik (3.80), Eşitlik (3.67)'de yerine konulduğunda Eşitlik (3.84) elde edilir:

$$\underbrace{(a_1^2 + a_2^2 + 1)}_a z^2 + \underbrace{2(a_1 b_1 + a_2(b_2 - y_1) - z_1)}_b z + \underbrace{b_1^2 + (b_2 - y_1)^2 + z_1^2 - r e^2}_c \quad (3.84)$$

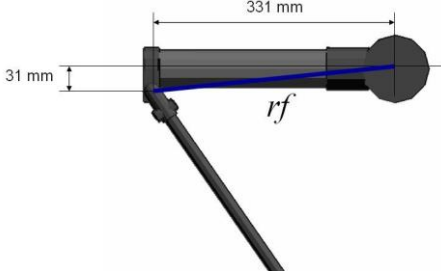

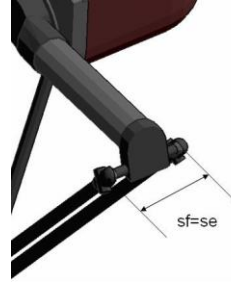
İkinci dereceden Eşitlik (3.84) denkleminin köklerinden, en küçük negatif kök  $z=z_0$ 'ı verir. Sırasıyla  $x_0$  ve  $y_0$ , Eşitlik (3.75) ve Eşitlik (3.80) denklemlerinde  $z$  yerine konularak bulunur [53].

### 3.8.4 Ölçülerin Belirlenmesi

Boyutlandırma için, çalışma alanı ölçülerine uygun, sektörde kullanılan çeşitli marka ve modellere ait teknik özellikler incelenmiştir. Belirlenen ölçüler (Şekil 3.40), ileri ve ters kinematik denklemlerde kullanılarak istenen konumlamayı sağlayıp sağlamadığı ilk olarak bilgisayar ortamında denenmiştir.



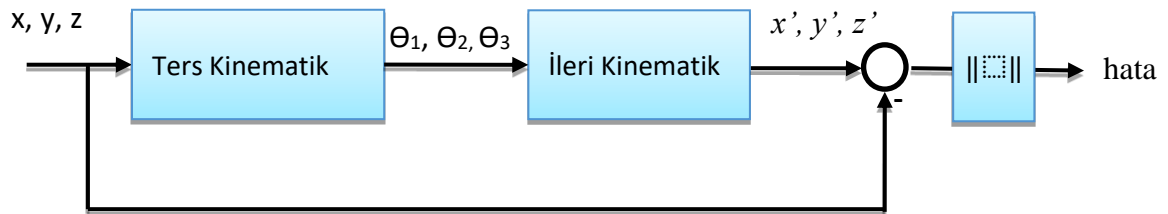
Şekil 3.40: Delta robot prototipinin ölçüleri.

$rf = \sqrt{331^2 + 31^2} = 332.44mm$	$re=870 mm$	$sf=se=50 mm$
		

Şekil 3.40: (devam)

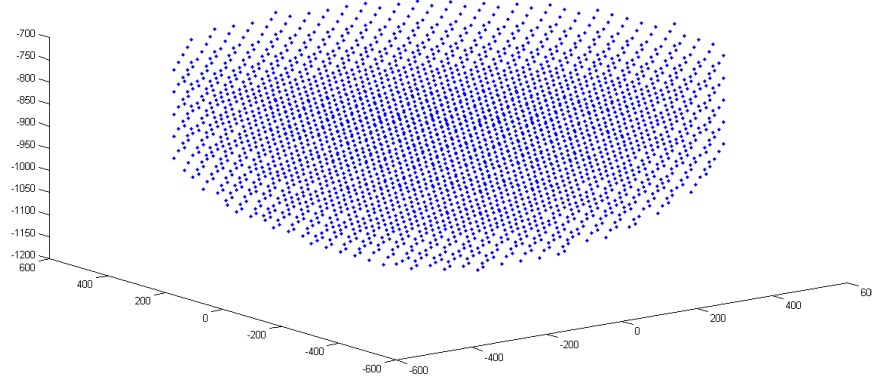
### 3.8.5 Ölçülerin Sağladığı Çalışma Bölgesinin Çıkarılması

Belirlenen Delta robot ölçülerinin sağladığı çalışma bölgesinin incelenmesi amacı ile,  $-1230 < z < -780 mm$ ,  $-400 < x, y < 400 mm$  çalışma bölgesinde,  $x, y$  düzlemindeki yarıçap  $r = \sqrt{x^2 + y^2} < 600 mm$  olacak şekilde, 50 mm düzenli aralıklarla  $(x, y, z)$  örnek konum verileri oluşturulmuştur. Örnek konum koordinatları, ters kinematik denkleme konarak  $\Theta_1$ ,  $\Theta_2$  ve  $\Theta_3$  eklem açıları elde edilmiştir. Elde edilen eklem açıları, ileri kinematik denkleme girilerek hesaplanan konum  $(x', y', z')$  verileri elde edilmiştir. Örnek konum verileri  $(x, y, z)$  ile kinematik denklemlerden hesaplanan konum  $(x', y', z')$  arasındaki Öklid uzaklık hesaplanarak elde edilen hata (Şekil 3.41), denklemlerin doğruluğunu sınamak için kullanılmıştır.



Şekil 3.41: Ters ve ileri kinematik denklemler arasındaki hata

Hata merkezden uzaklaştıkça, kenarlara doğru artmaktadır. Hatanın en büyük değeri:  $1.147 \cdot 10^{-3} mm$  dir (Şekil 3.42).  $z = -1030 mm$  den başlayarak, aşağı inildikçe uç işlevcinin erişebildiği bölgenin yarıçapı küçülmektedir. Seçilen delta robot ölçülerinin teorik olarak istenen çalışma bölgesini sağladığı görülmüştür.



**Şekil 3.42:** Çalışma bölgesi, 50mm aralıklarla örneklendirilmiştir.

### 3.8.6 Hareket Kontrol

Delta robot uç işlevcisinin konumlandırılması,  $\Theta_1$ ,  $\Theta_2$ ,  $\Theta_3$  üst kol açılarının kontrolü ile gerçekleştirilmektedir. Mermer mozaik parçalarının istenen hız ve doğrulukta dizilebilmesi için uç işlevciyi döndürmek ve hareket ettirmek üzere uygun motorlar, motor sürücüler, kontrol donanımı belirlenmiş, temin edilmiş ve geliştirilen yazılımlarla birlikte hareket kontrolü gerçekleştirilmiştir.

### 3.8.7 AC-Servo Motor, Sürücü ve Redüktör Belirlenmesi:

Bir mermer koroyu konumlandırırken, delta robotun ne kadar yük taşıyacağını belirlemek amacı ile, bir mozaik üreticisinden alınan, çeşitli mermer karo örnekleri incelenmiştir. Ölçüleri ve kütleleri Tablo 3.8 verilen mermer parçalarının özkütleleri  $2.74-2.87 \text{ gr/cm}^3$  aralığında ölçülmüştür.

**Tablo 3.8:** Bir mozaik üreticisinden alınan çeşitli örneklerin ölçü ve ağırlıkları.

Ağırlık(gr)	Ölçüler(mm)	Açıklama
2.8	10x10x10	Küp şeklinde gri
3.4	10x10x10	Küp şeklinde siyah
6	14x14x10	Kare prizma
11	20x20x10	Kare prizma
14.6	23x23x10	Kare prizma
26.2	30x30x10	Kare prizma
37	57x32, h=10	Paralelkenar
40	50x50, h=10	Dik üçgen
55	28, h=10	Altıgen
57.4	48x48x10	Köşeleri pahlı kare
60.5	47x47x10	Kare prizma

Delta robotun motorlarına yük olarak binen hareketli parçalarının ağırlıkları Tablo 3.8'de listelenmiştir.

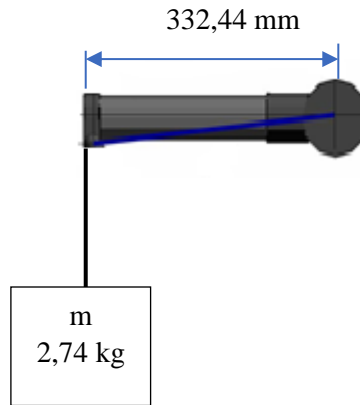
**Tablo 3.9:** Delta robot bileşenlerinin ağırlıkları.

Bileşen	Ağırlığı
Üst kol mili	734 gr
Alt kol mili	264 gr
Alt üçgen	768 gr
Step motor (Nema17)	350 gr
Mermer parça	100 gr

Bir motorun miline takılı bir üst kol ve ucuna asılı üç alt kol, alt üçgen, bir adım motoru ve bir mermer parça ağırlıklarını tahmini yük olarak kabul edersek (Şekil 3.43):

Yük = 734+3·264+768+350+100=2744 gr dır. Üst kol yatay durumdayken, motorun döndürme momenti Eşitlik (3.81) da hesaplanmıştır.

$$M_d = m \cdot g \cdot x = 2,744 \cdot 9,81 \cdot 0,33244 = 8,948 \text{ Nm} \quad (3.85)$$



**Şekil 3.43:** Kuvvet kolu, üst kol.

Kullanılması gereken motorun çalışma torku tahminen 9 Nm olmalıdır. Motor gücü arttıkça maliyeti de artmaktadır. Düşük güçte bir motorun miline redüktör takıldığında tork artırılarak istenen tork sağlanabilir. Redüktör aynı zamanda açı ile konumlama hassasiyetini

de arttırmaktadır. Redüktör kullanmanın tek dezavantajı bir miktar boşluğa sahip olmasıdır. 3 Adet NEUGART PLPE 70,  $i=20$  redüktör özellikleri; Maksimum çıkış torku 53 Nm, nominal çıkış torku 33 Nm, maksimum giriş hızı 4500 dev/dak ve boşluğu  $0,2^\circ$  (12 arcmin) dir. Çevrim Oranı  $i=20$  olduğundan, redüktör çıkışında, giriş hızını 20 kat düşürürken ( $3000/20=150$  dev/dak), 2,4 Nm motor torkunu 20 kat arttırarak 48 Nm lik döndürme momentini üst kollara uygulayabilir, ancak nominal çıkış torku olarak katalogunda 33 Nm önerilmektedir [54].

Üç kol için, 3 adet ADTECH marka, ACH-09075DC model, 750 W, 2,4 Nm, 3A, 3000 rpm AC servo Motor. 3 adet QS7AA030M AC Servo Motor Sürücü kullanılmıştır. Dönme eksenini  $\Theta$  için SY42STH47-1684A Nema17 0,5 Nm adım motoru ve adım motoru sürücü olarak ZM-2H042MD-1.70 A(0,5 Nm) 12-24 Vdc kullanılmıştır.

### **3.8.8 PLC Seçimi:**

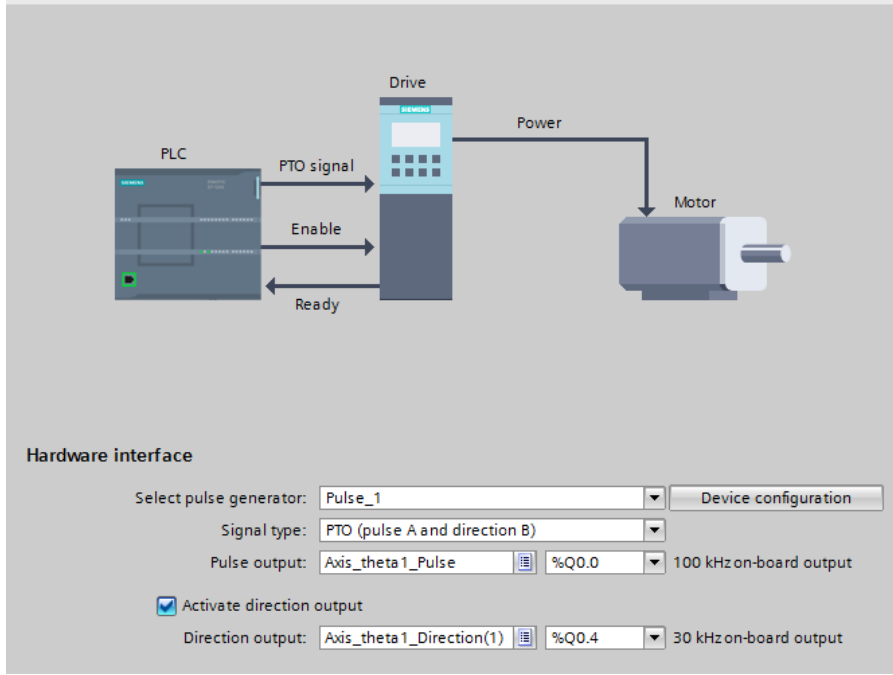
Robotun hareket kontrolünün gerçekleştirilmesi için geniş ve detaylı dokümantasyona sahip, Siemens marka PLC tercih edilmiştir. PLC'nin programlanması için görsel arayüz ve kapsamlı fonksiyon kütüphanesine sahip TIA Portal otomasyon yazılımı büyük kolaylık sağlamaktadır. Siemens S7-1200 (CPU1214 DC/DC/DC) PLC, iki motorun kontrolü için tavsiye edilen dört adet yüksek hızlı darbe katarı çıkışı (PTO 100 kHz) içermektedir. Motor dönüş yönü kontrolü için 30 kHz'lik çıkışlar da kullanılarak bir adet PLC ile, üçü servo, biri adım olmak üzere dört adet motor kontrol edilmiştir.

### **3.8.9 Motor Sürücü ve PLC Arasında Haberleşme**

Bilgisayar (PC) ile haberleşerek, bilgisayardan gelen veriler ile motor sürücülerinin kontrol edilebilmesi için PLC programlanmıştır. PLC de dört adet eksen tanımlanmıştır. Eksenlerin konumlandırma ölçüsü  $^\circ$ (derece) olarak ayarlanmıştır. Sinyal tipi olarak darbe/yön seçilmiştir. Eksenlerden birincisi( $\Theta_1$ ) Şekil 3.44'da görülmektedir. Kare dalga çıkışı (PTO) Q0.0 en fazla 100 kHz ve yön çıkışı Q0.4 30 kHz'lik kare dalga sinyalleridir.

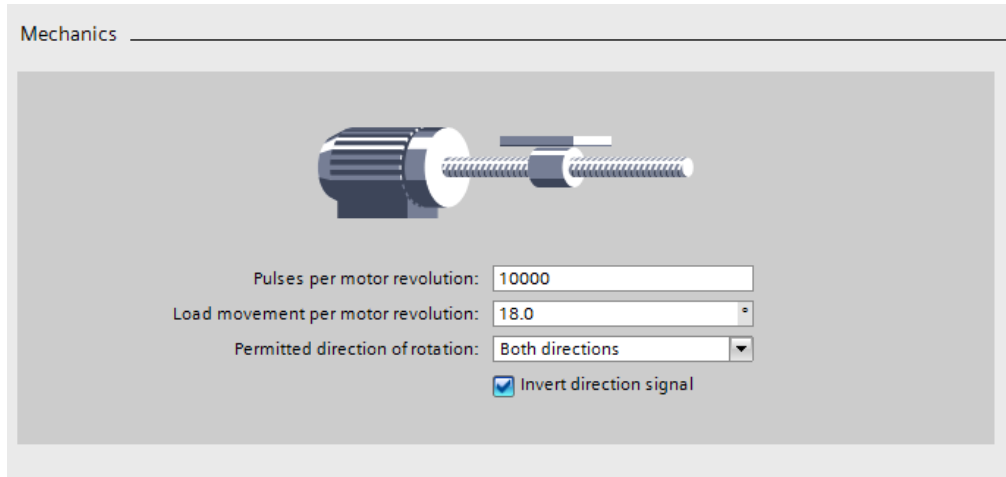
Pulse Output: Darbe Çıkışı, motor hızını kare dalga frekansı (maksimum 100 kHz) ile kontrol eden sinyaldir.

Direction Output: Yön Çıkışı, 0 için saat yönünde, 1 için saat yönünün tersi yönde motor dönüş yönünü kontrol eden 30 kHz'lik sinyaldir.



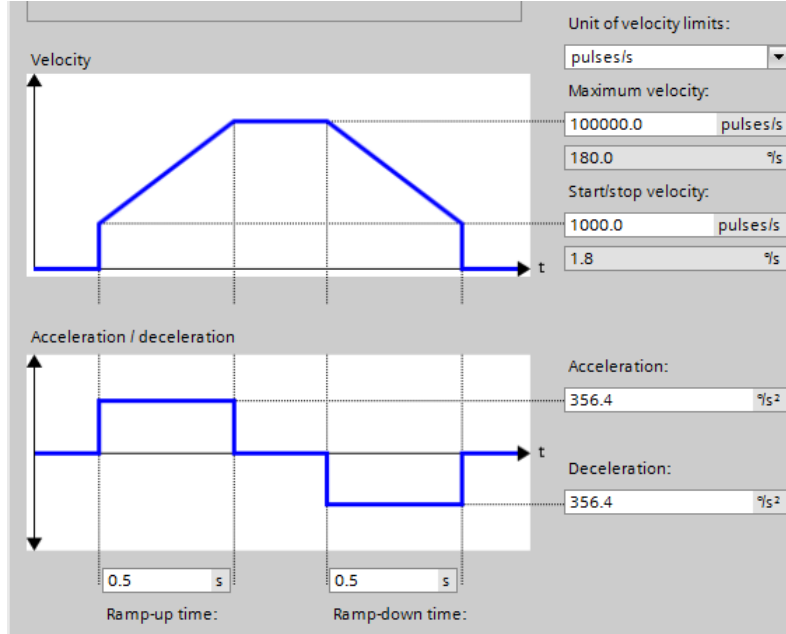
**Şekil 3.44:** Bir numaralı eksen darbe ve yön çıkışı.

$\Theta_1$ ,  $\Theta_2$ ,  $\Theta_3$  eksenlerinin mekanik ayarlarında, motorun bir tur dönüşü için gerekli darbe sayısı 10.000 darbedir. Redüktörün çevrim oranı  $i=20$  olduğu için, motorun bir turunda yükün ilerlemesi  $360^\circ/20=18^\circ$  dir ve her iki yönde dönmeye izin verilmiştir (Şekil 3.45).



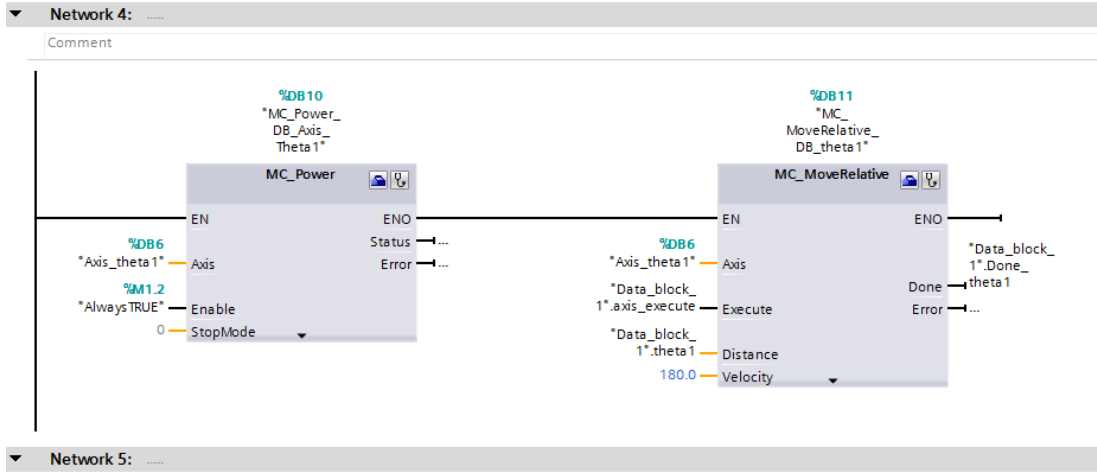
**Şekil 3.45:**  $\Theta_1$ ,  $\Theta_2$ ,  $\Theta_3$  mekanik ayarları.

$\Theta_1$ ,  $\Theta_2$ ,  $\Theta_3$  eksenlerinin dinamik ayarlarında, maksimum hız 100.000 darbe/s karşılık  $180^\circ/s$  ayarlanmıştır. Kalkış ve duruş hızları 1000 darbe/s ve maksimum hıza ulaşmak için ivmelenme 0,5 s olarak ayarlanmıştır. Böylelikle kalkış ve duruş ivmesi  $356,4 \text{ } ^\circ/s^2$  olarak belirlenmiştir (Şekil 3.46).



**Şekil 3.46:**  $\Theta_1$ ,  $\Theta_2$ ,  $\Theta_3$  dinamik ayarları.

Üç eksen için hareket kontrol (MC: Motion Control) komutları göreceli (relative) olarak yapılandırılmıştır. Motor milleri mevcut durumlarından, MC\_MoveRelative nesnesinin “Distance” girişine verilen açı bilgisi kadar dönerler (Şekil 3.47). Üç eksenin de hızları  $180^\circ/s$  olarak belirlenmiştir. Delta robotun uç işlevcisinin konum bilgisi, bilgisayardaki yazılım tarafından tutulmaktadır. Mevcut konumdan istenen konuma göreceli ilerleme, bilgisayardaki yazılım tarafından gerçekleştirilmektedir.



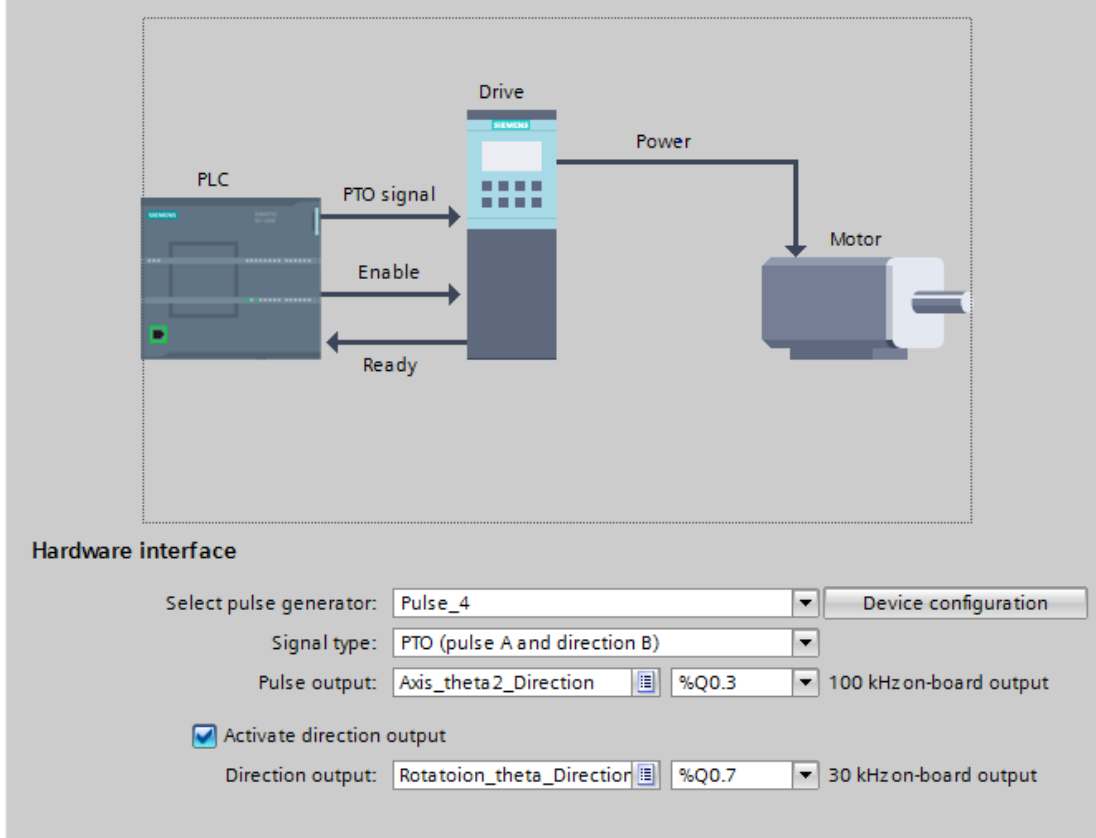
Şekil 3.47: Her bir eksen için hareket kontrol komutları.

### 3.8.10 Dönme Ekseni Kontrolü

Mermer karoları 0°, 90°, 180°, 270° döndürerek yerleştirmek için, dönme ekseni  $\Theta$  da hareketi sağlayan adım motoru FT42STH47-1684A teknik özellikleri: Nema17 0,5 Nm, 2,8 V, 1,6 A olan 4 telli adım motorudur [55].

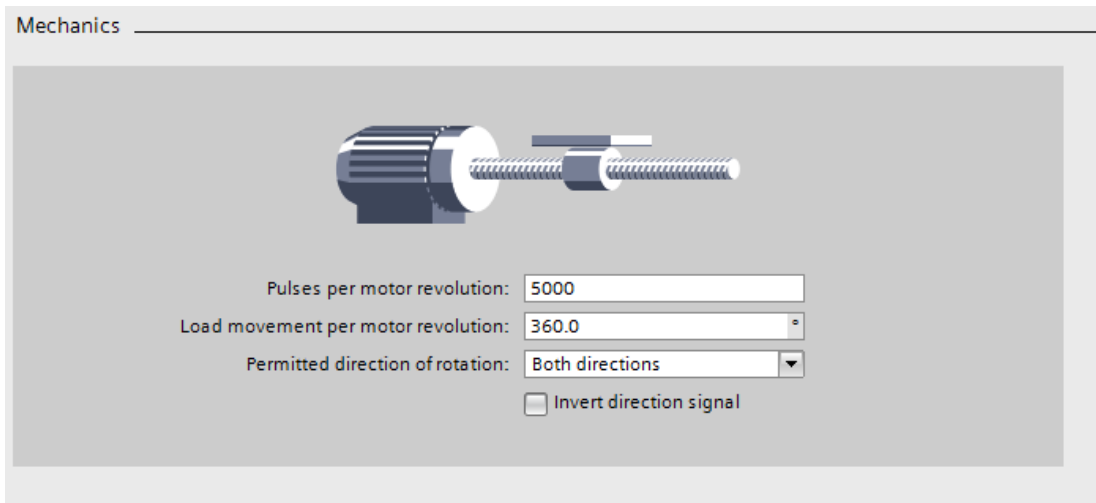
ZM-2H504 iki fazlı mikro adım motoru sürücüsüdür. Besleme gerilimi 24-50 V DC dir. Çıkışından elde edilebilecek darbe frekansı 200 kHz dir. DIP anahtarlar ile 400 adım/tur dan 25.000 adım/tur 15 farklı mikro adım seçeneği vardır. En fazla çıkış akımı 4,2 A olup DIP anahtarlar ile 1 A-4,2 A aralığında ayarlanabilir 8 farklı akım seçeneği vardır. En fazla 4,5 Nm motor sürebilir [56]. Uygulamada adım motor akımı 1,91 A ve mikro adım 5000 adım/tur olacak şekilde ayarlanmıştır. Motor hızını belirleyen “pulse output” Q0.3 çıkışından, dönme yönünü belirleyen “direction output” Q0.7 çıkışından verilecek şekilde ayarlanmıştır (Şekil 3.48).





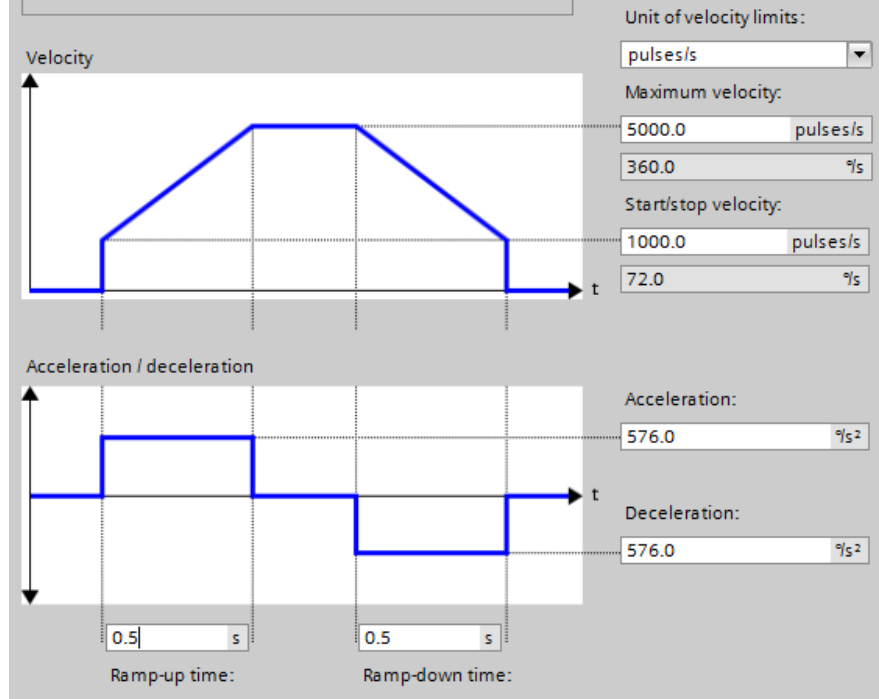
**Şekil 3.48:** Dönme eksenini darbe ve yön çıkışı.

Dönme ekseninde motorun bir tur dönüşü için gerekli darbe sayısı 5.000 darbedir. Motorun bir turunda yükün ilerlemesi, 360° dir ve her iki yönde dönmeye izin verilmiştir (Şekil 3.49). Vakum hortumunun sarılmaması için, tutucunun dönme açısı yazılım tarafından  $\pm 180^\circ$  ile sınırlandırılmıştır.



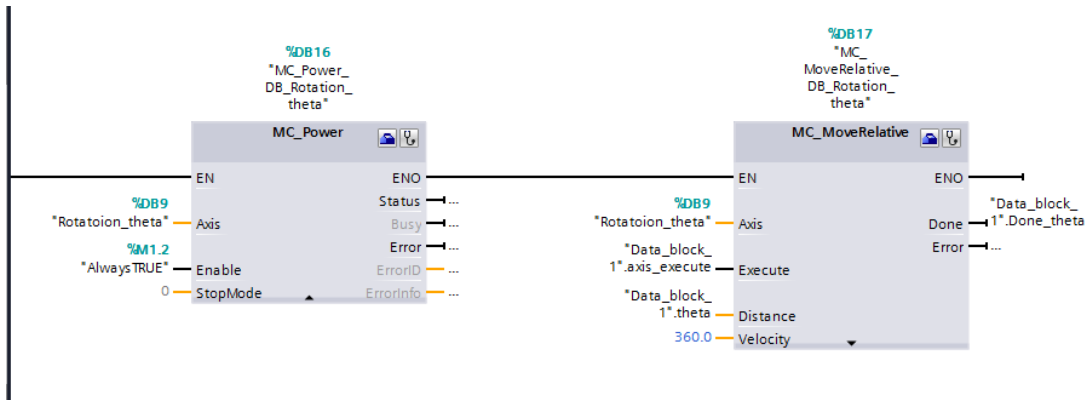
**Şekil 3.49:**  $\Theta$  dönme eksenini mekanik ayarları.

$\Theta$  dönme eksenlerinin dinamik ayarlarında, maksimum hız 5.000 darbe/s karşılık 360 °/s ayarlanmıştır. Kalkış ve duruş hızları 1000 darbe/s ve maksimum hıza ulaşmak için ivmelenme 0,5 s olarak ayarlanmıştır. Böylelikle kalkış ve duruştaki ivmelenme 576,0 °/s<sup>2</sup> olarak ayarlanmıştır (Şekil 3.50).



Şekil 3.50:  $\Theta$  Dönme eksenini dinamik ayarları.

Dönme eksenini için de hareket kontrol (MC: Motion Control) komutları göreceli (relative) olarak yapılandırılmıştır. Adım motoru mevcut durumundan MC\_MoveRelative nesnesinin “Distance” girişine verilen açı değeri kadar döner (Şekil 3.51). Dönme hızı 360°/s olarak belirlenmiştir. Delta robotun uç işlevcisindeki vakumlu tutucunun açısı, bilgisayardaki yazılım tarafından tutulmaktadır.



Şekil 3.51:  $\Theta$  Dönme eksenini göreceli hareket kontrol blokları.

### 3.8.11 PLC ile PC Arasında Haberleşme

PC ile PLC arasındaki haberleşme Profinet üzerinden TCP/IP haberleşme protokolü ile gerçekleştirilmektedir. Profinet, farklı üreticilerin cihazları arasında yüksek hızda, güvenli veri alışverişi sağlayan bir Ethernet standardıdır [57].

PLC de TCP/IP haberleşme ayarları yapılmış, PLC (local) 192.168.0.1 IP adresi verilmiş ve 2000 numaralı port iletişim için açılmıştır. PC ise 192.168.0.100 IP adresi ile aktif bağlantı kuracak şekilde ayarlanmıştır (Şekil 3.52).

The screenshot shows the 'Connection parameter' dialog box with the 'General' tab selected. The 'Local' side is configured with 'End point: PLC\_1', 'Interface: PLC\_1, PROFINET interface\_1[X1 : PN(LAN)]', 'Subnet: [empty]', 'Address: 192.168.0.1', 'Connection type: TCP', 'Connection ID (dec): 1', and 'Connection data: PLC\_1\_Connection\_DB'. The 'Partner' side is currently 'Unspecified'. The 'Address details' section shows 'Local Port: 2000' and 'Partner Port: [empty]'. There are radio buttons for 'Active connection establishment' on both sides, with the 'Partner' side being selected.

Şekil 3.52: TCP/IP haberleşme ayarları.

Güvenlik amacıyla, ters kinematik denklemler ile elde edilen eklem açılarının fiziksel sınırlar dahilinde olup olmadığını kontrol etmek amacıyla, eklem açıları ileri kinematik denklemlere konarak, istenilen Kartezyen koordinatları sağlayıp sağlamadığı kontrol edilir. İleri ve ters kinematik denklemler kullanılarak robot uç işlevcisini bulunduğu konumdan yeni konumuna göndermek için göreceli olarak hangi motorun hangi yönde ne kadar döndürülmesi gerektiğini hesaplayan bilgisayar programı, gerekli bilgileri TCP/IP haberleşme üzerinden PLC ye gönderir. Ters kinematik denklemler ile elde edilen  $\Theta_1$ ,  $\Theta_2$ ,  $\Theta_3$  göreceli dönme açıları,  $\Theta$  mermer karo döndürme açısı ve vakum aç/kapat komutu “kmt” birlikte paketlenerek PLC ye gönderilir. Açılar ve “kmt” her biri 100.000 ile çarpılarak 4 baytlık tam sayılara (int) dönüştürülürler. Tam sayılara ait baytlar 25 elemanlı bayt dizisinden oluşan tampon (buffer) belleğe yerleştirilirler. Csharp.net’in TcpClient() nesnesi

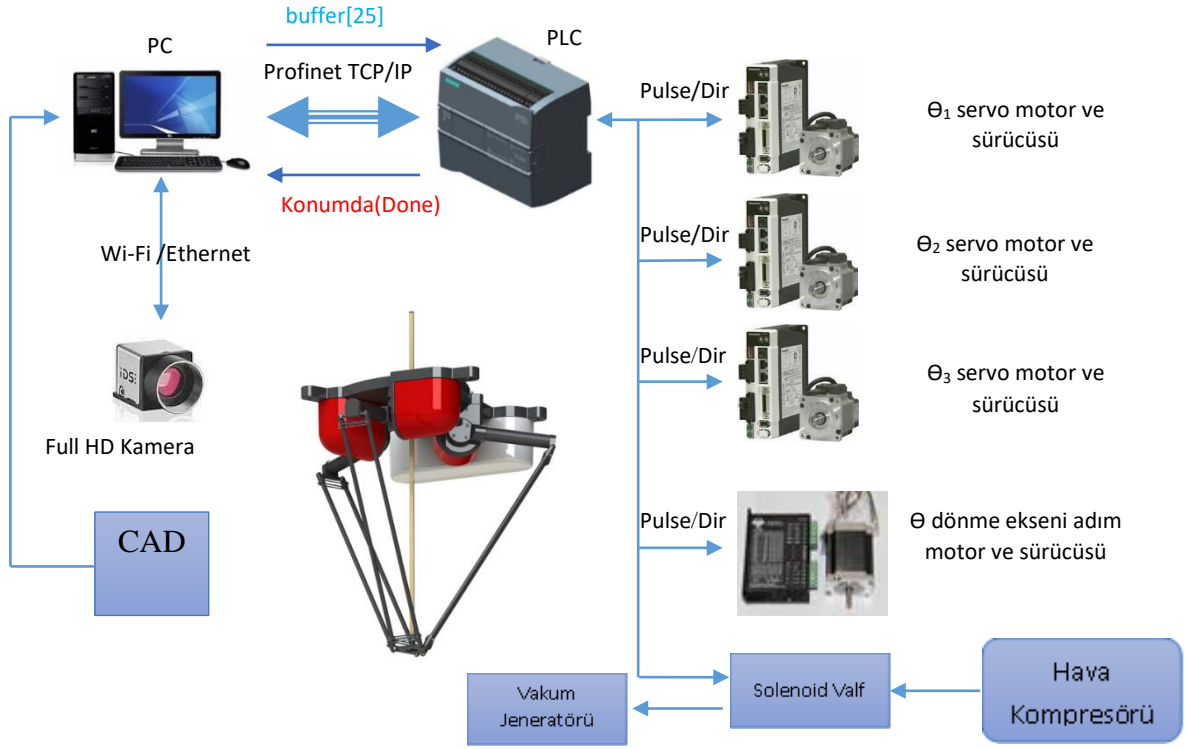
ile PLC ye bağlantı açılır. NetworkStream.Write() metodu ile tampon bellekteki bayt dizisi, PROFINET ağ üzerinden PLC ye gönderilir.

PLC, TCP/IP protokolü ile ağda 0. baytı 0xFF ve 1. baytı 0xAA olan 25 baytlık paketler beklemektedir. Beklenen paket geldiğinde, PLC delta robota ait açı bilgilerini okumak için paketi işler, dörder bayt birleştirerek, önce tamsayı olarak, daha sonra 100.000'e bölerek ondalık sayı olarak açılırları hesaplar. 22. Bayt "SağlamaToplamı" gönderilen verinin doğruluğunu kontrol etmek içindir. 23. ve 24. baytlar gerektiğinde kullanılmak üzere konmuş, bilgi taşımamaktadır. Paketin içeriği Tablo 3.10'da görülmektedir.

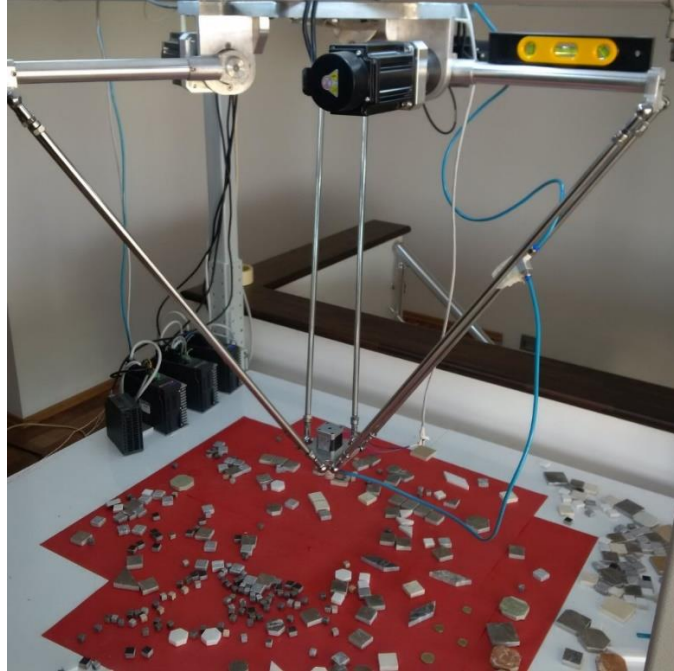
**Tablo 3.10:** Tampon (buffer) 25 baytlık paketin içeriği.

Bayt No	İçerik	Bayt No	İçerik	Bayt No	İçerik
0	0xFF	10	$\Theta_2$ bayt0	20	$\Theta$ bayt2
1	0xAA	11	$\Theta_2$ bayt1	21	$\Theta$ bayt3
2	kmt bayt0	12	$\Theta_2$ bayt2	22	SağlamaToplamı
3	kmt bayt1	13	$\Theta_2$ bayt3	23	0x00
4	kmt bayt2	14	$\Theta_3$ bayt0	24	0x00
5	kmt bayt3	15	$\Theta_3$ bayt1		
6	$\Theta_1$ bayt0	16	$\Theta_3$ bayt2		
7	$\Theta_1$ bayt1	17	$\Theta_3$ bayt3		
8	$\Theta_1$ bayt2	18	$\Theta$ bayt0		
9	$\Theta_1$ bayt3	19	$\Theta$ bayt1		

Delta robot uç işlevcisi, hedef konuma ulaşmadan önce yeni bir konumlama bilgisi bilgisayardan PLC ye gönderilmemelidir. Bu amaçla tüm eksenlere ait ( $\Theta_1$ ,  $\Theta_2$ ,  $\Theta_3$ ,  $\Theta$ ) MC\_MoveRelative nesnelerinin "Done" sinyali "1" olduğunda, PLC bir "Konumda" sinyali üreterek TCP/IP üzerinden bilgisayara gönderir. Bu sinyali almak için bilgisayar .net'in NetStream.Read() metodunu kullanarak PLC'den "Konumda" sinyalini bekler (Şekil 3.53). Delta robotun prototipi Şekil 3.54'de görülmektedir.



Şekil 3.53: Delta robotun PLC üzerinden PC ile kontrolü.

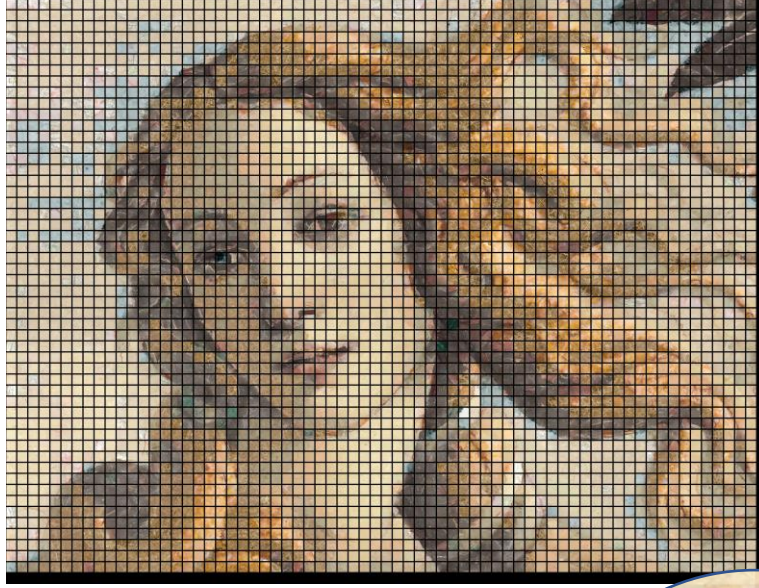


Şekil 3.54: Mozaik dizmek için dört serbestlik dereceli tut ve yerleştir robotu, delta robot prototipi.

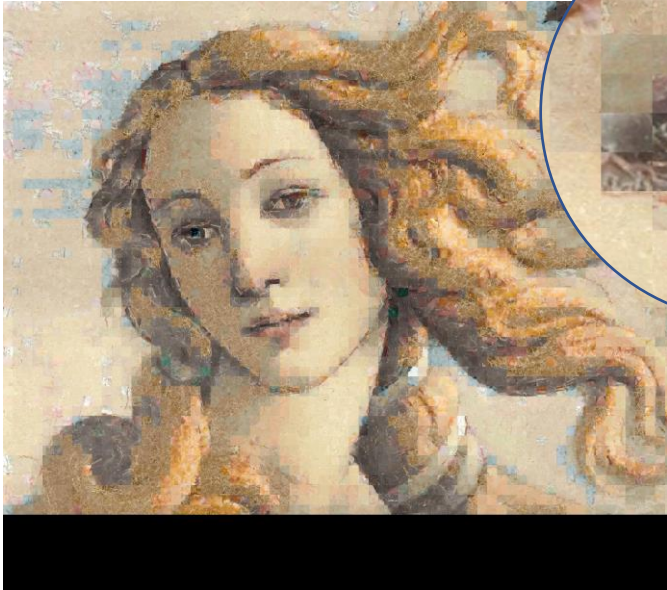
## 4. UYGULAMA

### 4.1 Aynı Ölçülerde Karolar ile Uygulama

Sandro Botticelli'nin Venüs'ün Doğuşu eserinin [58], Venüs'ün 2178x1672 piksellik baş bölgesi 32 piksellik 3536 adet görüntü parçasına bölünmüştür. 32 piksellik 14.696 adet mermer karo görüntüsünden Macar yöntemi ile seçilen 3.536 adet parçanın atanması ile mozaiği oluşturulmuştur. Şekil 4.1(a)'da oluşturulan mozaik derzli, Şekil 4.1(b)'de derzsiz olarak görülmektedir.



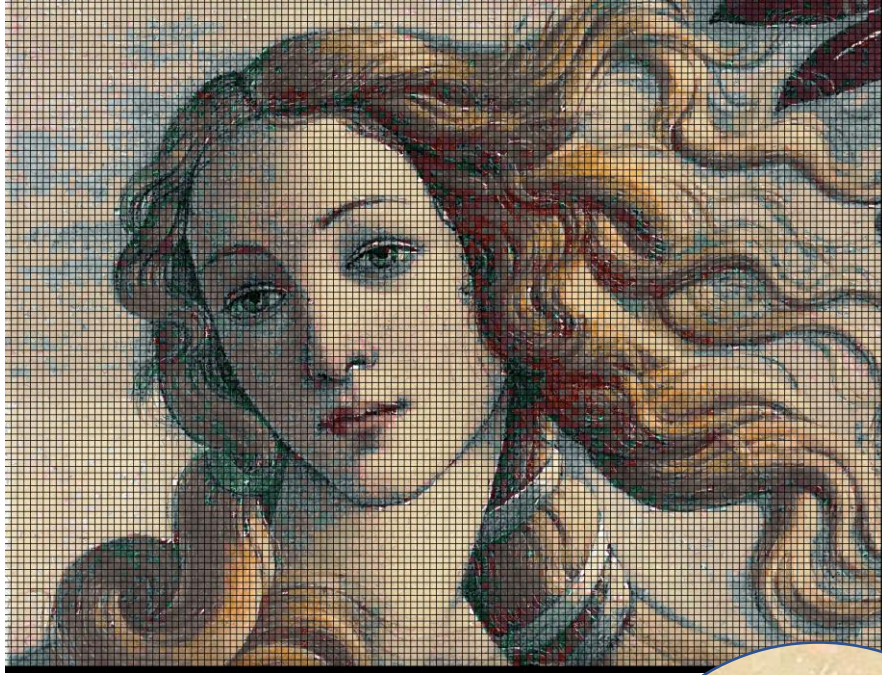
a) Derzli



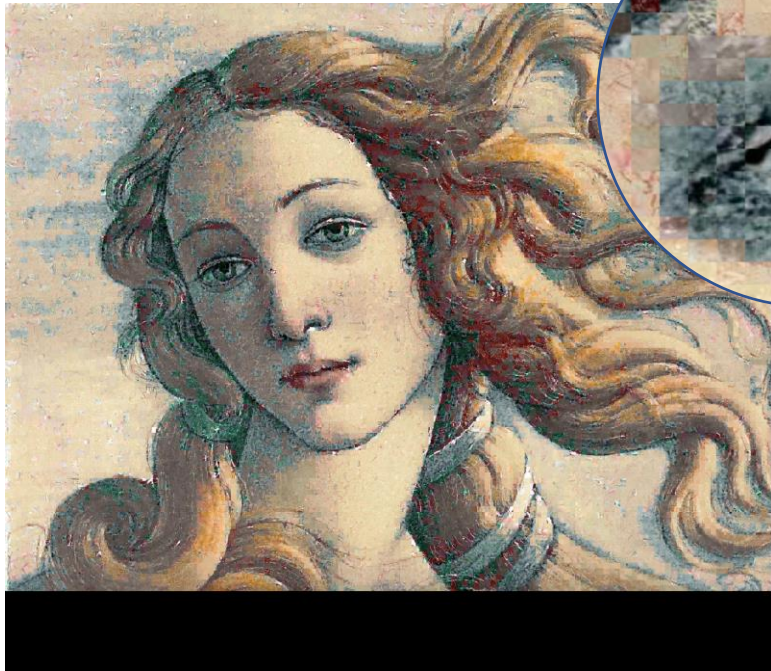
b) Derzsiz

**Şekil 4.1:** 32 piksel, 3.536 adet karo atanması ile oluşturulan mozaik (%15 ölçekli).

Aynı 2178x1672 piksellik Venüs görüntüsü, 16 piksellik 14.144 adet görüntü parçasına bölünmüştür. 16 piksellik 15.293 adet mermer karo görüntüsünden Macar yöntemi ile seçilen 14.144 adet parçanın ataması ile mozaiği oluşturulmuştur. Şekil 4.2(a)' da oluşturulan mozaik derzli, Şekil 4.2(b)'de derzsiz olarak görülmektedir.



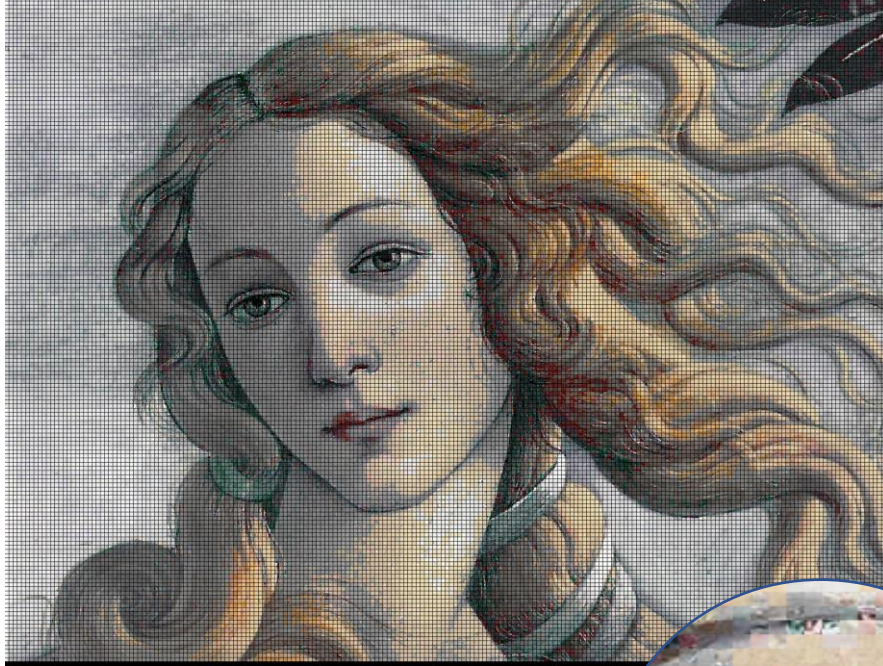
a) Derzli



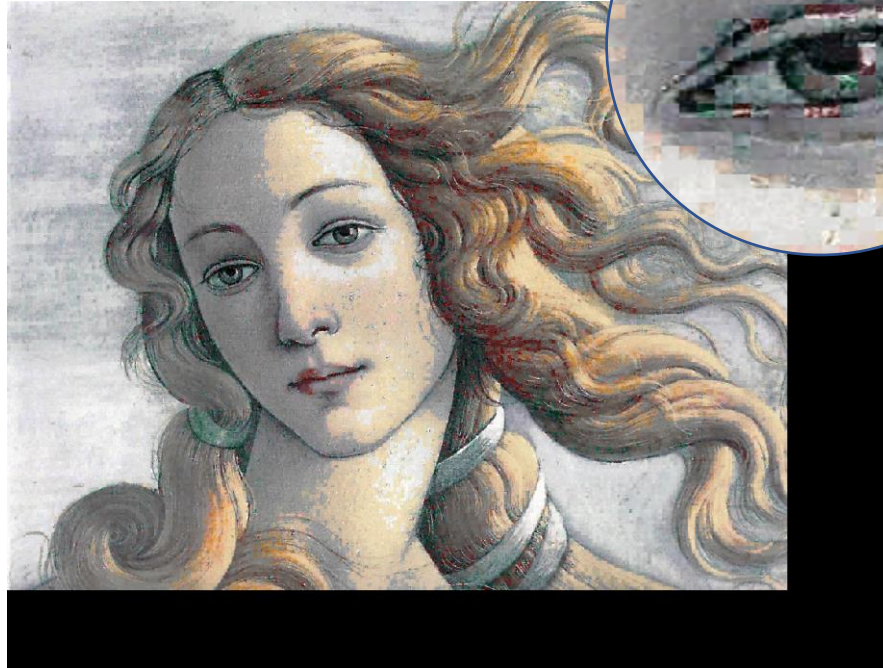
b) Derzsiz

**Şekil 4.2:** 16 piksel, 14.144 adet karo ataması ile oluşturulan mozaik (%18 ölçekli).

2178x1672 piksellik Venüs görüntüsü, veri tabanında sadece 34.868 adet mermer karo görüntüsü olduğundan, 1696x1280 piksele ölçeklenerek, 8 piksellik 33.920 adet görüntü parçasına bölünmüştür. 8 piksellik 34.868 adet mermer karo görüntüsünden Macar yöntemi ile seçilen 33.920 adet parçanın ataması ile mozaiği oluşturulmuştur. Şekil 4.3(a)' da oluşturulan mozaik derzli, Şekil 4.3(b)'de derzsiz olarak görülmektedir



a) Derzli



b) Derzsiz

Şekil 4.3: 8 piksel, 33.920 adet parçanın ataması ile elde edilen mozaik (%23 ölçekli).



Şekil 4.1, Şekil 4.2 ve Şekil 4.3 karşılaştırıldığında, karo ölçüleri küçüldükçe, mozaik kalitesinin arttığı görülmektedir.

#### 4.1.1 Karo sayısının ve karo ölçüsünün etkisi

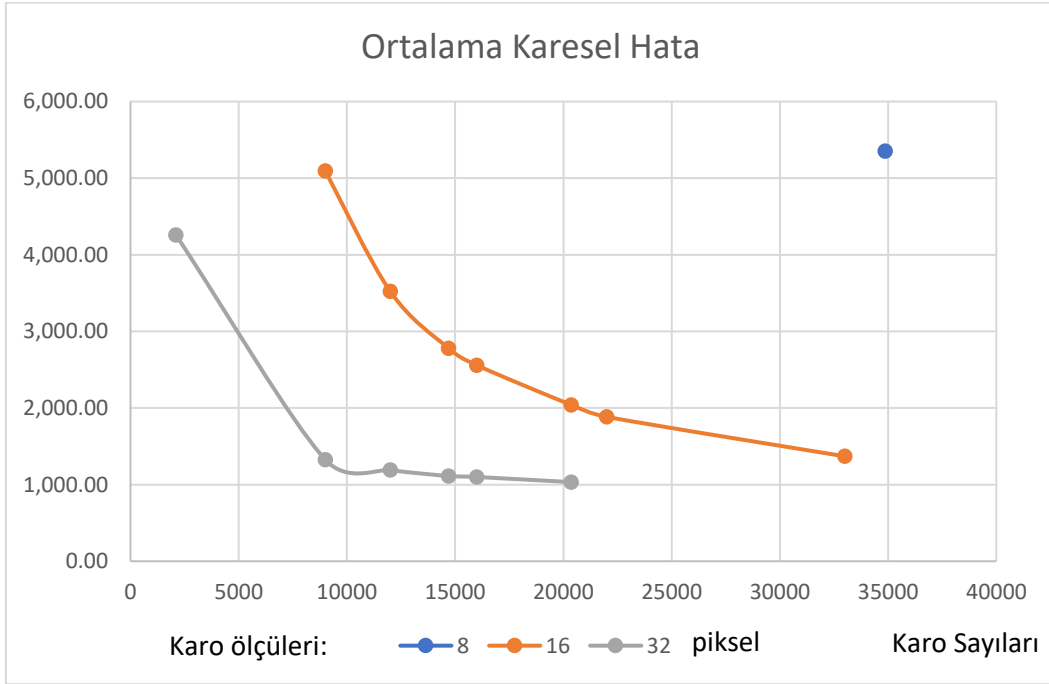
Karo ölçüsünün mozaik dizme hatasına etkisinin incelenmesi için Şekil 3.1 ile verilen renk gruplarından mermer levhalar kullanılarak, 8 piksel için 34.868 adet, 16 piksel için 34.455 adet, 32 piksel için 20.369 adetlik karo veri kümeleri oluşturulmuştur.

Karo sayısının mozaik dizme üzerine etkisini incelemek için, aynı ölçüde, çeşitli sayılarda karolar, stoktan rastgele seçilip, mozaik dizmede kullanılarak sonucun hatası hesaplanmıştır. Tablo 4.1 görüldüğü gibi 5 mozaik dizme denemesi, 16 piksel ölçüsü için 34,455 karodan rastgele seçilerek oluşturulmuş 9.000, 12.000, 16.000, 22.000, 33.000 adetlik karo alt kümeleri ile yapılmıştır. 6 mozaik dizme denemesi 32 piksel ölçüsündeki 20.369 adet karodan rastgele seçilerek oluşturulmuş karo alt kümeleri ile yapılmıştır. 8 piksel ölçüsündeki karolar ile yapılan deneme çok fazla zaman aldığından ikinci bir deneme yapılmamıştır.

**Tablo 4.1:** Karo ölçüsünün ve karo sayısının mozaik dizme hatasına etkisi

Karo Ölçüsü (Piksel)	Karo Stoku (Adet)	Karo Stokundan Rastgele Seçilen (Adet)	Kullanılan Karo (Adet)	Hesaplama Süresi saat:dakika saniye:mili s.	İki Resim Arası Karesel Hataların Toplamı (RGB)	İki Resim Arası Karesel Hataların Ortalaması MSE (RGB)
8	34.868	<b>34.868</b>	32.144	33:47:59:972	34.106.181.885	<b>5.350,63</b>
16	34.455	<b>9.000</b>	8.036	00:47:48:634	32.476.394.742	<b>5.094,95</b>
16	34.455	<b>12.000</b>	8.036	00:58:49:531	22.444.001.609	<b>3.521,05</b>
16	34.455	<b>16.000</b>	8.036	02:06:00:370	16.304.970.736	<b>2.557,95</b>
16	34.455	<b>22.000</b>	8.036	02:58:06:261	12.012.993.849	<b>1.884,62</b>
16	34.455	<b>33.000</b>	8.036	04:38:52:934	8.730.522.108	<b>1.369,66</b>
32	20.369	<b>2.100</b>	2.009	00:09:19:351	27.140.611.384	<b>4.257,86</b>
32	20.369	<b>9.000</b>	2.009	00:18:53:169	8.452.329.184	<b>1.326,02</b>
32	20.369	<b>12.000</b>	2.009	00:28:10:175	7.583.282.750	<b>1.189,68</b>
32	20.369	<b>14.696</b>	2.009	00:38:42:160	7.094.190.916	<b>1.112,95</b>
32	20.369	<b>16.000</b>	2.009	00:32:54:424	7.010.942.311	<b>1.099,89</b>
32	20.369	<b>20.369</b>	2.009	00:47:12:027	6.581.938.011	<b>1.032,59</b>

Şekil 4.4’de görüldüğü gibi veri kümesindeki karo sayısı arttıkça, dizilen mozağin ortalama karesel hatası azalmaktadır.



**Şekil 4.4:** Karo sayısının ve karo ölçüsünün dizme hatası üzerine etkisi

Yapılan denemelerde kullanılan 1338x1588 piksel ölçülerinde renkli görüntü Şekil 4.5’de %22 ölçekli olarak görülmektedir.



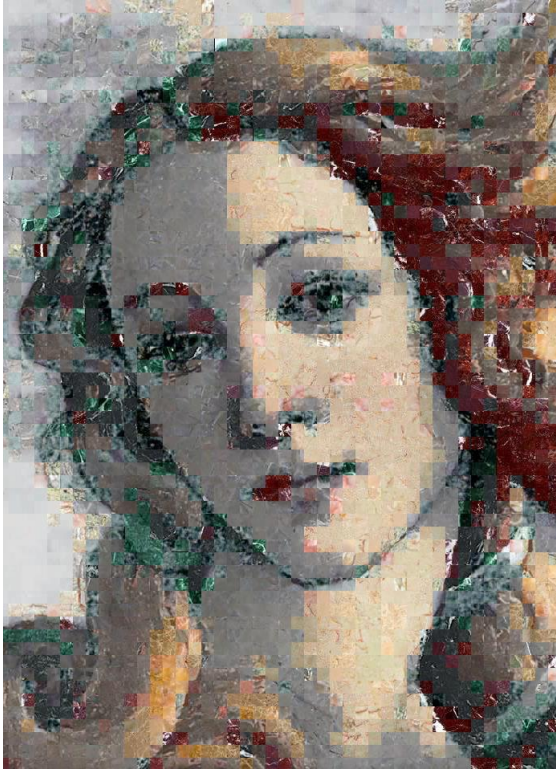
**Şekil 4.5:** Orijinal Görüntü [59]

16 piksel ölçülerinde 22.000 karodan 8.036 adeti kullanılarak elde edilen mozaik Şekil 4.6 da %38 ölçekli, derzsiz olarak görülmektedir. Ortalama karesel hatası 1884,62 dir.

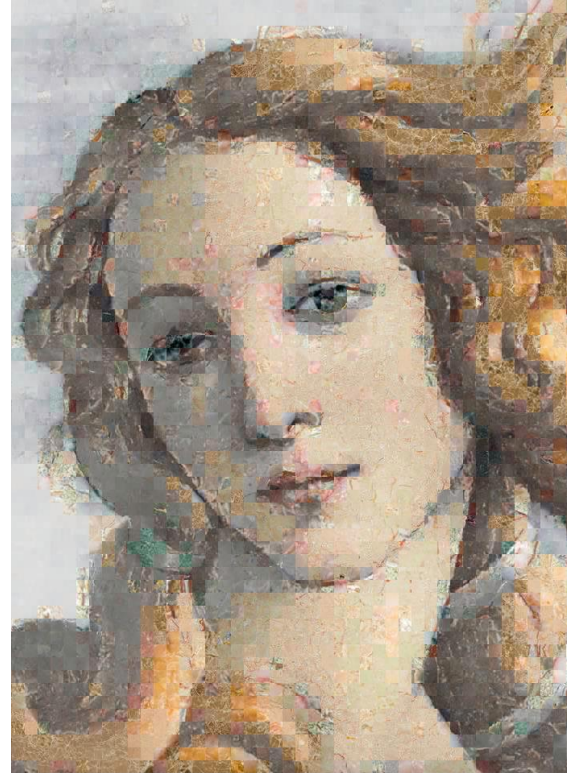


**Şekil 4.6:** 16 piksel ölçülerinde 22.000 karodan 8.036 adeti kullanılarak elde edilen mozaik

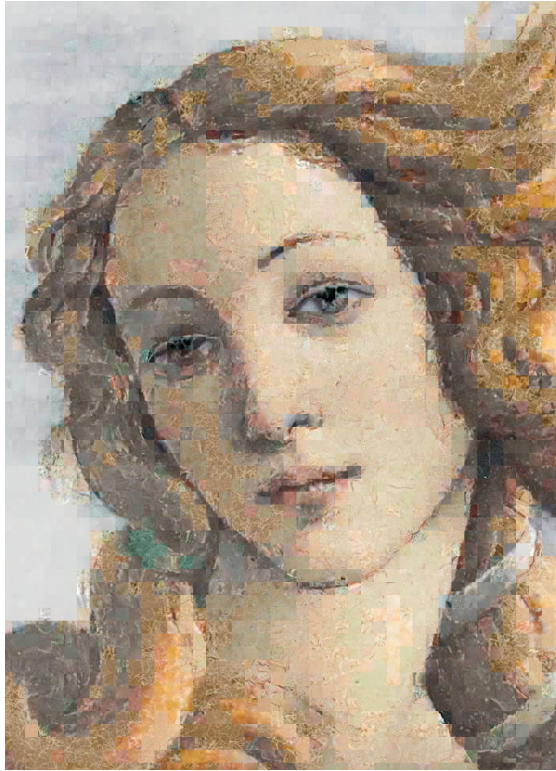
Veri kümesindeki karo sayısının mozaik kalitesine etkisinin daha iyi görülmesi için, Venüs'ün 2178x1672 görüntüsü [58], 1250x960 piksele ölçeklenmiş ve 4680, 10.000,15.000 ve 21.041 adet 16 piksel karo veri kümelerinden 4680 adet karo kullanılarak dizilmiştir. Elde edilen mozaikler ve ortalama karesel hataları (MSE) Şekil 4.7'de sunulmuştur. Şekil 4.8'de aynı sonuçlar derzli olarak görülmektedir. Sonuçların görüntüleri %40 ölçeklidir.



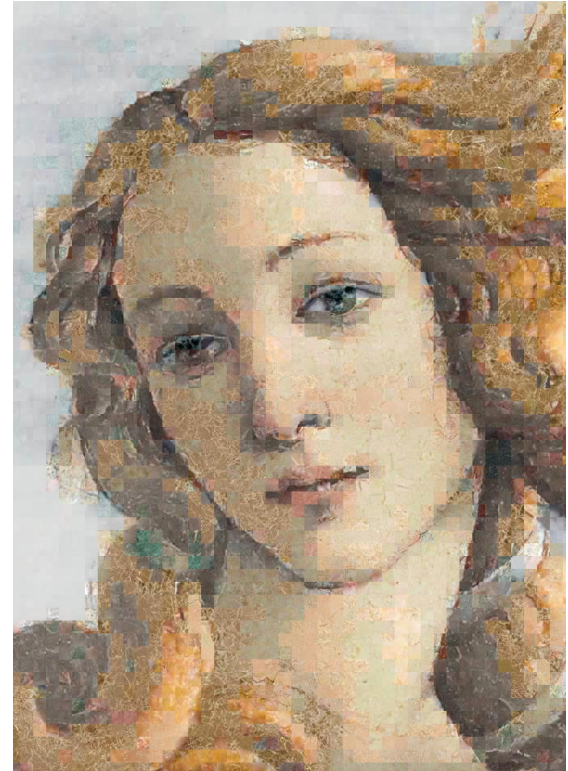
a) 4680 karodan, MSE: 2426



b) 10.000 karodan MSE: 991

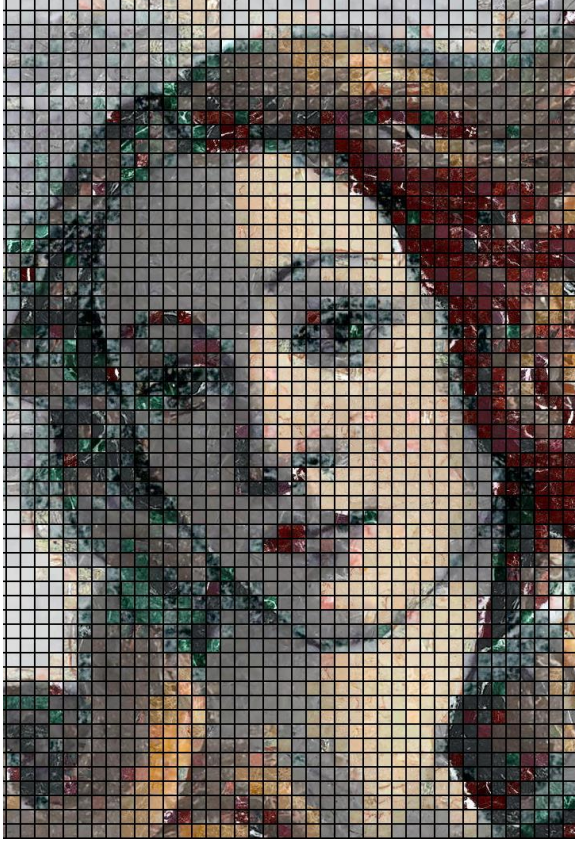


c) 15.000 karodan, MSE: 893

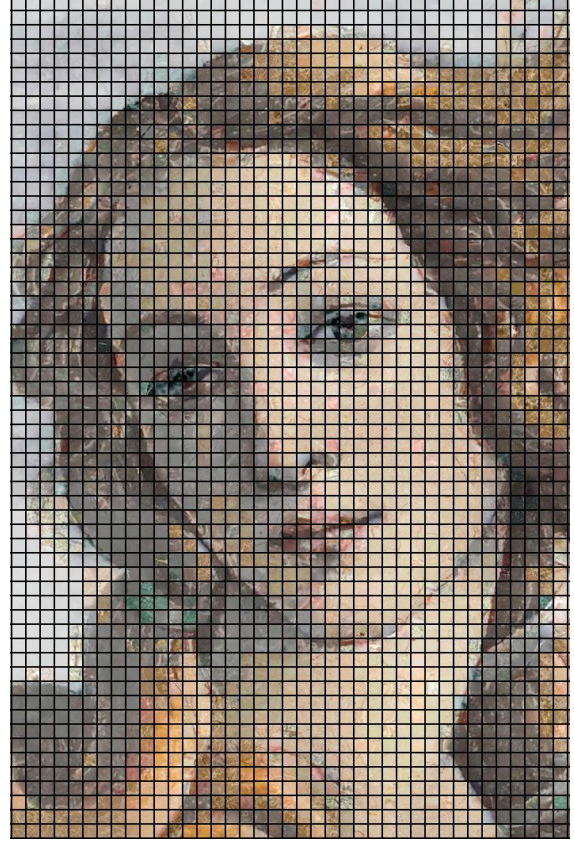


d) 21.041 karodan, MSE: 838

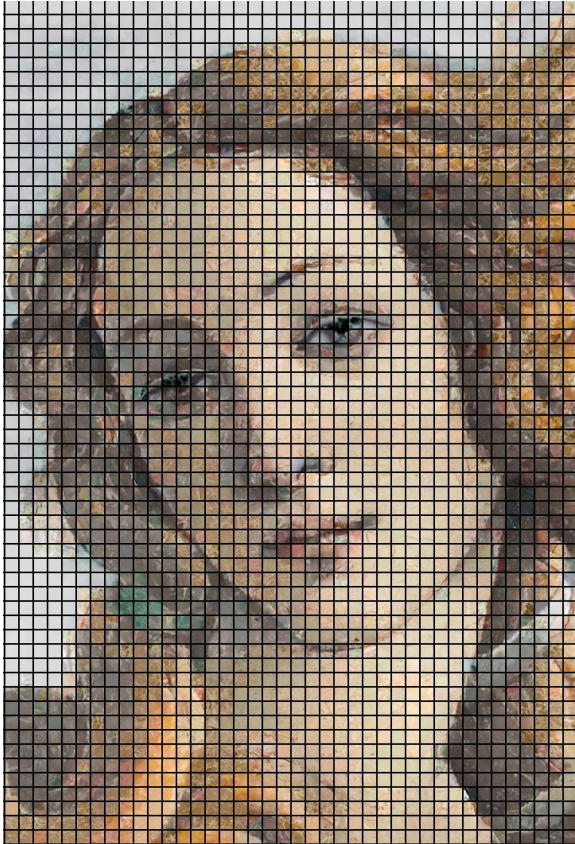
**Şekil 4.7:** Veri kümesindeki karo sayısının mozaik kalitesine etkisi (derzsiz).



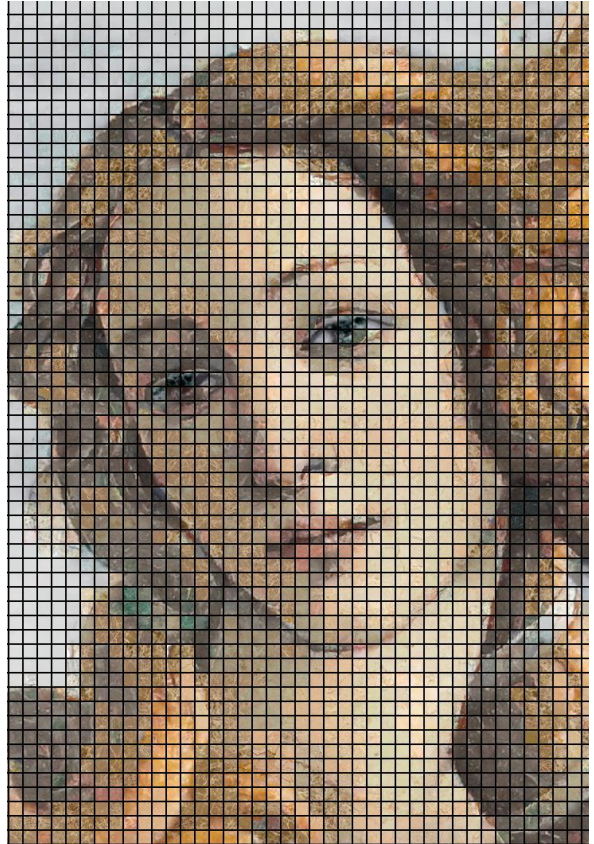
a) 4680 karodan, MSE: 2426



b) 10.000 karodan MSE: 991



c) 15.000 karodan, MSE: 893



d) 21.041 karodan, MSE: 838

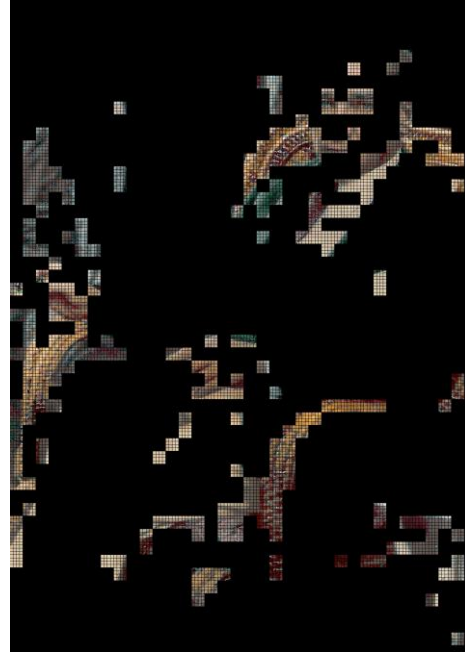
**Şekil 4.8:** Veri kümesindeki karo sayısının mozaik kalitesine etkisi (derzli).

## 4.2 Farklı Ölçülerde Karolar ile Uygulama

Mozaik dizme sırasında farklı ölçülerde karoların kullanılabilmesi için geliştirilen yöntem, Şekil 4.9(a)'da 1120x1600 piksel ölçülerinde, %18 ölçekli görülen renklendirilmiş Atatürk görüntüsü [60] üzerinde denenmiştir. 13.136 adet 8 piksel karodan 4.176 adet, 12.857 adet 16 piksel karodan 5.956 adet kullanılmıştır. Atama 5 saat 15 dakika sürmüştür. Sonuçlar derzli olarak Şekil 4.9'da, ve derz olmaksızın Şekil 4.10'da görülmektedir.



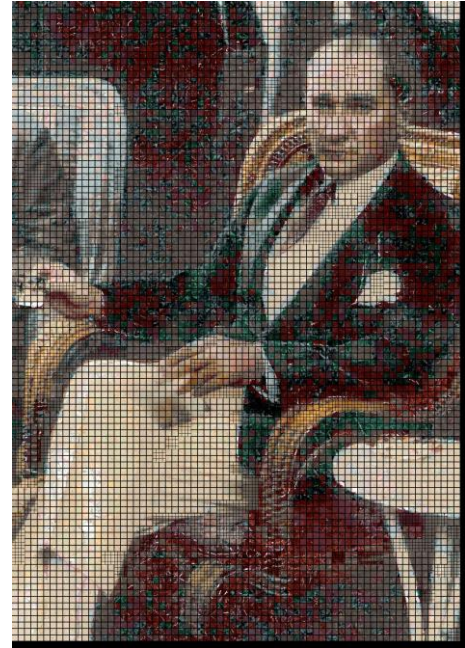
a) Mozaik dizilecek görüntü



b) 4176 adet 8 piksel karo ataması

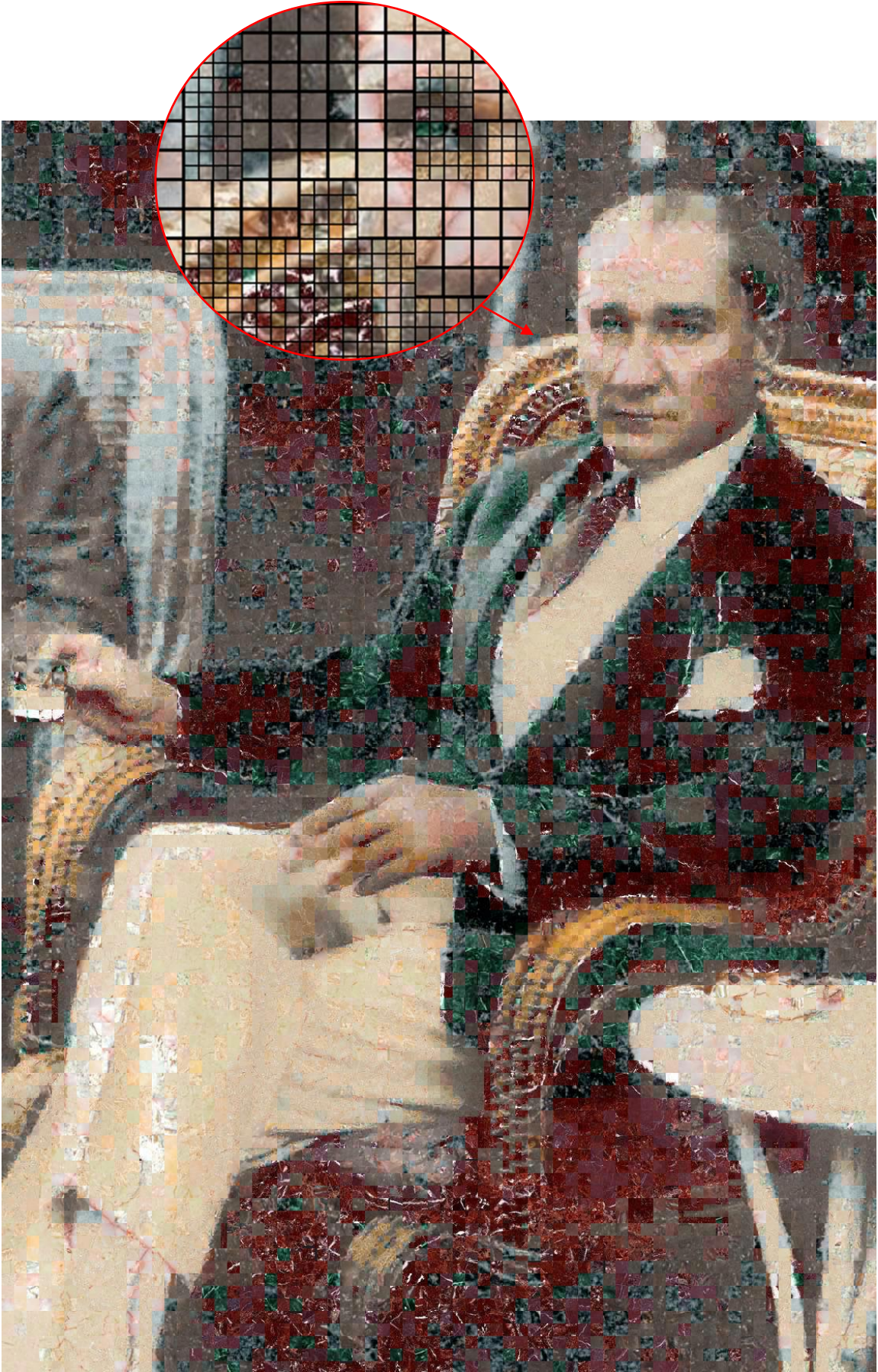


c) 5956 adet 16 piksel karo ataması



d) 8 ve 16 piksel karolar birlikte

**Şekil 4.9:** Farklı ölçülerde karoların görüntü parçalarına atanması, derzli yerleşim.



**Şekil 4.10:** Farklı ölçülerde karoların görüntü parçalarına atanması 4176 adet 8 piksel, 5956 adet 16 piksel karonun yerleşimi derz olmaksızın (%51 ölçekli).

### 4.3 Nesne İçeren Görüntü Parçalarına Öncelik Verilmesi

#### 4.3.1 Uygulama 1

Orijinal görüntü 960x1344 piksel Atatürk portresi [61], 21.041 adet 16 piksel karodan 5.040 adet yerleştirilerek dizilmiştir. YOLO nesne detektörü ile algılanan “kişi” Şekil 4.11(a)’da kutulanmıştır ancak neredeyse görüntünün tamamını kapsadığından kullanılmamıştır. OpenPose yüz ile algılanan yüz anahtar noktaları Şekil 4.11(b)’de görülmektedir. Şekil 4.11(c)’de yüz anahtar noktaları doğrular ile birleştirilip, morfolojik kabartma ve kapama uygulandıktan sonra kullanıcının son şeklini verdiği kalıp görüntü görülmektedir. 1418 adet görüntü parçası nesne olarak işaretlenmiştir.



a) YOLO ile nesne algılama b) OpenPose yüz noktaları c) Kalıp görüntü

**Şekil 4.11:** YOLO ve OpenPose ile algılama ve kalıp görüntü.

Elde edilen mozaik Şekil 4.12(a)’da derssiz ve %51 ölçeklenmiş olarak görülmektedir. Şekil 4.14 (b)’de ise, aynı görüntüye, aynı karoların sadece Macar yöntemi kullanılarak atanması durumunda elde edile mozaığın sol göz bölgesi, karşılaştırma amacı ile verilmiştir. Sınırlı sayıda karo olduğunda, nesne içeren görüntü parçalarını etiketleyen yöntem kullanılarak, sadece Macar yöntemi kullanılarak yapılan atamalardan daha uygun karolar yüz bölgesine yerleştirilmektedir.





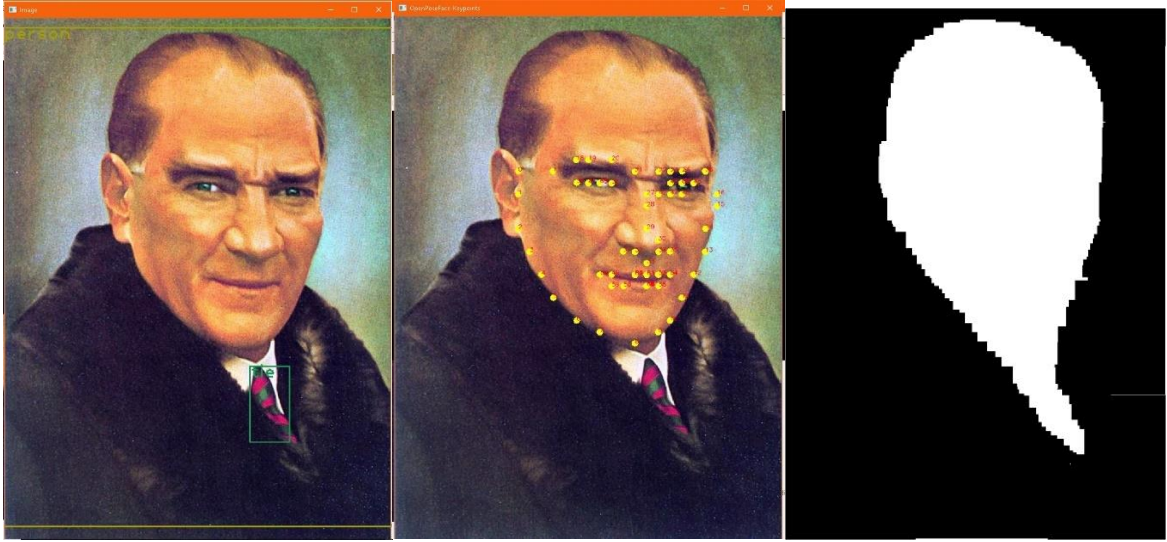
a)

**Şekil 4.12:** a) Nesne içeren görüntü parçalarını etiketleyen yöntem ile elde edilen mozaik.  
b) Aynı karoların, aynı görüntüye sadece Macar yöntemi ile atanması durumunda sol göz bölgesi.

### 4.3.2 Uygulama 2

Orijinal görüntü 980x1367 piksel Atatürk portresi [62], görüntü parçalarına bölünmek üzere 960x1344 piksel ölçülerinde kırılmıştır. 21.041 adet 16 piksel karodan 5040 adet karo kullanılarak dizilmiş mozaik Şekil 4.14’de görülmektedir. YOLO nesne detektörü ile algılanan “kişi” ve “kravat” Şekil 4.13(a)’da kutulanmıştır. Kişi kutusunun alanı,

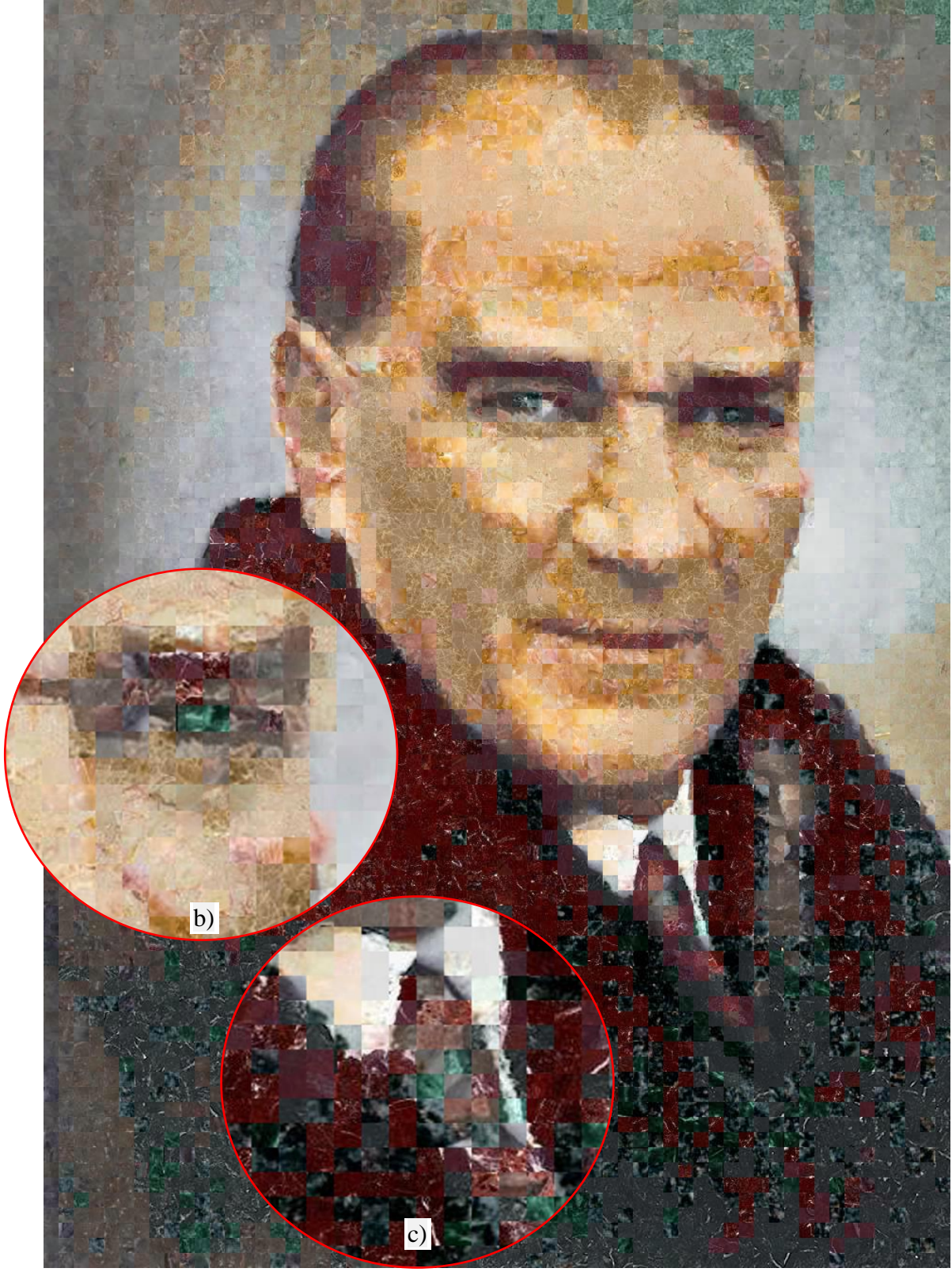
görüntünün alanının yarısından büyük olduğundan kullanılmamış, kravat kullanılmıştır. OpenPose yüz ile algılanan yüz anahtar noktaları Şekil 4.13(b)'de görülmektedir. Şekil 4.13(c)'de Yolo'nun kravat kutusu ve yüz anahtar noktaları doğrular ile birleştirilip, morfolojik kabartma ve kapama uygulandıktan sonra kullanıcının son şeklini verdiği kalıp görüntü görülmektedir. 1843 adet görüntü karesi nesne olarak işaretlenmiştir.



a) YOLO ile nesne algılama      b) OpenPose yüz      c) Kalıp görüntü

**Şekil 4.13:** YOLO ve OpenPose ile algılama ve kalıp görüntü.

Atatürk portresinin, 21.041 adet 16 piksel karo içinden seçilen 5040 adet karonun atanması ile elde edilen mozaik Şekil 4.14(a)'da görülmektedir. Şekil 4.14(b) ve (c)'de, aynı görüntüye, aynı karoların sadece Macar yöntemi kullanılarak atanması durumunda elde edile mozaığın sırasıyla sağ göz ve kravat bölgeleri, karşılaştırma amacı ile verilmiştir. Sınırlı sayıda karo olduğunda, nesne içeren görüntü parçalarını etiketleyen yöntemin, yüz ve kravat bölgelerinde, sadece Macar yöntemi kullanılarak yapılan atamalardan daha uygun karolar yerleştirdiği Şekil 4.14'de görülmektedir.

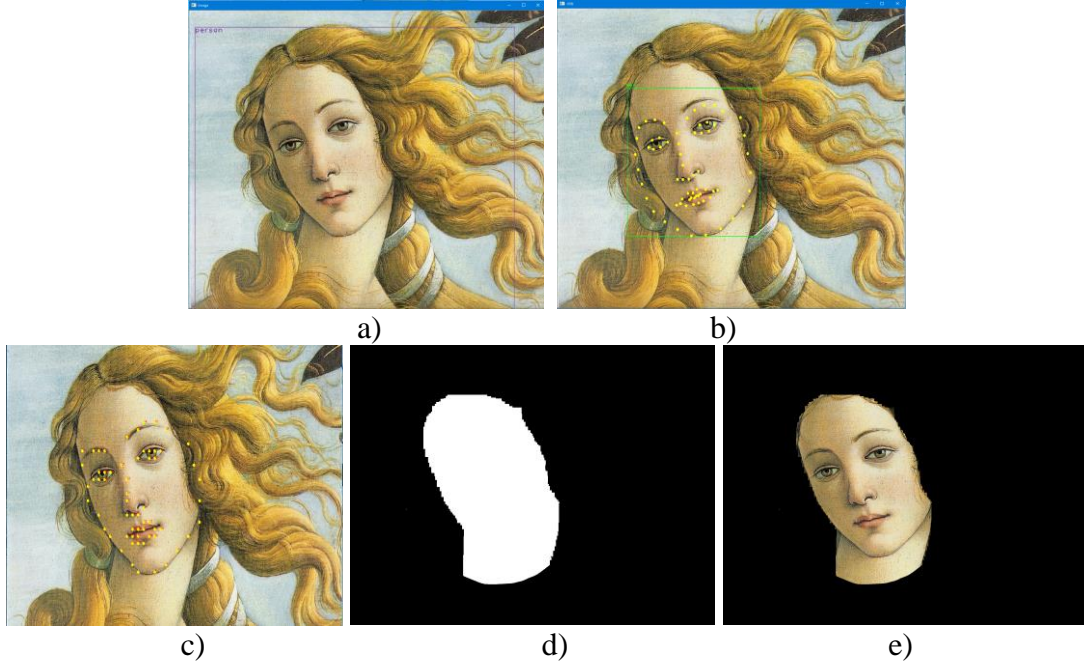


a)  
**Şekil 4.14:** a) 21.041 adet 16 piksel karodan 5.040 adet karonun atanması ile elde edilen mozaik. b) Aynı karoların, aynı görüntüye sadece Macar yöntemi ile atanması durumunda sağ göz bölgesi ve c) Kravat

### 4.3.3 Uygulama 3

Nesne içeren görüntü parçalarının etiketlenmesi yöntemi, sınırlı sayıda karo darboğazını aşmak için önerilmiştir. Bu durumu daha iyi görebilmek için, Venüs'e [58] ait görüntünün, 2178x1672 piksellik bölgesi, 32 piksellik 3536 adet görüntü parçasına bölünmüştür. Şekil 3.1 ile verilen renk gruplarından mermer levhalar kullanılarak elde edilen 20.369 adet 32 piksel karo içinden rastgele seçilen 3.536 adet karo dizme için kullanılmıştır. Nesne içeren görüntü parçalarını etiketleyen yöntem ile sadece Macar yöntemi ile dizilen mozaikler karşılaştırılmıştır. Macar yöntemi ile yapılan atamalar, Şekil 4.16(a)'da görülmektedir.

Nesne içeren görüntü parçalarını etiketlemek için uygulanan YOLO nesne detektörü ile algılanan "kişi" Şekil 4.15(a)'da çerçeve içine alınarak işaretlenmiştir. Şekil 4.15(b)'de dlib yüz detektörü ile algılanan yüz çerçevesi yeşil ve yüzün anahtar noktaları sarı renkte işaretlenmiştir. Şekil 4.15(c)'de OpenPose ile algılanan yüz anahtar noktaları işaretlenmiştir. Algılanan anahtar noktalara ve noktaları birleştiren doğrulara morfolojik kabartma ve kapama uygulandıktan sonra elde edilen nesne silüeti kullanıcı tarafından biçimlendirilerek Şekil 4.15(d)'de görülen son haline getirilmiştir. Şekil 4.15(d)'deki kalıp görüntü ile kesişen görüntü parçaları etiketlenmiştir. Etiketlenmiş görüntü parçaları, Şekil 4.15(e)'de görülmektedir. Nesne içeren görüntü parçaları ile uzaklık matrisi güncellenmiştir. Güncellenmiş uzaklık matrisinin Macar yöntemi ile ataması sonucu elde edilen mozaik Şekil 4.16(c)'de görülmektedir.

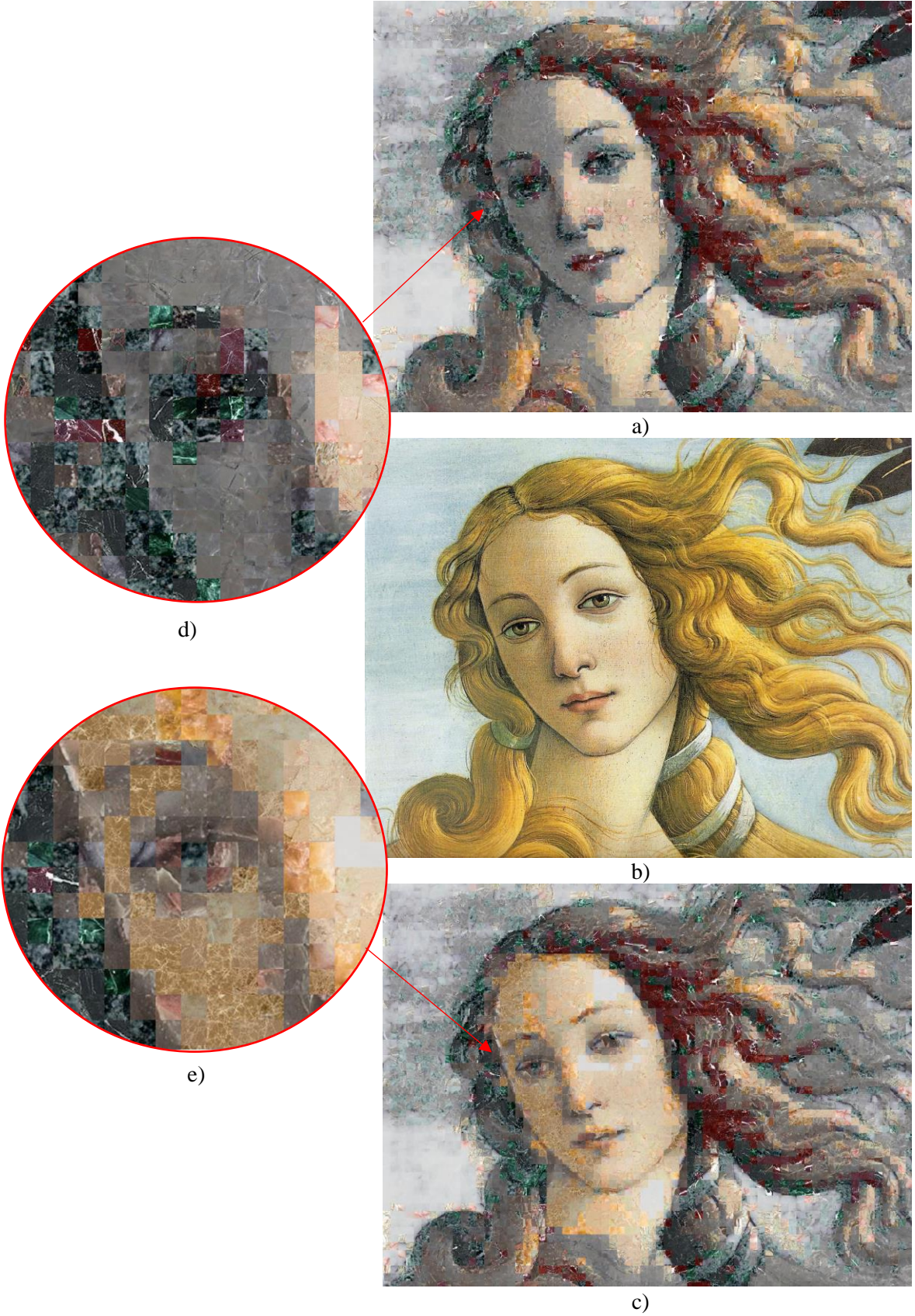


**Şekil 4.15:** a) Orijinal görüntüden [58] YOLO ile algılanan kişi b) Dlib ile algılanan yüz anahtar noktaları ve çerçevesi c) Openpose ile algılanan yüz anahtar noktaları d) Nesnenin kalıp görüntüsü e) Kalıp görüntü ile kesişen 32 piksel görüntü parçaları.

Etiketleme yapılmaksızın sadece Macar yöntemi kullanılarak dizilen mozaik Şekil 4.16 (a)'da görülmektedir. Görüntünün toplam hatası minimize edilmiştir. 3536 adet karo görüntüyü ancak kapladığından, yerleştirme hatası yüksek, gürültülü bir sonuç elde edilmiştir. Sarı ve bej karolar saçlarda kullanılmıştır ve yüze daha soluk ve gri tonlarda karolar yerleştirilmiştir. Şekil 4.16(b) orijinal Venüs görüntüsü görülmektedir.

Nesne olarak etiketlenen görüntü parçalarına, arka plana göre daha uygun karolar atayan yöntem ile dizilen mozaik sonucu Şekil 4.16(c)'de görülmektedir. Yüzdeki görüntü parçalarının orijinal görüntüdeki rengine daha yakın olduğu ve yine yüze ait görüntü parçalarında, gürültünün daha az olduğu görülmektedir. Yöntemin yüz için, Şekil 4.16(a)'da görülen, sadece Macar yöntemi kullanılarak dizilen mozaik sonucundan daha başarılı olduğu görülmektedir.

Şekil 4.16(a) ve Şekil 4.16(c)'de verilen sonuçların göz bölgesi büyütülmüş olarak, Şekil 4.16(d) ve Şekil 4.16(e)'de sunulmuştur. Sınırlı sayıda karo ile çalışılmasından kaynaklanan yüksek yerleştirme hatası daha belirgin olarak Şekil 4.16(d)'de görülmektedir. Yüze ait görüntü parçalarının etiketlendiği Şekil 4.16(e)'de yerleştirilen 32 piksel karoların, Şekil 4.16(d) ile karşılaştırıldığında orijinal rengine daha yakın ve daha az gürültülü olduğu görülmektedir.



**Şekil 4.16:** a) Sadece Macar yöntemiyle atama b) Orijinal Görüntü [58]. c) Nesne içeren görüntü parçaları etiketlenerek atama. d) Macar yöntemi ile elde edilen sonucun büyütülmüş göz bölgesi. e) Nesnelere etiketlenerek elde edilen sonucun büyütülmüş göz bölgesi

## 5. SONUÇ ve ÖNERİLER

### 5.1 Bulgular

Nesnelerin bölütlenerek en uygun yapı taşları ile dizilmesi alanında çalışan araştırmacılara yardımcı olması amacıyla, çalışmada denenilen, ancak yararlı olmayan fikir ve yöntemlerin de sunulmasının olumlu olacağı düşünülmüştür.

#### 5.1.1 En Benzer Karoyu Bulmak İçin Çeşitli Uzaklık Hesaplama Yöntemleri ve Yapısal Benzerlik Yöntemi:

$X$  mermer karo görüntüsü,  $Y$  görüntü parçası,  $\mu$  beklenen değer (ortalama) ve  $\Sigma$  kovaryans matrisi olmak üzere,  $X$  ve  $Y$  görüntüleri arasındaki uzaklığın hesaplanması için çeşitli uzaklık hesaplama yöntemleri denenmiştir [63]:

Mahalanobis Uzaklığı Eşitlik (5.1) ile verilmiştir:

$$d^2(X, Y) = (\mu_X - \mu_Y) + \left[ \frac{1}{2} (\Sigma_X^{-1} - \Sigma_Y^{-1}) \right] (\mu_X - \mu_Y)^T \quad (5.1)$$

Frobenius Uzaklığı Eşitlik (5.2) ile verilmiştir:

$$d^2(X, Y) = \|\mu_X - \mu_Y\|^2 + \|\Sigma_X - \Sigma_Y\|^2 \quad (5.2)$$

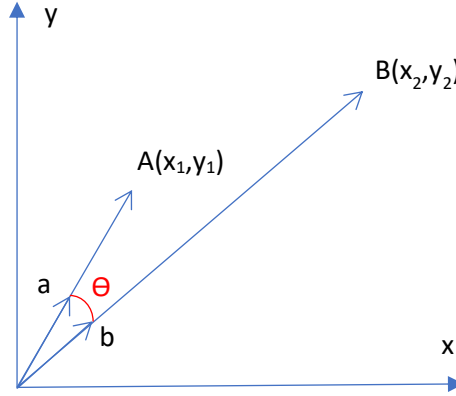
Frechet Uzaklığı Eşitlik (5.3) ile verilmiştir:

$$d^2(Y, X) = \|\mu_X - \mu_Y\|^2 + tr \left[ \Sigma_X + \Sigma_Y - 2(\Sigma_X \Sigma_Y)^{\frac{1}{2}} \right] \quad (5.3)$$

Yukarıda listelenen uzaklık hesaplamaları, karo görüntüsünün piksellerini bir grup olarak ele alarak, bu grubun kütle merkezini ve grup içi dağılımını da hesaba katarak, iki grup arası uzaklığı hesaplamaktadırlar. Bu hesaplamalar ondalık sayı kullanılmasını gerektirdiğinden işlemler uzun sürmekte ve yapısal bilgi kaybolmaktadır.

Eşitlik (5.4) ile verilen kosinüs uzaklığı, birbirlerine benzerliği hesaplanmak istenen vektörleri birim vektörlere dönüştürerek, uzaklığı ihmal eder, iki vektör arasındaki açıyı ( $\Theta$ )

benzerlik ölçüsü olarak kullanır (Şekil 5.1). Karo görüntülerinin benzerliğini ölçmek için uygun değildir.



Şekil 5.1: A ve B vektörleri arasında kosinüs benzerliği  $\Theta$ .

$$\cos(\theta) = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (5.4)$$

Oysa görüntünün mozağini dizmek için, görüntünün ortalaması (bias, DC bileşen) renk ve parlaklık açısından çok önemli rol oynamaktadır. Fakat kosinüs uzaklığı bu bilgiyi ihmal etmektedir.

Çalışmada denenen diğer bir yöntem Yapısal Benzerlik  $S(x,y)$  Eşitlik (5.5) ile verilmiştir:

$$S(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y) \\ = \left( \frac{2\mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \right) \left( \frac{2\sigma_x \sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \right) \left( \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3} \right) \quad (5.5)$$

Görüntü parçası  $Y$  yerine en benzer karo görüntüsü  $X$  'i bulmak için yapısal benzerlik (Structural Similarity: SSIM) denenmiştir. SSIM işaretlerin doğruluğunun hesaplanmasında yüksek etkinliğini kanıtlamış bir yöntemdir. Işık, kontrast ve yapısal benzerliği (sırasıyla ortalama, standart sapma ve kovaryans matris) için Dice benzerliğini kullanmaktadır. Yarı metriktir [64].

Ancak iki görüntüyü piksel piksel karşılaştırmak yerine ortalamalarını  $\mu$  ve standart sapma  $\sigma$  ya da kovaryans matrisi  $\Sigma$  karşılaştırdığımızda görüntü parçasındaki yapısal bilgi önemli



ölçüde kaybolmaktadır. Bu nedenle görüntünün ortalaması, standart sapması ve kovaryans matrisi hesabında yapısal bilgiyi kaybetmeden piksel piksel işlemler sonrası  $E\{\}$  beklenen değeri hesaplayan yöntemler denenmiş olsa da hesaplama ondalık sayı (float) ile yapıldığından işlemler çok uzun zaman almaktadır.

Yapılan birçok denemede, büyük bir görüntü parçasındaki yüksek yapısal bilgiye uygun mermer karo parçası bulmak için gelişmiş ve hassas benzerlik yöntemleri aramak yerine, karo boyutlarını küçülterek, dolayısıyla karo içerisindeki yapısal bilgiyi azaltarak, çok sayıda karoyu uygunluk için gözden geçirmenin daha iyi çözümler sunacağı sonucuna varılmıştır. Örneğin yazı içeren bir görüntünün mozaigi dizilmek istensin ve büyük hücrelere sahip bir ızgara ile görüntü parçalansın. Bir görüntü parçasına “A” harfi görüntüsü rast gelsin. Bu “A” harfi görüntüsüne sahip bir doğal taş arayıp bulmak, imkansız istemek olacaktır. Eğer görüntü daha küçük hücreler ile parçalansaydı, “A” harfi görüntüsü, birkaç görüntü parçasına birkaç doğru olarak dağılacaktı ve bir tane “/” ve bir tane “\” görüntüsü içeren doğal taş arama ve bulma şansı daha yüksek olacaktı. Bu nedenle çok sayıda küçük karonun benzerliğini hesaplayacak yöntem basit ve hızlı olmalıdır. Denenen çok sayıda yöntem arasından, -aslında en başından beri kullandığımız- basit, hızlı, tam sayılar ile hesaplanabilen, Öklid uzaklık özellikle esnek üretim sisteminde kullanmak için en uygun yöntemdir.

### **5.1.2 Benzer Karoyu Bulmak İçin Görüntü Özellik Tanımlayıcıları:**

Bilinen nesnelere, bilinmeyen uzaklık ve bakış açısından tanımak amacıyla ötelemeden, ölçeklemeden ve dönmeden bağımsız “yerel özellik vektörleri” hesaplayan çeşitli görüntü tanımlayıcı yöntemler vardır. Bu yöntemler, görüntüdeki noktaların tamamını incelemek yerine, sadece bazı anahtar noktaları tespit edip, kullanarak hızlı sonuç verirler (Şekil 5.2). Bu yöntemlerden bazıları, kamera ile gerçek zamanlı çalışacak kadar hızlıdır. Bazı örnek görüntü tanımlayıcıları:

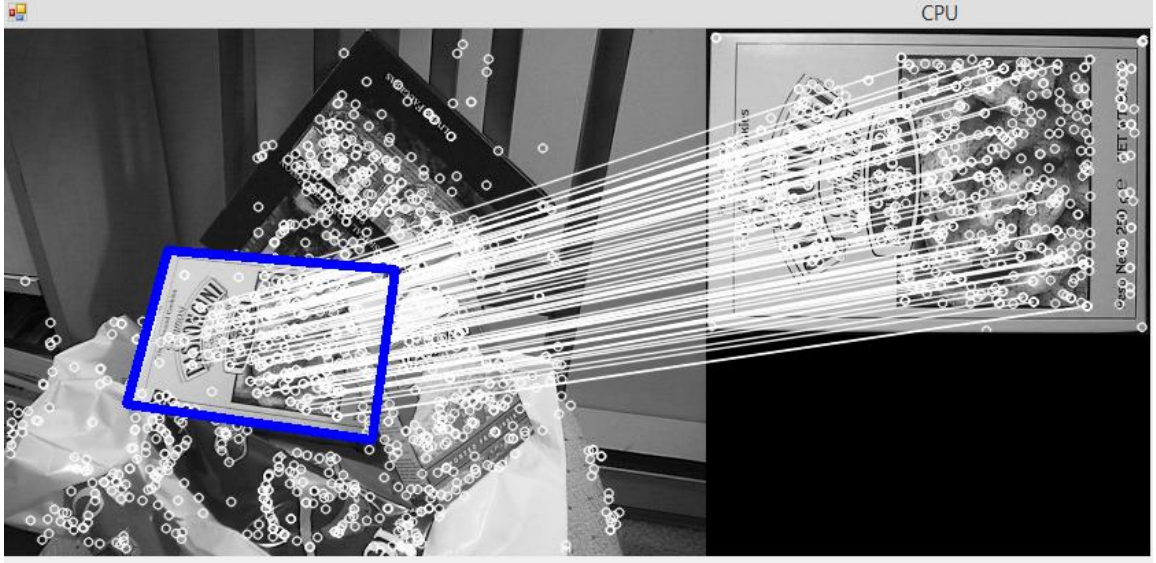
Doku Tanımlayıcıları:

HOG, LBP, Haar

Anahtar nokta (keypoint) Tanımlayıcıları:

SIFT, SURF, FREAK, ORB, GIST, AGAST, KAZE

Bu anahtar noktalar görüntünün yerel maksimum veya minimumlarıdır. Parlak veya koyu lekeler (blob) veya köşelere karşılık gelir.



**Şekil 5.2:** Emgu CV SIFT Features örnek uygulama.

Mozaiği dizilecek görüntü düzenli ızgara ile bölündükten sonra bir görüntü parçasına en uygun mermer karoyu bulmak için SIFT, SURF, KAZE, GIST ve HOG görüntü tanımlayıcıları denenmiştir ancak HOG dışında herhangi olumlu bir sonuç alınamamıştır. Bunun birinci nedeni mermer karo görüntülerinin yapısal özelliklerinin az olması nedeni ile, görüntü tanımlayıcılarının aradığı yerel maksimum/minimumlar içermemeleridir. İkinci olarak da görüntü tanımlayıcılarının, görüntüdeki değişimi hesaba katarken renk bilgisini göz ardı etmeleridir ki çalışma için renk önemli bir bilgidir.

HOG bir doku tanımlayıcısıdır. Nesne algılayan evrişimli nöron ağları için özellik çıkarmada (örneğin: araba, motosiklet, yaya, trafik tabelası tespiti vb.) çok sık kullanılmaktadır ve gradyan değişimlerin histogramını hesapladığından desen ile ilgili önemli bilgi çıkarmaktadır. Sadece bir mermer kaynağından gelen monokrom taşların kullanıldığı uygulamalarda, çok büyük mermer karo kümelerinde, HOG kullanılarak çok hızlı sonuçlar alınabilir. Renk bilgisi ile birleştirilirse görüntü parçalarına uygun karo bulmak için hızlı ve etkili bir çözüm olabilir.

Normalize edilmiş görüntü parçalarına uygulanan Öklid uzaklık yöntemi ile görüntünün yapısal özelliklerinin ön plana çıktığı ve HOG'a benzer sonuçlar elde edildiği görülmüştür. Bu nedenle HOG kullanarak çözüm aramaktan vazgeçilmiştir.

Öklid uzaklık yönteminin zayıf noktası, görüntü parçasına atanan karoların ortalama renk veya parlaklığı çok farklıysa, yapısal değişimler bu ortalama hatanın altında kaldığında maskelenmektedir.

### 5.1.3 Dizme

Macar yöntemi kullanılmaya başlanmadan önce mermer karoları, görüntü parçalarına atamak için iki çeşit sıralama stratejisi denenmiştir, bunlar:

1. İlk mozaik dizme denemelerinde, görüntü parçalarına en uygun karolar yerleştirme işlemine bilgisayarlı grafik koordinat sisteminin sıfır noktası olan sol üst köşeden başlanarak, oluşturulan mozaığın sol üst köşeden uzaklaştıkça kötüleştiği gözlenmiştir. Bunun nedeni ilk yerleştirilen karo için daha fazla seçme şansı varken, karolar yerleştirildikçe, boşta karo sayısı azaldığından, uygun karo bulma şansının da azalmasıdır. Bu soruna ilk çözüm olarak görsel algı sistemin yapısından yararlanılmaya çalışılmıştır. İnsan görsel algılama sisteminin hassaslığı, merkezden kenarlara doğru zayıflayan bir Gauss dağılımı çizer. Gözlerimizi odakladığımız noktayı en net görürken, kenarlara doğru görüntü bulanıklaşır. Dolayısıyla, bir görüntünün ilgi odağı, en çok bilgi taşıyan kısmı “görüntünün ortası” olması daha yüksek bir ihtimaldir. Görüntünün ortasını en önemli bölge olduğu kabul edilerek, görüntü parçaları görüntünün merkezine olan uzaklıkları artan sırada sıralanır ve dizmeye ilk olarak merkezdeki karodan başlanıp uzaklaşarak devam edilir. Böylelikle merkezdeki karolara uygun görüntü parçası bulma ihtimali daha yüksek olacaktır. Dizme işlemi merkezden uzaklaştıkça kötüleşecektir. Uzun bir süre bu şekilde çalışmalar sürdürülmüş, görüntü parçası ile karo arasındaki benzerliğin ölçülmesi üstünde durulmuştur. Ancak ilgi odağındaki nesnelere her zaman görüntünün merkezinde olmayabilir, örneğin insan yüzü görüntü merkezinin sağına veya soluna kaymış olabilir bu durumda yöntem başarısız olacağından terkedilmiştir.
2. Görüntüde yapısal elemanlar daha büyük renk ve ışık değişimine sahiptir. İnsan ve nesnelere görüntüdeki yerini tespit etmek için görüntü parçalarının standart sapmalarının  $\sigma$ , ortalamalarının  $\mu$  oranına yani “bağıl standart sapmaları” incelenmiştir.  $\frac{\sigma}{\mu}$  oranı yüksek olan görüntü parçalarının nesne veya yapısal eleman içerme ihtimali daha yüksektir. Görüntü parçaları bağıl standart sapması yüksek olandan düşük olana doğru azalan sırada sıralanır. Dizme işlemine bağıl standart

sapması en yüksek olandan başlanır. Bağıl standart sapma  $\frac{\sigma}{\mu}$  değerinin yüksek olduğu görüntülere “belirgin” denilir. Dizmeye belirgin karodan başlamak, ona daha fazla mermer karo seçme şansı vermek demektir. Mermer karolar yerleştirildikçe seçme şansı azalacağından, tek düze homojen görüntü parçalarını sona bırakmak daha doğru olacaktır. Yöntemin zayıf noktası, görüntüye konu olan insan veya nesnenin düşük bağıl standart sapma  $\frac{\sigma}{\mu}$  değerine sahip bölgelerinde ortaya çıkacaktır. İnsan yüzünde yanaklar  $\frac{\sigma}{\mu}$  bağıl standart sapması düşük bölgedir. Ağaç yaprakları, saç ve kürk içeren görüntü bölgeleri de yüksek  $\frac{\sigma}{\mu}$  oranına sahiptirler. Dolayısıyla,  $\frac{\sigma}{\mu}$  oranı yüksek bölgeye öncelik verildiğinde sınırlı karo stoku, bir kişinin görüntüsünü dizmek yerine, bir ağacın yapraklarını ya da bir köpeğin tüylerini dizmek için kullanılıyor olabilir ki bu da yöntemin diğer bir zayıf noktasıdır.

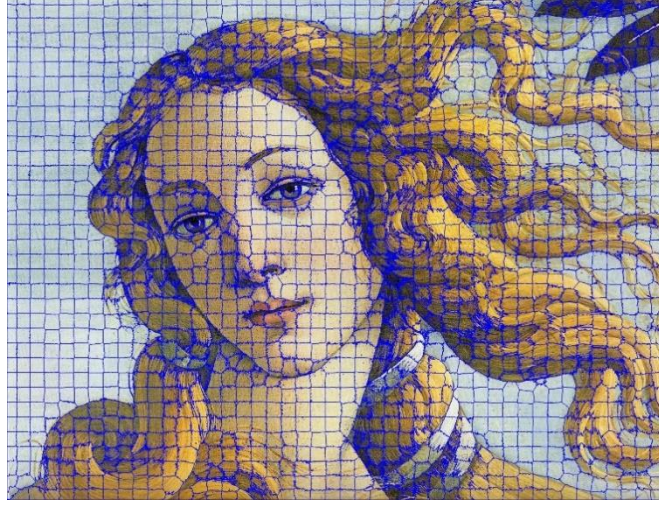
Karo atama işleminde kullanılan Macar yöntemi görüntünün tamamına atanan karoların toplam hatasını düşürerek, mozaik dizme işleminde önemli iyileştirmeler sağlamıştır. Macar yöntemi, mozaik dizme işleminde, görüntüye konu olan insan veya nesne bölgeleri ile arka plan arasında herhangi bir fark gözetmeksizin, tüm görüntü parçalarına eşit önem vererek atamalar gerçekleştirmektedir. Bu durum, çalışmada, insan beden, yüz, el ve nesne detektörleri kullanarak anahtar noktaların tespit edilmesi ve bu noktalar ile örtüşen görüntü parçalarına, arka plana göre daha az hata ile mermer karo atanması fikrinin doğmasına neden olmuştur.

#### **5.1.4 Görüntünün Süper Pikseller İle Bölütlenerek Paledyen Görüntü Parçaları Elde Edilmesi**

Mozaği oluşturulacak görüntüsü bölütlenerek, görüntü parçalarının içindeki değişimin daha durağan ve homojen olması için süper piksel algoritması denenmiştir. Superpiksel, görüntüde benzer renk ve parlaklık karakteristiği gösteren piksellerin gruplanarak aynı renk ve parlaklık verisi ile oluşturulmuş görüntünün çokgensel bölgesidir.

Superpixel SLIC (Simple Linear Iterative Clustering: Basit doğrusal iteratif kümeleme) [65] yöntemi ile bölütlenen görüntüde Superpiksel'lere en uygun karolar yerleştirilerek görüntü yeniden sentezlenebilir, ancak bu durum, görüntüdeki yapısal şekillere bağlı olarak, dizmede kullanılacak karoların paledyen olmasını gerektirmektedir. Görüntü parçaları için aranan uygun karonun belirlenmesinde renk ve ölçüye ek olarak şekil benzerliğinin de incelenmesi,

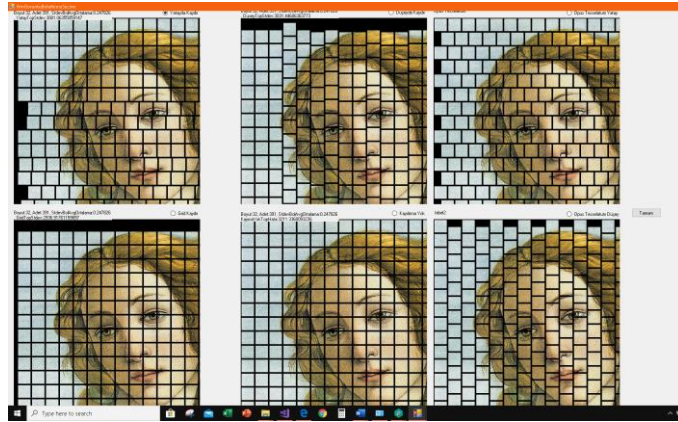
problemin çözümünü oldukça zorlaştıracaktır. SLIC yönteminin parametresi  $m$ , süperpiksellerin şeklinin, piksellerin renk ve ışık benzerliğine bağımlılığını belirler.  $m$  1 ile 40 arasında değer alır.  $m$ 'in küçük değerleri daha düzenli kare süperpikseller oluştururken,  $m$ 'in büyük değerleri piksellerin ışık ve renk değişimine daha bağımlı olarak çokgenlere neden olurlar. Şekil 5.3'de görüldüğü gibi,  $m=2$  için, kenarlardan etkilenen çokgen süper piksellerin şekline uygun karo bulmak oldukça zor olacaktır.



Şekil 5.3: Superpixel SLIC  $m=2$  için bölütleme sonucu.

### 5.1.5 Görüntüyü Parçalayan Satır ve Sütunları Kaydırarak Grid Hücreleri İçinde Standart Sapmayı Azaltmak

Desen görüntüsünü parçalayan ızgaranın hücreleri içindeki standart sapmayı azaltmak amacıyla, satırlar yada sütunlar, Şekil 5.4'de görüldüğü gibi kaydırılmıştır. Beklenen iyileşmeyi sağlanamamakla birlikte, opus tessellatum (tuğla deseni) etkisi yaratması açısından değerlendirilebilir.



Şekil 5.4: Soldan sağa; 1. Satır: yatay kaydır, dikey kaydır, opus tessellatum yatay, 2. Satır: Grid kaydır, kaydırma yok, opus tessellatum dikey.

### 5.1.6 32,16,8 piksel Karoların Bulunduğu Durumda, Desen Görüntüsünün Bölünmesi

32,16,8 piksel boyutunda mermer karoların bulunduğu durumda, desen görüntüsü 32x32 piksel hücrelere parçalanıp görüntü parçaları elde edildikten sonra, görüntü parçalarının bağıl standart sapması  $\frac{\sigma}{\mu}$  azalan sırada sıralanıp, değişimin en çok olduğu karodan başlanarak gerektiği kadarını 4'e bölünerek 16 piksel görüntü parçaları elde edilir. 16 piksel görüntü parçaları bağıl standart sapmaları  $\frac{\sigma}{\mu}$  azalan sırada sıralanıp yine değişimin fazla olduğu karodan başlanarak gerektiği kadarı 4'e bölünerek 8 piksellik görüntü parçaları elde edilir. Bölme işlemi sırasında dizilecek karo sayısı göz önünde bulundurulur. 16 ve 8 piksel boyutlarında dizilecek karo varsa adeti aşmayacak miktarda görüntü parçasına 4'e bölme işlemi uygulanır. Şekil 5.5'de değişimin fazla olduğu bölgelerde karo boyutunun küçüldüğü görülmektedir.

Yöntemde ana sorun, sadece değişime bakıldığı için renk, parlaklık ve yapısal özellikleri uyumsuz karoların atanmış olmasıdır. Bağıl standart sapmaya değerine göre karoyu 4'e bölme işlemi sonucu elde edilen görüntü parçalarının bağıl standart sapması azaltılarak dizimde kullanılacak karolara uygun hale getirilmiş olsa da uygun renk ve tonda mermer karo bulunamadığı durumda, Şekil 5.5'de görüldüğü gibi, karo içi bağıl standart sapması uyumlu ancak rengi uyumsuz karolar yerleştirmektedir.



**Şekil 5.5:** Bağıl standart sapmanın büyük olduğu bölgelerde görüntü parçasının küçültülerek değişimin azaltılması.

## 5.2 Sonular:

Sayısal bir grntnn mozaigini, en uygun mermer karolar ile dizgen, sanatsal mozaik olarak deęerlendirilebilecek, dzenli ızgara (opus regulatum) dizme yntemini kullanan, esnek retim sisteminin gerekleřtirilebilmesi iin yapılan alıřmalarda elde edilen sonular řunlardır:

- Grnt dzenli ızgara ile blnerek en uygun yapı tařları ile dizilmiřtir.
- alıřma dzlemi grntsndeki mermer karolar bltlenerek karo grntleri izole edilmiřtir.
- En uygun yapı tařlarının bulunması iin, iki grnt arasında benzerlięin ls olarak kullanılan yntem, hızlı ve etkili, klid uzaklıęı olduęu sonucuna varılmıřtır.
- Yerleřtirilen mermer karoların toplam hatası Macar yntemi ile optimize edilmiřtir.
- Farklı llerde 8, 16, 32 piksel llerinde kare karolar kullanılarak mozaik dizmek iin bir yntem sunulmuřtur.
- 32 piksel karolar, nesne gibi yapısal zellik ieren grnt paraları yerine kullanılmak iin yetersiz kalırken, arka plan grnt paraları iin daha elveriřli oldukları gzlenmiřtir.
- 16 piksel karolar, nesnelerin mozaiginin dizilmesi iin en iyi zmdr. Grnt paraları iin gerekli řartları saęlarken, hesaplama sresi aısından da kabul edilebilir sonular vermektedir.
- 8 piksel karolar, nesnelere iin en kaliteli zm sunmakla birlikte, grnty kaplamak iin binlercesi gerektięinden ok uzun hesaplama zamanına neden olmaktadırlar. Baęıl standart sapması yksek nesne grnt paralarında kullanılmak zere sınırlı sayıda projeye eklenebilirler.
- Mozaik kalitesine etki eden en nemli faktrlerden biri, karo veri kmesinin sayısıdır. Karo sayısı sadece grnty kaplayacak kadar olduęunda, karo atamalarının hatası byk olurken, karo sayısı arttıka hata azalmaktadır (řekil 4.4).
- Grntde nesnelere algılayarak, bu blgelere yapılacak karo atamalarında, nesne iermeyen blgelere nazaran ncelik veren bir yntem sunulmuřtur. El, yz, beden algılamak iin OpenPose, dięer nesnelere algılamak iin YOLO nesne detektrleri kullanılmıřtır. Arka plan grnt paralarının yerleřtirme hatası ktlterek uzaklık matrisi  $D'$  gncellenmiřtir. Yntem, karo sayısı az olduęunda faydalı olmaktadır, karo sayısı, grnt para sayısının katları olduęunda, ancak kk bir iyileřme saęlamaktadır.

- Görüntünün mozaiğini dizebilecek esnek üretim sistemi prototipi tasarlanmıştır. Robot kolun çalışma düzleminin alanı beklenenden küçük olmuştur. Delta robot ile yapılan denemelerde yüksek konumlama hatası ile karşılaşmıştır.

### 5.3 Öneriler

Robot kolun konumlama hatasının nedenleri araştırılacak ve giderilecektir. Çalışma düzlemi alanını sınırlayan etkenler de incelenip giderilecektir. Yapılacak ilk çalışma, hızlı ve hassas tut ve yerleştir işlemlerini gerçekleştiren delta robotun çalışır hale getirilmesi olacaktır.

Çalışmada, mermer karolar ile bir görüntünün mozaiğini dizmenin başarımını arttırabilmek için çeşitli şekillerde karolar, paledyen karolar problemin dışında tutulmuştur. Parlaklık, renk ve doku özelliklerine ek olarak, şekil özelliği eklenebilir, çeşitli şekillerde karolar ya da paledyen karolar kullanarak nesnelerin en uygun yapıtaşları ile dizilmesi çalışılabilir.

Sanatsal mozaiklerde düzenli ızgara (opus regulatum) yerine diğer karo yerleştirme teknikleri örneğin opus vermiculatum çalışılabilir.

Uzaklık matrisinden, üyelik matrisine geçilerek, en yüksek üyelik değerine sahip mermer karoları bulmak için problem, maksimum akış problemi olarak ele alınabilir ve örneğin Ford-Fulkerson Algoritması kullanılarak en uygun yapıtaşları aranabilir.

Nesnelerin en uygun yapıtaşları ile dizilmesi problemi üç boyuta taşınabilir.



## 6. KAYNAKLAR

- [1] “Tessellation.”, Wikipedia. <https://tr.wikipedia.org/wiki/Tesselasyon>. (Eriřim tarihi: 8 Ağustos 2020).
- [2] MEB, MEGEP. Ankara (2015). *Mozaik Analizleri*. [Online] Eriřim adresi: [http://megep.meb.gov.tr/mte\\_program\\_modul/moduller\\_pdf/Mozaik%20Analizleri.pdf](http://megep.meb.gov.tr/mte_program_modul/moduller_pdf/Mozaik%20Analizleri.pdf) . (Eriřim tarihi: 5 Kasım 2020).
- [3] G. Saruřık, A. Saruřık, A. řentürk, “Mermer Mozaik Teknikleri ve Uygulama řekilleri,” *I. Burdur Sempozyumu*, Burdur, Mehmet Akif Ersoy Üniversitesi, 2005, s. 875-888
- [4] M.Y. Çelik ve A. Baęcı, “Dekoratif Mermer Mozaik Yapım Teknikleri,” *Türkiye IV Mermer Sempozyumu*, Afyon, 2003, s. 399-414.
- [5] E.P. İnal, “Robot ile Mozaik Dizme Otomasyonu”, Yüksek Lisans Tezi, Balıkesir Üniversitesi, Balıkesir, 2006.
- [6] MEB, MEGEP. Ankara (2011). *Sanayi Mozaięi Montajı*, [Online] Eriřim adresi: [http://www.megep.meb.gov.tr/mte\\_program\\_modul/moduller\\_pdf/Sanayi%20Mozai%20C4%9Fi%20Montaj%20C4%B1.pdf](http://www.megep.meb.gov.tr/mte_program_modul/moduller_pdf/Sanayi%20Mozai%20C4%9Fi%20Montaj%20C4%B1.pdf) .(Eriřim tarihi: 5 Kasım 2020).
- [7] “Cutting and finishing mosaic production.”, Tecnema Breton Group, [Online] Eriřim adresi: <http://www.gvservice.com/cutting-finishing-mosaic-production/tecnema-technology/>. (Eriřim tarihi: 5 Kasım 2020).
- [8] R. Navon, “Process and quality control with a video camera, for a floor-tilling robot”, *Autom. Constr.* vol. 10, no. 1, pp. 113-125, 2000.
- [9] A. Oral, F. Erzincanlı; “Computer-assisted robotic tiling of mosaics”, *Robotica*, vol. 22, no. 02, pp. 235-239, 2004.
- [10] B. Kaya, A. Berkay and F. Erzincanlı, “Robot assisted tiling of glass mosaic with image processing”, *Indus. Rob.* vol. 32, no. 5, pp. 338– 392, 2005.
- [11] A. Oral A., E.P. İnal, “Marble mosaic tiling automation with a four degrees of freedom cartesian robot”, *RCIM*, vol. 25, no. 3, pp. 589-596, 2009.
- [12] A. Oral, “Patterning automation of square mosaics using computer assisted SCARA robot”, *Robotica*, vol. 27, no. 06, pp. 897-903, 2009.
- [13] İ. Çayıröęlü, B.E. Demir, “Computer assisted glass mosaic tiling automation”, *RCIM*, vol. 28, no 5, pp. 583–591, 2012.

- [14] Z. Falomir, L.M. Cabedo, L.G. Abril, F.V. Morente, “Measures of similarity between qualitative descriptions of shape, colour and size applied to mosaic assembling”, *J. Vis. Commun. Image Represent.* vol. 24, no. 3, pp. 388-396, 2013.
- [15] N. King, M. Bechthold, A. Kane, and P. Michalatos, “Robotic tile placement: Tools, techniques and feasibility” *Autom. Constr.*, vol. 39, pp. 161 – 166, 2014.
- [16] D. H. Jung, J. Park and M. Schwartz, “Towards on-site autonomous robotic floor tiling of mosaics,” *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, 2014, pp. 59-63, doi: 10.1109/ICCAS.2014.6987959.
- [17] İ. S. Üncü, O. Kilim, Y. Kubuz, A. Uysun, “Görüntü İşleme Metoduyla Mermer Mozaik Otomasyonu,” *2. Uluslararası Mühendislik ve Bilim Alanında Yenilikçi Teknolojiler Sempozyumu*, KARABÜK, 18-20 Haziran 2014, s. 891-899.
- [18] O. Önal. “Breşli kayalar kullanılarak fotogerçekçi mozaik üretimi için hesaba dayalı bir metodoloji”, *PAJES*, cilt 22, no 3, 136-141, 2016.
- [19] P. Nuntaporn, K. Pisut, “Development of tiling automation for custom mosaic design”, *RCIM*, vol. 35, p. 55. 2015.
- [20] P. Panitcha, K. Pisut. “Improving decision making in product flow-based tiling automation for custom mosaic design”, *Assembly Autom.*, vol. 37, Issue. 4, p. 391, 2017.
- [21] X. Li, C. Sun, W. Cheng, X. Jiang and Y. Liu, "Adaptive Vision-Based Control for Robotic Tiling with Uncalibrated Cameras and Limited FOV," *2019 IEEE 15th International Conference on Control and Automation*, Edinburgh, United Kingdom, 2019, pp.168-173, 2019.
- [22] S. Battiato, G. Di Blasi, G.M. Farinella, G. Gallo, “A Survey of Digital Mosaic Techniques,” *Fourth Eurographics Italian Chapter Conference*, University of Catania, Italy, 2006.
- [23] G. Di Blasi and M. Petralia, “Fast photomosaic,” Poster proc. *The 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, Plzen, Czech Republic, Jan. 31 – Feb. 4 2005, pp. 15-16.
- [24] D. D'Souza, D. Ciesielski, M. Berry and K. Trist, "Generation of self-referential animated photomosaics," *Proc. of 15th Int. Conf. on Multimedia*, 2007, pp. 489-492.
- [25] H. Narasimhan and S. Satheesh, "A randomized iterative improvement algorithm for photomosaic generation," *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, India, 2009, pp. 777-781, doi: 10.1109/NABIC.2009.5393882.

- [26] M. Mikamo, M. Slomp, S. Yanase, B. Raytchev, T. Tamaki and K. Kaneda, "Maximizing Image Utilization in Photomosaics," *2010 First International Conference on Networking and Computing*, Higashi, Japan, 2010, pp. 275-278, doi: 10.1109/IC-NC.2010.17.
- [27] Y. Yang, Y. Ito and K. Nakano, "Photomosaic Generation by Rearranging Subimages, with GPU Acceleration," *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, Lake Buena Vista, FL, 2017, pp. 942-951, doi: 10.1109/IPDPSW.2017.55.
- [28] Zhang, H., Gao, X., Du, J. et al. "Tile selection method based on error minimization for photomosaic image creation." *Front. Comput. Sci.* 15, 153702 (2021). <https://doi.org/10.1007/s11704-020-9242-6>
- [29] G. Silahtaroglu, *Veri Madenciliği*, 1. Baskı. İstanbul: Papatya Yayıncılık, 2008.
- [30] CIE LAB renk uzayı, Wikipedi Özgür Ansiklopedi [Online] Erişim adresi: [https://en.wikipedia.org/wiki/CIELAB\\_color\\_space](https://en.wikipedia.org/wiki/CIELAB_color_space). Erişim tarihi 25 Temmuz 2021.
- [31] R. Özdemir, "Üniversite Ders Zaman Çizelgeleme Problemi", Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, İstanbul, 2012
- [32] V. V. Nabiyev, *Algoritmalar:Teoriden Uygulamalara*, 4. Baskı. Ankara: Seçkin Yayıncılık, Haziran 2013.
- [33] "Hungarian Algorithm.", Wikipedia. [https://en.wikipedia.org/wiki/Hungarian\\_algorithm](https://en.wikipedia.org/wiki/Hungarian_algorithm). (Erişim tarihi: 5 Kasım 2020).
- [34] "Jonker-Volgenant Algorithm for linear assignment problem.", MathWorks.com. <https://www.mathworks.com/matlabcentral/fileexchange/26836-lapjv-jonker-volgenant-algorithm-for-linear-assignment-problem-v3-0>. (Erişim tarihi: 29 Temmuz 2021).
- [35] "Assignment problem and Hungarian Algorithm.", TopCoder.com. <https://www.topcoder.com/thrive/articles/Assignment%20Problem%20and%20Hungarian%20Algorithm> . (Erişim tarihi: 5 Nisan 2021).
- [36] "Hungarian Method.", Technische Universität München. [https://algorithms.discrete.ma.tum.de/graph-algorithms/matchings-hungarian-method/index\\_en.html](https://algorithms.discrete.ma.tum.de/graph-algorithms/matchings-hungarian-method/index_en.html). (Erişim tarihi: 11 Ağustos 2021)
- [37] H. Cui, J. Zhang, C. Cui, Q. Chen, "Solving large-scale assignment problems by Kuhn-Munkres algorithm," *2nd International Conference on Advances in Mechanical Engineering and Industrial Informatics*, 2016, pp. 822-825.

- [38] Z. Wang and A. C. Bovik. "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures", *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98-117, Jan. 2009, doi: 10.1109/MSP.2008.930649.
- [39] A. Amidi ve S. Amidi, "C230 Derin Öğrenme Ders Notu", *Evrışimli Sinir Ağları El kitabı*. [Online] Erişim adresi: <https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-convolutional-neural-networks#object-detection>. Erişim tarihi: 5 Kasım 2020
- [40] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
- [41] "Read YOLO V3 in a text", [Online] Erişim adresi: <http://www.programmingsought.com/article/7523856979/>. Erişim tarihi: 5 Kasım 2020
- [42] B. KILIÇ, "Panorama ile Üretilen Plevral Efüzyon Sitopatoloji Görüntülerinde YOLO V3 ile Otomatik Çekirdek Algılama", Yüksek Lisans Tezi, Karadeniz Teknik Üniversitesi, Trabzon 2020
- [43] Dlib C++ library, [Online] Erişim Adresi: <http://dlib.net/> Erişim Tarihi: 20 Haziran 2021
- [44] "Facial landmarks with dlib, OpenCV, and Python" [Online] Erişim adresi: <https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>. Erişim tarihi 5 Kasım 2020.
- [45] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1867-1874, doi: 10.1109/CVPR.2014.241.
- [46] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields", *IEEE PAMI*, vol. 43, no. 1, pp. 172-186, 1 Jan. 2021.
- [47] "Derin öğrenme temelli insan pozü kestirimi öğreticisi" [Online] Erişim adresi: <https://www.learnopencv.com/deep-learning-based-human-pose-estimation-using-opencv-cpp-python/>. Erişim tarihi: 5 Kasım 2020
- [48] CMU Algısal Hesaplama Laboratuvarının OpenPose projesi Github sayfası, [Online] Erişim adresi: <https://github.com/CMU-Perceptual-Computing-Lab/openpose>. Erişim Tarihi 5 Kasım 2020

- [49] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd Edition. Prentice-Hall, Inc., USA, 1992.
- [50] İ. Çelebi, “Endüstriyel bir Delta Robotun Geri Adımlama Tekniği ile Yörünge İzleme Kontrolü”, Yüksek Lisans Tezi, Gebze Teknik Üniversitesi, Gebze 2019
- [51] P. J. Zsombor-Murray, “Descriptive Geometric Kinematic Analysis of Clavel's, Delta robot.”, Dept of Mechanical Engineering , McGill University, Canada 2004. <http://www.robotics.caltech.edu/~jwb/courses/ME115/handouts/DescriptiveGeometryDeltaRobot.pdf>. (Erişim tarihi: 20 Haziran 2021)
- [52] Z. Bingül, S. Küçük, *Robot Kinematiği*. İstanbul: Birsen Yayın Evi, 2009
- [53] J. M. García, JMG Fecha, *Inverse – Forward Kinematics of Delta Robot*, 2010
- [54] Neugart Redüktör San. Tic. Ltd. Sti. İstanbul, Tr. *Ekonomi serisi planet redüktör Manuel*, Erişim tarihi: 20 Haziran 2021. [Online]. Erişim adresi: <https://www.neugart.com/tr-tr/ueruenler/cikis-milli-planet-reduektoerleri/plpe#PLPE70>.
- [55] “Step Motor verileri.” <http://www.robosan.com.tr/step/sy42sth47.htm>. (Erişim tarihi: 20 Haziran 2021).
- [56] “ZM-2H504 İki Faz Step Motor Sürücü Verileri.” <http://www.robosan.com.tr/pdfs/ZM-2H504%20TURKCE.pdf>. (Erişim tarihi: 20 Haziran 2021)
- [57] Siemens Global. *Siemens S7-1200 System Manual*. (06/2015 A5E02486680-AJ). Erişim tarihi: 20 Haziran 2021. [Online]. Erişim adresi: [https://support.industry.siemens.com/cs/attachments/109478121/s71200\\_system\\_manual\\_en-US\\_en-US.pdf?download=true](https://support.industry.siemens.com/cs/attachments/109478121/s71200_system_manual_en-US_en-US.pdf?download=true).
- [58] “Venüs'ün Doğuşu (Botticelli).”, Wikipedia. [https://tr.wikipedia.org/wiki/Ven%C3%BCs%27%C3%BCn\\_Do%C4%9Fu%C5%9Fu\\_\(Botticelli\)](https://tr.wikipedia.org/wiki/Ven%C3%BCs%27%C3%BCn_Do%C4%9Fu%C5%9Fu_(Botticelli)). (Erişim Tarihi: 26 Haziran 2021).
- [59] “Atatürk portresi 1040x1600 piksel.”, Wikimedia Commons. [https://upload.wikimedia.org/wikipedia/commons/4/4a/Mustafa\\_Kemal\\_Atat%C3%B5Crk.jpg](https://upload.wikimedia.org/wikipedia/commons/4/4a/Mustafa_Kemal_Atat%C3%B5Crk.jpg). (Erişim Tarihi: 25 Temmuz 2021).
- [60] “Atatürk fotoğrafı 1120x1600 piksel” [Online] Erişim adresi: <https://pbs.twimg.com/media/DAL86-hVYAEDXJq.jpg>. (Erişim tarihi: 26 Haziran 2021).

- [61] “Genelkurmay'dan renkli Atatürk fotoğrafları.”, NTV Haber Sitesi. <https://www.ntv.com.tr/galeri/turkiye/genelkurmaydan-renkli-ataturk-fotograflari,Tk1ZleUbiUmmJIQqM34hSA>. (Erişim tarihi: 26 Haziran 2021).
- [62] Atatürk Portresi [Online] Erişim adresi: <https://i.pinimg.com/originals/99/58/00/9958001a356e2ccdefc237a558e024f1.jpg>. (Erişim tarihi: 26.Haziran 2021).
- [63] J. Vergés-Llahí, A. Sanfeliu, “Evaluation of Distances Between Color Image Segmentations.”, In: Marques J.S., Pérez de la Blanca N., Pina P. (eds) Pattern Recognition and Image Analysis. IbPRIA 2005. LNCS, vol 3523. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/11492542\\_33](https://doi.org/10.1007/11492542_33)
- [64] Zhou, W., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity." *IEEE Trans. Image Process.* Vol. 13, Issue 4, April 2004, pp. 600–612.
- [65] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," in *IEEE PAMI*, vol. 34, no. 11, pp. 2274-2282, Nov. 2012, doi: 10.1109/TPAMI.2012.120.

## ÖZGEÇMİŞ

### Kişisel Bilgiler

Adı Soyadı : Alp İMREK  
Doğum tarihi ve yeri : 10 Haziran 1971, İzmir  
e-posta : alpimrek@balikesir.edu.tr

### Öğrenim Bilgileri

Derece	Okul/Program	Yıl
Yüksek Lisans	Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Anabilim Dalı	1993–1997
Lisans	Uludağ Üniversitesi, Mühendislik Fakültesi, Elektronik Mühendisliği Bölümü	1988–1993
Lise	Balıkesir Lisesi	1985–1988

### Yayın Listesi:

#### Uluslararası bilimsel toplantılarda sunulan ve bildiri kitaplarında basılan bildiriler:

- [1] A. İmrek, A. Oral, “A software for pick and place robot movements from marble mosaic pattern configuration designed by CAD, 17. Uluslararası Makina Tasarım ve İmalat Kongresi, 2016, (Tam Metin Bildiri/Sözlü Sunum)(Yayın No:3626296) [Tezden Türetilmiştir].