



**DERİN ÖĞRENME TABANLI  
YAPI ELEKTRİK PLAN ÇİZİMİ**

**Bayram AKGÜL**

**2022  
DOKTORA TEZİ  
BİLGİSAYAR MÜHENDİSLİĞİ**

**Tez Danışmanı  
Doç. Dr. Hakan KUTUCU**

# DERİN ÖĞRENME TABANLI YAPI ELEKTRİK PLAN ÇİZİMİ

**Bayram AKGÜL**

**T.C.  
Karabük Üniversitesi  
Lisansüstü Eğitim Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalında  
Doktora Tezi  
Olarak Hazırlanmıştır**

**Tez Danışmanı  
Doç. Dr. Hakan KUTUCU**

**KARABÜK  
Mayıs 2022**

Bayram AKGÜL tarafından hazırlanan “DERİN ÖĞRENME TABANLI YAPI ELEKTRİK PLAN ÇİZİMİ” başlıklı bu tezin Doktora Tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Hakan KUTUCU .....

Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından Oy Birliği ile Bilgisayar Mühendisliği Anabilim Dalında Doktora tezi olarak kabul edilmiştir. 27/05/2022

Ünvanı, Adı SOYADI (Kurumu)

İmzası

Başkan : Prof. Dr. Ergin YILMAZ (BEÜ) .....

Üye : Doç. Dr. Hakan KUTUCU (KBÜ) .....

Üye : Doç. Dr. Sedat AKLEYLEK (OMÜ) .....

Üye : Dr. Öğr. Üyesi Yüksel ÇELİK (KBÜ) .....

Üye : Dr. Öğr. Üyesi Hüseyin ALTINKAYA (KBÜ) .....

KBÜ Lisansüstü Eğitim Enstitüsü Yönetim Kurulu, bu tez ile, Doktora derecesini onamıştır.

Prof. Dr. Hasan SOLMAZ .....

Lisansüstü Eğitim Enstitüsü Müdürü



*“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”*

Bayram AKGÜL

## **ÖZET**

**Doktora Tezi**

### **DERİN ÖĞRENME TABANLI YAPI ELEKTRİK PLAN ÇİZİMİ**

**Bayram AKGÜL**

**Karabük Üniversitesi**

**Lisansüstü Eğitim Enstitüsü**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Tez Danışmanı:**

**Doç. Dr. Hakan KUTUCU**

**Mayıs 2022, 126 sayfa**

Bu tez çalışması, konut binalarında derin öğrenme kullanarak elektrik iç tesisat plan çizimlerinin otomatik oluşturulması ile ilgilidir. Sistem, mimari kat planlarındaki tefriş (dekoratif) sembollerinin tespiti ve sınıflandırılmasına dayalı olarak mekânsal ilişkilerin oluşturulması esasına dayanmaktadır.

Elektrik tesisat plan çizimi, bir binanın elektrik ihtiyacını sağlamak için uygulanması gereken kurulum projesinin şematik olarak gösterilmesidir. Çizgilerden ve sembollerden oluşan elektrik plan çizimleri, 2D CAD programları kullanılarak mimari kat planları üzerinde hazırlanmaktadır. Elektrik iç tesisat plan çizimlerinin ilk adımı, kat planları üzerinde uygun yerlere priz, ışık, anahtar gibi elektrik malzeme sembollerinin yerleştirilmesidir. İkinci adım ise mekânsal ilişkiler dikkate alınarak konulmuş bu elektrik sembollerinin bağlantı şemalarının oluşturulmasıdır. Elektrik iç tesisat plan çizimleri, zaman ve dikkat gerektiren işlemdir.

Bu çalışma kapsamında, konut binalarında elektrik iç tesisat planları çizilirken kat planları üzerinde uygun yerlere uygun elektrik malzeme sembollerinin otomatik olarak yerleştirilmesi için tefriş tabanlı bir sistem geliştirilmiştir. Mimari projelerde tefrişler buzdolabı, televizyon, ocak-fırın, çamaşır makinesi, bulaşık makinesi; koltuklar, masalar, yataklar, dolaplar, kapılar ve pencereler gibi çizim sembollerine verilen genel isimlerdir. Tefriş sembolleri, çizilen mimari kat planlarında oluşturulan odaların tipini ve kullanılması gereken elektrik malzemesini göstermesi bakımından önemlidir. Çalışma kapsamında mobilya, beyaz eşya, sıhhi tesisat, kapı ve pencere gibi kat planlarında en sık kullanılan malzemelerin sembollerini içeren düzinelerce mimari kat planı çizimlerinden bir *tefriş sembolleri veri seti* oluşturulmuştur. Tefriş sembollerini sınıflandırmak için derin evrişimli bir sinir ağı olan inception-v3 modeli transfer öğrenme ile kullanılmıştır. Oluşturulan model, 20 gerçek kat planı üzerinde test edilerek tefriş sembol sınıflandırmasında %97,05 gibi tatmin edici bir doğruluk elde edilmiştir.

Yapılan çalışma ile kat planlarında tefriş sembolleri tespit edilerek sınıflandırılmış ve bu tefrişlere uygun elektrik malzeme sembolleri olması gereken yerlere otomatik olarak yerleştirilmiştir. Daha sonra sınıflandırılan tefriş sembollerine göre mahal tipleri belirlenmiş ve mahaller arası mekânsal ilişkiler oluşturulmuştur. Son adımda mekânsal ilişkiler göz önüne alınarak elektrik bağlantı şemaları otomatik olarak gerçekleştirilmiştir. Böylece elektrik tesisat plan çizimi, otomatik hale getirilerek zamandan ve işçilikten kazanç sağlanmıştır.

**Anahtar Sözcükler :** Derin öğrenme, transfer öğrenme, bilgisayar destekli tasarım, elektrik plan çizimi, otomatik tasarım.

**Bilim Kodu** : 92431

## **ABSTRACT**

**Ph. D. Thesis**

### **DEEP LEARNING BASED BUILDING ELECTRICAL PLAN DRAWING**

**Bayram AKGÜL**

**Karabük University  
Institute of Graduate Programs  
Department of Computer Engineering**

**Thesis Advisor:**

**Assoc. Prof. Dr. Hakan KUTUCU**

**May 2022, 126 pages**

This thesis is related with the automatic generation of electrical plan drawings in residential buildings by using deep learning. The system is based on the creation of spatial relations through the detection and classification of the furnishing (decorative) symbols in architectural floor plans.

Electrical plan drawings are schematic displays of an installation project that must be applied to provide the electricity requirement for construction. Electrical plan drawings consisting of lines and symbols are prepared on architectural floor plans using 2D CAD programs. The placement of electrical power symbols—such as sockets, lights, and switches—is the first step of an electrical plan drawing. The second step is to create the connection diagrams of these electrical power symbols based on spatial relations. Electrical plan drawings are a process that requires time and attention.

With this study, a furnishing-based system has been developed to automatically draw the appropriate electrical power symbols on the floor plans while drawing the electrical plan drawings in residential buildings. Furnishing in architectural plans is a general term for drawing symbols representing all furniture, electrical appliances, plumbing fixtures, windows, and doors. Furnishing symbols are important in terms of showing the type of rooms created in architectural floor plans and the electrical materials that should be used. We have created a *furnishing symbol dataset* drawing on dozens of architectural plan drawings that contain symbols of the most commonly used tools in floor plans, such as furniture, appliances, plumbing, doors, and windows. We used the Inception-v3 model, which is a deep convolutional neural network, with transfer learning to classify furnishing symbols. We tested the model on 20 real floor plans and achieved a very satisfactory accuracy of 97.05% in furnishing symbol classification.

With this study, furnishing symbols were detected and classified on the floor plans and then electrical power symbols complying with these symbols were drawn automatically where they should be. Then room types were determined according to classified furnishing symbols and spatial relations between the rooms were established. As a final step, the electrical wiring diagrams were drawn automatically considering the spatial relations. Thus, the electrical plan drawing has been automated, saving time and labor.

**Key Word** : Machine learning, transfer learning, computer aided design, electrical plan drawing, automated design.

**Science Code** : 92431

## TEŐEKKÜR

Bu tez alıőmasının planlanmasında, araőtırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteęini esirgemeyen, engin bilgi ve tecrübelerinden yararlandıęım, yönlendirme ve bilgilendirmeleriyle alıőmamı bilimsel temeller ışığında őekillendiren sayın hocam Do. Dr. Hakan KUTUCU'ya sonsuz teőekkürlerimi sunarım. alıőma boyunca tezin őekillenmesinde yardımlarını esirgemeyen Dr. Öğr. Üyesi Yüksel ELİK ve Dr. Öğr. Üyesi Hüseyin ALTINKAYA hocalarıma, tez savunmasında önerileri ile katkıda bulunan Prof. Dr. Ergin YILMAZ ve Do. Dr. Sedat AKLAYLEK hocalarıma teőekkür ederim.

alıőma boyunca teknik ve maddi desteklerini esirgemeyen, alıőmayı sahada test eden Hikmet DİNÇTÜRK, Din DİNÇTÜRK ve Bartın Din Mühendislik A.Ő. alıőanlarına; tezi dilbilgisi ve imla kuralları yönünden inceleyerek yardımlarını esirgemeyen Öğr. Gör. Hasan AKGÜL'e teőekkürlerimi sunarım.

Sevgili aileme manevi hiçbir yardımını esirgemedен yanımda oldukları için tüm kalbimle teőekkür ederim.

## İÇİNDEKİLER

	<u>Sayfa</u>
KABUL .....	ii
ÖZET.....	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER .....	ix
ŞEKİLLER DİZİNİ.....	xii
ÇİZELGELER DİZİNİ .....	xv
SİMGELER VE KISALTMALAR DİZİNİ .....	xvi
SİMGELER .....	xvi
KISALTMALAR .....	xvii
BÖLÜM 1 .....	1
GİRİŞ .....	1
1.1. ÇALIŞMANIN AMACI.....	1
1.2. LİTERATÜR ÖZETİ.....	5
1.3. ÇALIŞMANIN LİTERATÜRE KATKILARI.....	11
BÖLÜM 2 .....	12
BİLGİSAYAR DESTEKLİ TASARIM .....	12
2.1. VEKTÖREL ÇİZİM .....	13
2.2. VEKTÖREL İŞLEMLER.....	15
BÖLÜM 3 .....	18
ELEKTRİK TESİSAT PLANLARI .....	18
3.1. ELEKTRİK MALZEMELERİ .....	19
3.1.1. Priz ve Motor Tesisatında Kullanılan Malzemeler .....	20
3.1.2. Aydınlatma Tesisatında Kullanılan Malzemeler.....	20
3.2. HATLAR (BAĞLANTILAR) .....	21
3.3. HESAPLAMALAR VE DÖKÜMLER.....	21

	<b><u>Sayfa</u></b>
BÖLÜM 4 .....	23
MAKİNE ÖĞRENMESİ .....	23
4.1. YAPAY SİNİR AĞLARI .....	24
4.2. DERİN ÖĞRENME .....	28
4.2.1. Evrimsel Sinir Ağları .....	30
4.2.1.1. ESA Katmanları .....	31
4.2.1.2. ESA Parametreleri .....	34
4.2.1.3. Eğitim ve Test Veri Setlerinin Ayarlanması .....	38
4.2.2. Sınıflandırma Ölçüm Metrikleri .....	39
4.2.2.1. Doğruluk Oranı .....	39
4.2.2.2. Karışıklık Matrisi .....	40
4.2.2.3. F1-puanı .....	41
4.2.3. Öğrenme Kontrolü .....	43
4.3. TRANSFER ÖĞRENME .....	45
4.4. NESNE ALGILAMA VE SINIFLANDIRMA .....	48
BÖLÜM 5 .....	50
DERİN ÖĞRENME İLE TEFRİŞ SINIFLANDIRMA .....	50
5.1. VERİ SETİ OLUŞTURMA .....	51
5.1.1. Veri Seti Çoğaltma ve Çeşitlendirme .....	54
5.2. TEFRİŞ SINIFLANDIRMA İÇİN ÖNERİLEN METOD .....	55
5.2.1. Modelin Oluşturulması ve Eğitim .....	56
5.2.2. Deneysel Sonuçlar .....	57
5.3. GERÇEK KAT PLANLARINDA TEST .....	63
BÖLÜM 6 .....	65
ELEKTRİK PLAN ÇİZİMİNİN OTOMATİK OLUŞTURULMASI .....	65
6.1. ELEKTRİK MALZEME SEMBOLÜ YERLEŞTİRME .....	66
6.1.1. Mahal Bölgesi Seçimi .....	67
6.1.2. Tefriş Sembollerinin Bulunması .....	69
6.1.3. Tefriş Sembollerinin Sınıflandırılması .....	69
6.1.4. Tefriş Sembolüne Göre Elektrik Malzemesi Önerme .....	70
6.1.5. Elektrik Malzeme Sembolü Konumlandırma .....	72

	<b><u>Sayfa</u></b>
6.2. MAHAL TİPİNE GÖRE AYDINLATMA TASARIMI.....	74
6.2.1. Tefriş Sembollerinden Mahal Tipi Belirleme .....	74
6.2.2. Aydınlatma Hesabı.....	77
6.2.3. Aydınlatma Tasarımı.....	83
6.2.4. Aydınlatma Armatürü Seçimi .....	84
6.2.5. Aydınlatma Hesabının Yapılacağı Noktaların Belirlenmesi.....	84
6.2.6. Armatür Adedi Öneri Listesi Oluşturma.....	85
6.3. BAĞLANTI HATLARININ OLUŞTURULMASI .....	89
6.3.1. Ön İşlemler.....	89
6.3.2. Graf ve Ağaç Veri Yapıları .....	91
6.3.3. Kat Planı Modelleme.....	94
6.3.4. Elektrik Dağıtım Planının Modellenmesi.....	97
6.3.5. Elektrik Dağıtım Ağacının Oluşturulması.....	102
6.3.5.1. Aydınlatma Tesisatı Çizimi .....	103
6.3.5.2. Priz Tesisatı Çizimi.....	105
BÖLÜM 7 .....	109
SONUÇ VE ÖNERİLER .....	109
7.1. SONUÇLAR.....	109
7.2. KISITLAR .....	112
7.3. ÖNERİLER VE SONRAKİ ÇALIŞMALAR .....	112
KAYNAKLAR .....	114
ÖZGEÇMİŞ .....	126

## ŞEKİLLER DİZİNİ

### Sayfa

Şekil 1.1. Tefriş sembolleri içeren örnek bir daire kat planı.....	3
Şekil 1.2. Veri setindeki tefriş sembollerinden örnekler.....	10
Şekil 2.1. Blok kullanımı. ....	14
Şekil 2.2. CCW yönteminin gösterimi [51]. ....	16
Şekil 2.3. İki doğru parçasının kesişimi. ....	16
Şekil 2.4. Polyline nesnesinin paralelini oluşturma. ....	17
Şekil 3.1. Kuvvetli akım elektrik malzemeleri. ....	19
Şekil 4.1. Yapay zeka, makine öğrenmesi ve derin öğrenme arasındaki ilişki.....	23
Şekil 4.2. Genel hatlarıyla makine öğrenmesi yöntemleri. ....	24
Şekil 4.3. Geleneksel yapay sinir ağı ve bir nöron yapısı. ....	25
Şekil 4.4. Sigmoid ve Tanh fonksiyonları.....	27
Şekil 4.5. Geleneksel makine öğrenmesi ve derin öğrenme. ....	28
Şekil 4.6. Evrimsel yapay sinir ağı yapısı. ....	31
Şekil 4.7. Örnek konvolüsyon işlemi. ....	32
Şekil 4.8. $2 \times 2$ filtre 2 adım kaydırma ile max-pooling. ....	33
Şekil 4.9. Standart ve dropout uygulanmış sinir ağı [92].....	35
Şekil 4.10. Optimizasyon algoritmalarının karşılaştırılması [95].....	36
Şekil 4.11. Eğitimin erken bitirilmesi [99]. ....	37
Şekil 4.12 Transfer öğrenme diyagram gösterimi.....	45
Şekil 4.13. Orijinal GoogleNet mimarisi [120].....	47
Şekil 4.14. Inception-v3 şematik modeli [121].....	47
Şekil 4.15. Orijinal ve yeniden şekillendirilmiş inception modülleri. ....	48
Şekil 5.1. Örnek tefriş sembolleri. ....	50
Şekil 5.2. Mimarlar tarafından kullanılabilen farklı ocak-fırın sembolleri.....	51
Şekil 5.3. Vektörel tefriş sembolü pikselleştirme. ....	53
Şekil 5.4. Farklı ölçeklerde bitmap yapılmış ocak-fırın sembolü. ....	54
Şekil 5.5. Bir sembolün döndürülmüş ve yansıtılmış halleri. ....	55
Şekil 5.6. Tefriş sembolleri sınıflandırması için önerilen metod. ....	56
Şekil 5.7. Eğitim ve doğrulama veri setlerindeki doğruluk oranları.....	58

Şekil 5.8. Eğitim ve doğrulama veri setlerindeki kayıp oranları.....	58
Şekil 5.9. Test veri setinde doğruluk oranları. ....	58
Şekil 5.10. FDS_Full veri seti için hazırlanmış sınıflandırıcının eğitim ve doğrulama veri setleri üzerindeki doğruluk ve kayıp oranları.....	59
Şekil 5.11. Örnek bir daire kat planında tefriş sınıflandırma .....	64
Şekil 6.1. İki nokta ile mahal çokgeni oluşturma. ....	67
Şekil 6.2. Tek nokta ile mahal çokgeni oluşturma. ....	68
Şekil 6.3. Sınırlarını belirleyerek mahal çokgeni oluşturma.....	68
Şekil 6.4 Bloklardan görüntü oluşturma ve sınıflandırma. ....	70
Şekil 6.5. Bazı elektrik malzeme sembolleri.....	72
Şekil 6.6. Uygun duvara elektrik sembolünün otomatik konulması. ....	73
Şekil 6.7. Elektrik malzemesi sembollerinin, tanımlanmış tefriş sembollerine göre otomatik yerleştirilmesi. ....	73
Şekil 6.8. Tefriş puanlama ile mahal tipi tespiti. ....	76
Şekil 6.9. Eulumdat dosyası içeriği.....	77
Şekil 6.10. Bir aydınlatma armatürünün 2D ve 3D ortamda ışık dağılım eğrileri. ....	78
Şekil 6.11. A noktasındaki armatür ile C noktasında aydınlatma şiddeti hesaplamak için gerekli olan $\alpha$ ve $\theta$ açıları. ....	78
Şekil 6.12. Işığın bir düzlemden yansımaları. ....	80
Şekil 6.13. Işığın duvarlardan yansımaları. ....	82
Şekil 6.14. Örnek armatür yerleşimleri. ....	87
Şekil 6.15. Armatürlerin düzenli bir şekilde dağıtımı.....	88
Şekil 6.16. Örnek bir mahalde aydınlatma.....	88
Şekil 6.17. Sınırları ve tipleri belirlenmiş mahal çokgenleri (odalar) ve elektrik malzeme sembolleri yerleştirilmiş kat planı.....	90
Şekil 6.18. Örnek bir graf yapısı. ....	91
Şekil 6.19. Genel bir ağaç yapısı.....	93
Şekil 6.20. Dört daire içeren örnek bir kat planı. ....	94
Şekil 6.21. Kat planının ağaç veri yapısı şeklinde modellenmesi.....	94
Şekil 6.22. Kat planı hiyerarşisi için kapıların tespit edilmesi.....	95
Şekil 6.23. Örnek daire hiyerarşisi. ....	97
Şekil 6.24. Aynı yükün farklı senaryolar ile bağlanması. ....	98
Şekil 6.25. Ağaç-liste hibrit veri yapısı.....	99
Şekil 6.26. Kat planının ağaç-liste hibrit yapısı. ....	100
Şekil 6.27. İki mahalın ortak duvar çizgileri.....	100

	<b><u>Sayfa</u></b>
Şekil 6.28. Mahal çokgeni birleştirme örnekleri.....	101
Şekil 6.29 Oda çokgenleri birleştirme.....	101
Şekil 6.30. İç oda çokgenleri birleştirme. ....	102
Şekil 6.31. Anahtar koyma noktası tespit etme.....	104
Şekil 6.32. Sistem tarafından otomatik oluşturulan aydınlatma tesisatı. ....	105
Şekil 6.33. Priz linyeleri için taslak hazırlama.....	107
Şekil 6.34. Sistem tarafından otomatik çizilmiş elektrik tesisatı planı. ....	108



## ÇİZELGELER DİZİNİ

	<u>Sayfa</u>
Çizelge 4.1. İkili sınıflandırma için karışıklık matrisi. ....	40
Çizelge 4.2. Çok sınıflı sınıflandırıcı için karışıklık matrisi. ....	42
Çizelge 4.3. Eksik öğrenme, olması gereken ve aşırı öğrenme [99]. ....	44
Çizelge 5.1. Veri setindeki tefriş sınıfları. ....	52
Çizelge 5.2. Tefriş sembolleri eğitim ve test veri setleri. ....	55
Çizelge 5.3. Eğitim ve test sonucu doğruluk ve kayıp sonuçları. ....	57
Çizelge 5.4. FDS_Full test veri seti için karışıklık matrisi. ....	60
Çizelge 5.5. FDS_Full test veri setinde her sınıf için hesaplanmış F1-puanı .....	62
Çizelge 5.6. Gerçek kat planlarında test sonuçları. ....	63
Çizelge 6.1. Tefrişe göre elektrik malzemesi önerme tablosu. ....	71
Çizelge 6.2. Tefriş sembolüne göre mahal-tipi puanlama tablosu. ....	75
Çizelge 6.3. Örnek mahalde tefriş sembollerine göre mahal tipi belirleme. ....	76
Çizelge 6.4. Mahal-Tipine göre aydınlatma armatürü tipi belirleme. ....	84
Çizelge 6.5. Örnek mahal aydınlatma parametreleri. ....	88

## SİMGELER VE KISALTMALAR DİZİNİ

### SİMGELER

A : Amper

k : İletkenlik katsayısı

kW : Kilowatt

lm : Lümen (Işık akısı)

lx : Lüks (Aydınlatma şiddeti)

m : Metre

U : Gerilim

V : Volt

VA : Voltamper

W : Watt

$\Omega$  : Direnç

$\varphi$  : Voltaj ile akım arasındaki faz açısı ( $\cos \varphi$  - Güç faktörü)

## KISALTMALAR

2D	: Two Dimensional (2B – İki Boyutlu)
3D	: Three Dimensional (3B – Üç Boyutlu)
AC	: Alternatif Akım
AG	: Alçak Gerilim
AI	: Artificial Intelligence (Yapay Zekâ)
CAD	: Computer Aided Design (Bilgisayar Destekli Tasarım)
CCW	: Counter-Clockwise (Saat yönünün tersi)
CNN	: Convolutional Neural Network
D-CNN	: Deep Convolutional Neural Network (Derin Evrişimsel Sinir Ağları)
DL	: Deep Learning (Derin Öğrenme)
DWG	: Autocad Drawing (Autocad Standart Çizim Formatı)
DXF	: Drawing eXchange Format (Drawing interchange format)
EMO	: Elektrik Mühendisleri Odası
ESA	: Evrişimsel Sinir Ağları
FDS	: Furnishing Dataset (Tefriş Veri Seti)
GPU	: Graphical Process Unit (Grafik İşlem Birimi)
ILSVRC	: The ImageNet Large Scale Visual Recognition Challenge
ML	: Machine Learning (Makine Öğrenmesi)
MST	: Minimum Spanning Tree (En Az Yayılan Ağaç)
SMT	: Steiner Minimal Tree
TL	: Transfer Learning (Transfer Öğrenme)
TMMOB	: Türk Mühendis ve Mimar Odaları Birliği
TS	: Türk Standardı
VLSI	: Very-Large-Scale Integration (Çok geniş ölçekli entegre devreleri)
YSA	: Yapay Sinir Ağları
YZ	: Yapay Zekâ

# BÖLÜM 1

## GİRİŞ

Elektrik iç tesisat plan çizimleri, bir binanın elektrik ihtiyacını sağlamak için uygulanması gereken kurulum projesinin şematik olarak gösterilmesidir. Çizgiler ve sembollerden oluşan elektrik plan çizimleri, bilgisayar yardımıyla tasarım uygulamaları kullanılarak vektörel çizim ortamında hazırlanmaktadır. Elektrik tesisatı plan çizimleri için kurallar, ülkelere veya bölgelere göre değişiklik gösterse de temel işlem, şebeke elektriğinin elektrik malzemelerine ulaştırılmasıdır.

Yapıların elektrik tesisat projeleri, kuvvetli ve zayıf akım olmak üzere ikiye ayrılır: Kuvvetli akım tesisatı, aydınlatma ve priz/motor tesisatlarını kapsar. Zayıf akım tesisatı ise zil, telefon, televizyon, kamera, ses, yangın alarm sistemleri gibi birçok tesisat detaylarını içermektedir. Bu çalışma, elektrik tesisat planlarının kuvvetli akım kısmı ile ilgilidir.

Bir binanın kuvvetli akım elektrik projesinin hazırlanması, çizim ve hesaplama olmak üzere iki aşamadan oluşmaktadır. Çizimler, bilgisayar ortamında mimarların hazırlamış oldukları mimari kat planları temel alınarak 2D CAD programları ile hazırlanırlar [1]. Hesaplamalar bazı uygulamalarda otomatik yapılabilsen bile çizimler manuel olarak yapılmaktadır.

### 1.1. ÇALIŞMANIN AMACI

Elektrik tesisat plan çizimleri; mimari kat planlarındaki duvarlar, kapılar ve diğer dekoratif semboller dikkate alınarak hazırlanmaktadır. Çizim işlemleri, kat planlarındaki dekoratif sembolere uygun bir şekilde elektrik malzeme sembollerinin yerleştirilmesi ile başlar. Daha sonra konulan bu malzeme sembolleri ile dağıtım panoları arasındaki bağlantı hatlarının uygun bir şekilde çizilmesi ile bitirilir. Bu işlemler manuel olarak yapılan zaman alıcı işlemlerdir.

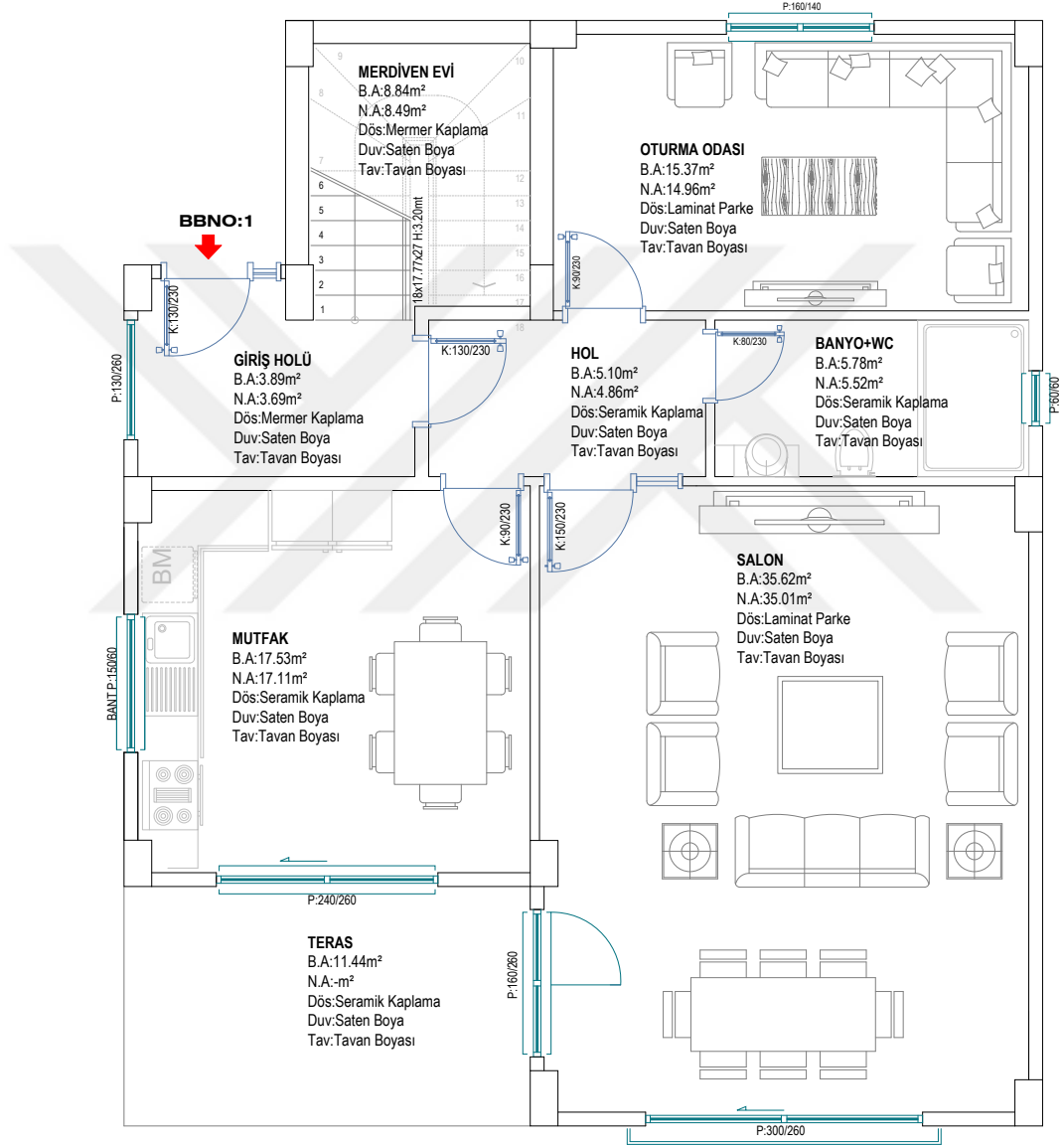
Günümüzde sürekli artan kalite ve rekabetçilik taleplerini karşılamak için hızlı, ucuz ve güvenilir çözümler üretmek mühendislerin en önemli amaçlarından biri olmuştur. Bugün yapay zekâ gelişmekte ve neredeyse her alanda kullanılmaktadır. Bu bağlamda elektrik plan çizimlerinde de yapay zekâ yöntemleri kullanılarak insan faktörünü en aza indirme, zamandan ve işçilikten kazanç sağlama ihtiyacı doğurmuştur.

Bu amaçla yapılan çalışmada, konut binalarında elektrik iç tesisat plan çizimleri hazırlanırken priz ve aydınlatma armatürleri gibi elektrik malzeme sembollerinin mimari kat planları üzerindeki tefriş sembolleri baz alınarak uygun yerlere otomatik olarak yerleştirilmelerini ve bağlantı şemalarının mekânsal ilişkiler göz önüne alınarak otomatik olarak oluşturulmasını sağlamak için akıllı bir model geliştirilmiştir. Sistem; mimari kat planlarındaki tefriş sembollerinin bulunması, tanımlanması ve bulunduğu mahal bölgelerinin birbirleri ile ilişkilendirilmesi esasına dayanmaktadır.

Kat planları, binaların mekânsal düzenlerini gösteren 2D ölçekli çizimlerdir. Bir projenin ihtiyaçlarına göre kat planlarının detayları farklı olabilir. Örneğin; taslak bir kat planında sadece duvar çizimleri bulunurken uygulanacak bir kat planında yapısal detaylar, sıhhi ve elektriksel özellikler ve peyzaj öğeleri bulunabilir. Özetle bir kat planı; büyüklükler, odalar, koridorlar, açıklıklar, iç özellikler, kapılar/pencereler, lavabolar ve mobilyalar dahil olmak üzere oda yerleşimlerini temsil eder.

Tefriş sembolleri, kat planlarında buzdolabı, televizyon, ocak-fırın, çamaşır makinesi, bulaşık makinesi; koltuklar, masalar, yataklar, dolaplar, kapılar ve pencereler gibi malzemelerin çizim sembollerine verilen genel isimdir. Tefriş sembolleri, çizilen mimari projelerde oluşturulan odaların kullanım sırasında gerekli olan eşyaların yerleşip yerleşmediğini göstermesi bakımından önemlidir. Özellikle elektrik ile sıhhi tesisat gibi çizim ve hesap gerektiren projelerde, gerekli çizim ve hesaplamaların yapılabilmesi mimari projelerdeki tefriş sembollerinin yerleşimine bağlıdır [2]. Örneğin, mimari projelerde bir bölgeye çamaşır makinesi sembolü konulmuş ise elektrik plan çizimleri hazırlanırken o sembolün olduğu yere çamaşır makinesi için bir priz sembolü konulmaktadır. Benzer şekilde, sıhhi tesisat projesi çiziminde de bu çamaşır makinesi tefrişi için temiz su temini ve kirli su gideri ayarlaması için gerekli çizimler yapılmaktadır.

Şekil 1.1'de tefriş sembolleri içeren örnek bir daire kat planı verilmiştir. Bu kat planında mobilya yerleşimleri yanında kapı, pencere, merdiven gibi diğer mekânsal düzenler de gösterilmiştir. Bu kat planı için elektrik tesisat planı çizilirken tefriş sembolleri baz alındığında hangi elektrik malzemesinin nereye konulması gerektiği ve mekânsal düzene göre bağlantıların nasıl yapılması gerektiği kolaylıkla bilinmektedir.



Şekil 1.1. Tefriş sembolleri içeren örnek bir daire kat planı.

Bir tasarımın belirlenmiş bazı kısıtlara göre otomatik yapıldığı *otomatik tasarım* kavramı, bilgisayar destekli tasarımın otomatikleştirilmiş ya da genişletilmiş halini ifade etmektedir. Otomatik tasarım, bilinen bir aralık içinde en iyi tasarımı bulmak için öğrenme veya optimizasyon yöntemlerini kullanmaktadır.

Bu tez çalışmasında, konutlarda kuvvetli akım elektrik tesisat proje çizimlerinin kat planları üzerinde otomatik oluşturulması için derin öğrenme tabanlı bir model geliştirilmiştir.

Önerilen sistemin özellikleri aşağıdaki şekildedir:

- Kat planlarındaki tefriş sembollerinin sınıflandırılması için derin evrişimsel yapay sinir ağlarıyla beraber transfer öğrenme kullanılmıştır.
- Tefriş sembollerinin tanımlanması ve sınıflandırılması için geliştirilmeye açık *tefriş sembolleri veri seti* oluşturulmuştur.
- Tefriş sembollerine uygun elektrik malzemesi kullanımı için bir öneri sistemi geliştirilmiştir.
- Kat planı bölümlerinde bulunan tefriş sembollerinden mahal tanımak için bir puanlama sistemi önerilmiştir.
- Kat planlarındaki mahal bölgelerinin birbirileri ile ilişkilerini gösteren bir hiyerarşi grafi modeli önerilmiştir.
- Oluşturulan bu hiyerarşi grafları kullanılarak elektrik dağıtım planları otomatik olarak oluşturulmuştur.

Geliştirilen çalışma ile elektrik plan çizimi otomatikleştirilmiş, dolayısıyla zaman ve işçilikten kazanç elde edilmiştir. Bu çalışma ile son yıllarda yapay zekanın en popüler alanı olan derin öğrenmenin elektrik plan çizimi alanında kullanılması için yeni bir vizyon getirilmiş ve literatüre katkı sağlanmıştır.

Bu tez çalışması yedi bölümden oluşmaktadır. Birinci bölümde; yapılan çalışmanın amacı, kullanılan yöntemler, ilgili konuda yapılan çalışmalar ve literatüre yapılan katkı genel hatları ile anlatılmıştır. İkinci bölümde; bilgisayar destekli tasarım, vektörel çizim ortamı ve vektörel işlemlerden bahsedilmiştir. Üçüncü bölümde, yapılan çalışmanın anlaşılması için elektrik tesisatı plan/proje sektöründe kullanılan terimler ve yapılan işlemler anlatılmıştır. Dördüncü bölümde; yapay sinir ağları, makine öğrenmesi, derin öğrenme, evrişimsel sinir ağları ve transfer öğrenme hakkında kısa bilgiler verilerek evrişimsel sinir ağlarının katmanları, parametreleri, eğitim ve test veri setlerinin oluşturulması, sınıflandırma ölçüm metrikleri ve öğrenme kontrolünden

bahsedilmiştir. Beşinci bölümde, oluşturulan veri seti ve tefriş sınıflandırma için oluşturulan transfer öğrenme modeli hakkında bilgiler verilerek tefriş sınıflandırmanın deneysel sonuçları verilmiştir. Altıncı bölümde otomatik elektrik tesisatı çizimi ile ilgili detaylar açıklanmıştır. Bu bölümde kat planlarında tefriş sembolü tespiti ve sınıflandırılması, sınıflandırılan tefriş sembollerine göre otomatik elektrik sembollerinin uygun yerlere yerleştirilmesi açıklanmıştır. Ayrıca bu bölümde tefriş sembollerine göre mahal tanıma ve tanımlanan mahale göre aydınlatma tasarımı detaylandırılmıştır. Devamında konulmuş elektrik sembollerinin elektrik panosuna otomatik bağlanması için gerekli algoritmalar ve yapılan çalışmalar açıklanmıştır. Yedinci bölümde; sonuçlar verilmiş, kısıtlardan bahsedilmiş, öneriler ve sonraki çalışmalar tartışılmıştır.

## **1.2. LİTERATÜR ÖZETİ**

Bu çalışma, elektrik tesisatı plan çizimlerinin otomatik oluşturulması ile ilgilidir. Yapı sektöründe plan çizimleri genel olarak CAD uygulamalarında manuel olarak çizilmektedir.

Elektrik plan çizimleri veya devreleri AutoCAD Electrical, ElectricalOm, ProfiCAD, EPlan Electric, Designspark Electrical, Electra, Simaris gibi bazı CAD uygulamaları yardımıyla manuel olarak çizilip hesaplamalar bu çizimlere bağlı olarak otomatik yapılabilmektedir.

Bu çalışmanın öncesi sayılan ve “Yapı elektrik tesisat çizim ve optimizasyon uygulaması” isimindeki çalışmamız ile elektrik tesisat planlarının bilgisayar ortamında hızlı ve kolay çizilebilmesi için bazı algoritmalar geliştirilmiş, hesapların çizime entegre edilmesi ile otomatik olarak yapılması sağlanmıştır [1,3].

Bugüne kadar kat planı verilerinden elektrik planlarının otomatik olarak oluşturulması ile ilgili literatürde sadece bir çalışma yapılmıştır. Bu çalışma, yığın tabanlı bir sonlu durum makinesi modeli kullanarak konut elektrik planlarını otomatik olarak oluşturmak için prosedürel bir yaklaşım önermektedir [4]. Bu sistem giriş olarak duvar konumları gibi 2Dmimari verileri alıp çıktı olarak özelleştirilmiş bir elektrik planı vermektedir.

Elektrik tesisat planlarının otomatik çizilebilmesi, mimari kat planlarının düzenine, hiyerarşilerinin oluşturulabilmesine ve planlardaki tanımlayıcı semboller olan tefriş sembollerinin tespit edilip tanımlanmalarına bağlıdır.

Tefriş sembollerinin sınıflandırılması, bu çalışmanın temelini oluşturmaktadır. Çünkü bu çalışmadaki mahal bölgelerinin tanımlanması ve aralarındaki ilişkilerin kurulması gibi diğer tüm işlemler, tefriş sembollerinin bulunması ve tanımlanmasına dayalı olarak yürütülmektedir. Önerilen modelde tefriş sembollerinin sınıflandırılması, bir görüntü sınıflandırma problemidir. Görüntü sınıflandırma, bir görüntüyü girdi olarak alan ve onu önceden belirlenmiş bir sınıfa kategorize eden bir bilgisayarlı görü problemidir.

Bilgisayarlı görü; görüntü, video veya diğer görsel girdilerden anlamlı bilgiler üretmeyi sağlayan bir yapay zekâ alanıdır [5]. İnsanlar, görüntülere bakınca nesnelere ayırt edebilir, tanıyabilir veya gruplandırabilir. Benzer şekilde bilgisayarlı görüde de temel amaç, görüntüler veya videolar girdi olarak verildiğinde insan beyninin yapabildiği gibi nesnelere tespit edebilmek, tanıyabilmek ve sınıflayabilmektir.

Dokümanlarda sembol tanıma, genel örüntü tanıma probleminin özel bir uygulamasıdır. Örüntü tanıma belli örnekleri ve bunlara ait verileri kullanarak sonraki örnekler için karar verme sürecidir [6]. Daha açık bir ifade ile örüntü tanıma, "Bu nedir?" sorusuna yanıt verme ile ilgilenen bir yöntemdir. Semboller, belirli bir alanda anlamsal bilgiyi ileten minimum bileşenler olan grafiksel nesnelere sahiptir. Logolar, silüetler, müzik notaları ve mühendislik, elektronik veya mimari özelliklere sahip basit çizgi grupları, doküman görüntü analizi topluluğu tarafından araştırılan bazı sembol örneklerini oluşturmaktadır [7,8].

Geçmişten günümüze sınıflandırma yapmak için yapay sinir ağları, uzman sistemler ve bulanık kümeler gibi birçok sınıflandırma yaklaşımı yaygın olarak kullanılmıştır. Bu yöntemlerin her birinin başarımları, derin öğrenmeye nispeten daha düşüktür [9]. Başarımları dikkate alındığında derin öğrenmenin sınıflandırma işlemleri için günümüzde en iyi çözüm olduğu anlaşılmaktadır.

Son birkaç yılda, özellikle internet aracılığıyla toplanan çok büyük miktarda verinin mevcudiyeti ve erişilebilirliği sayesinde, bilgisayar uygulamaları basit veri işleme işlemlerinden makine öğrenmesi işlemlerine çarpıcı bir dönüşüm geçirmiştir. Makine öğrenmesi fikri, bilgisayarların zaman geçtikçe kendini geliştirme yeteneğine sahip olduğu gerçeğini göstermektedir. Son yıllarda, makine öğrenmesi, bilgisayarlı görü ve görüntü tanıma konularında yapılan konferanslar, çalıştaylar, tartışmalar, deneyler ve gerçek yaşam uygulamaları, araştırmacılar tarafından büyük ilgi görmektedir. Bilgisayarlı görüde makine öğrenmesinin en yeni uygulamaları nesne algılama, nesne sınıflandırma ve görüntülerden, grafiksel dosyalardan ve videolardan ilgili bilgilerin çıkarılmasıdır [10].

Yapılan bu tez çalışmasında çalışma ortamı vektörel çizimlerdir. Kullanılan dosyalar piksel görüntülerden değil, vektörel çizimlerden oluşan mimari kat planlarıdır. Mimari kat planları çoğunlukla CAD uygulamaları kullanılarak vektörel ortamda hazırlanmış olsa bile literatürde taranmış kat planları üzerinde çok fazla çalışma mevcuttur. Bu çalışmalar genellikle doküman analizi alanında yapılan çalışmalar olup taranmış kat planlarında sembol tespiti ve sembol tanıma ile ilgilenmektedir.

Kat planı dosyaları; duvarları, odaları, kapıları, pencereleri ve diğer dekoratif sembolleri içeren dosyalardır. Bugüne kadar mimari kat planlarını inceleyen birçok çalışma yapılmıştır. Bu çalışmalar çoğunlukla örüntü tanıma yaklaşımına dayalı olarak 2D mimari kat planlarından 3D yapı modelleri elde etmeyi amaçlamıştır [11–14].

Kat planı analizi, belirli bir 2D mimari kat planını analiz ederek anlamsal bilgileri elde eden kapsamlı bir işlemdir. Kat planı analiz sistemleri genellikle segmentasyon, yapısal analiz ve anlamsal bilgi çıkarma işlemlerinden oluşur. Oluşan bilgiler amaçlanan uygulamalarda kullanılmak üzere veri tabanına kaydedilir. Tipik olarak elde edilen bilgiler 3D bina modelleri oluşturmak için kullanılır [15].

Ah-Soon ve Tombre [16] bilgisayarlı görü ve örüntü tanıma teknikleri kullanarak taranan mimari kat planlarından kapı ve pencere gibi mimari sembolleri tanımak için bir yöntem önermişlerdir. Önerdikleri yöntem, geometrik özellikler üzerindeki bazı

kısıtlar aracılığıyla modelin tanımlanmasına ve çizimden çıkarılan özelliklerin kısıtlar ağı aracılığıyla yayılmasına dayanmaktadır.

Dosch vd. [17] segmentasyon, vektörleştirme ve özellik algılama gibi görüntü işleme ve bilgisayarlı görü araçlarını kullanarak 2D kat planlarındaki sembolleri tanıyıp 3D binaları oluşturmak için bir sistem tanımlamışlardır. Sistem ayrıca güçlü ve esnek bir kullanıcı arayüzü içermektedir.

Guo vd. [18] çalışmalarında mühendislik çizimlerinden sembolleri analiz etmek için örnek odaklı bir sembol tanıma yaklaşımı kullanmışlardır. Birçok alana uygulanabilen bu çalışma, mimari kat planlarında test edilmiş ve dekoratif semboller başarıyla tanınmıştır.

Heras vd. [19] mimari nesnelere ve yapısal ilişkileri için açıklamalı CVC-FP adlı bir tanımlanmış gerçek mimari kat planı veri tabanı sunmuşlardır. Bu veri tabanını oluşturmak için mimari nesnelere arasındaki ilişkileri doğal bir şekilde spesifik hale getirmeyi sağlayan SGT adında bir araç kullanmışlardır. Ayrıca çalışmalarında duvar segmentasyonu ve mahal algılama görevlerinde bir kıyaslama aracı sunmuşlardır.

Rezvanifar vd. [8] mimari çizim taramalarından sembol bulma (tanıma değil) hakkında geniş bir literatür taraması sunmuşlardır. Sembol bulma, büyük resim veya belgelerde gömülü sembollerin konumlarının tespitini ifade etmektedir.

Geleneksel sembol bulma yaklaşımları piksel tabanlı ve vektör tabanlı olmak üzere iki kategoride değerlendirilir:

Piksel tabanlı yaklaşımlarda; sembollerin F-imzalarını (ölçekleme, öteleme, simetri ve döndürme gibi temel geometrik dönüşümlerle değişmez imzaları) eşleştirme [20], histogram benzerliğini piksel düzeyinde eşleştirme [21], komşuluk tabanlı puanlama olan BSM (Blurred Shape Model) [22], ve CBSM (Circular Blurred Shape Model) [23] eşleştirme, ilgi noktalarına göre SCIP (Shape Context for Interest Points) [24] ve uzantısı olan ESCIP (Extension of SCIP) [25] eşleştirme yöntemleri önerilmiştir. Piksel tabanlı yaklaşımların yavaş olmaları ve eşleştirme aşamalarının yüksek

hesaplama karmaşıklığına sahip olmaları gerçek uygulamalarda ciddi problemler oluşturmaktadır.

Vektörel tabanlı yaklaşımlar; sembollerdeki anlamlı bölümler olan çizgileri, daireleri, köşeleri, uçları, kritik noktaları, dış bükey bölgeleri, kapalı bölgeleri, bağlı bileşenlerden oluşan görüntü bölgeleri gibi vektörel özellikleri bir yapısal graf ile kodlayarak oluşturmak ile başlar [26–36]. Bu yapısal graflar; nesnelerin birleşme veya kesişme niteliklerini belirleyen ARG grafi (Attributed Relational Graph) [27,37], farklı olası vektörleştirme hatalarını ele alan hiyerarşik uygunluk grafi HPG (Hierarchical Plausibility Graph) [30,31], bölge sınırlarını ve bölgeler arasındaki ilişkiyi karakterize eden bölge komşuluk grafi RAG (Region Adjacency Graph) şeklinde olabilir. Daha sonra bu temsil graflar ile eşleştirmeler yapılır. Graf eşleştirme maliyetli bir işlem olduğu için graf serileştirme [38], graf gömme [39] ve ilişki indeksleme [26] gibi alternatif eşleştirme yöntemleri önerilmiştir.

Geleneksel olmayan derin öğrenme tabanlı sembol bulma yöntemleri, literatüre yakın zamanda girmeye başlamıştır. Bu yaklaşımlar da genellikle sembol tanıma işlemlerine odaklandığı için geleneksel sembol bulma yaklaşımlarının güncel konularından uzaktırlar.

Dey vd. [40] çok yazarlı bir senaryoda sembol tanımlama için LeNet'ten ilham alan sığ bir evrişimsel sinir ağı önermiştir. Sonuç olarak önerdiği sinir ağı mimarisinin farklı yazarlardan elle çizilmiş sembolleri, verimli bir şekilde tanıyabildiği ve yazarlar arası sapmayı modelleyebildiğini ortaya koymuştur.

Riba vd. [41] grafik sembolleri graf tabanlı tanımak için sembollerin yapısal graf temsillerini tanımlayıp ardından eşleştirme için çıktığı kullanan MPNN (message passing neural network) derin öğrenme ağını kullanmışlardır.

Ghost vd. [42] mimari projelerde bulunan sembolleri piksel tabanlı olarak tespit edip sınıflandıran GSD-Net'i önermiştir.

Ziran ve Marinai [43] kat planı görüntülerinde mobilya, kapı ve pencere nesnelere tespiti ve tanımlanması için derin sinir ağlarının kullanımını önermiştir.

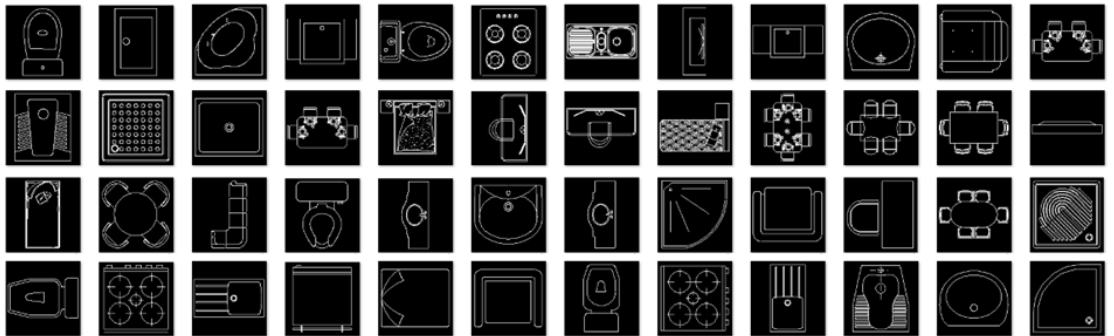
Rezvanifar vd. [44] mimari kat planlarında sembol tanımlama için YOLO (You-Only-Look-Once) mimarisini kullanarak derin öğrenmeye dayalı bir yöntem kullanmışlardır. Çalışmalarını SESYD veri seti üzerinde test ederek derin öğrenme ile önerilen yaklaşımın geleneksel yöntemlerden daha iyi olduğunu belirtmişlerdir.

SESYD (Systems Evaluation SYnthetic Documents) veri seti; kat planlarını, diyagramları, çizimleri ve diğer bazı nesnelere içeren halka açık doküman analizinde en çok kullanılan bir veri setidir [45,46].

Fan vd. [47] FloorPlanCAD adında 15.000 taramış kat planı içeren büyük bir CAD çizim veri seti oluşturmuşlardır. Literatürde kat planları için GREC'2003 [48], GREC'2011 [49], FPLAN-POLY [26] gibi birkaç başka veri seti olmasına rağmen, bunlar da taramış (piksel) görüntülerden oluşmaktadır.

Pratikte CAD uygulamalarında (özellikle tasarımla ilgili endüstride) taramış kat planları kullanılmaz. Bunun yerine vektörel formatta kat planları kullanılmaktadır. Bu nedenle, bu veri setlerinin hiçbiri çalışmamız için uygun değildir. Yapılan çalışmamızda girdi ve çıktı verileri vektör (DXF/DWG) formatındadır.

Bu çalışmada, ilk önce yüzlerce farklı vektörel mimari kat planından tefriş sembollerini ayıklayarak kendi veri setimizi oluşturduk. Oluşturulan veri seti hem vektör tabanlı hem de piksel tabanlı geliştirmeye açık bir şekilde kaydedildi. Oluşturulan veri seti, derin öğrenme tabanlı modelimizde eğitim ve test için kullanıldı. Oluşturulan veri setindeki tefriş sembollerinden bazı örnekler Şekil 1.2'de gösterilmiştir.



Şekil 1.2. Veri setindeki tefriş sembollerinden örnekler.

### 1.3. ÇALIŞMANIN LİTERATÜRE KATKILARI

Bu tez çalışması kapsamında, uluslararası indekslerde taranan hakemli dergide bir adet yayın ve uluslararası konferanslarda sunulmuş ve tam metni basılmış iki adet bildiri yapılmıştır:

SCI, SCI-Expanded, SSCI veya AHCI kapsamındaki hakemli dergilerde yayımlanmış makale:

- Akgül, B. and Kutucu, H., “An automated system for electrical power symbol placement in electrical plan drawing”, *Automatika*, 63(1), 78-89, (2022).

Uluslararası sempozyumda/kongrede sunulmuş ve tam metni basılmış sözlü bildiriler:

- Akgül, B. and Kutucu, H., “An Intelligent Material Placement for Electrical Installation Project”, *International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES 2018)*, Karabük, 442–446, (2018).
- Akgül, B., Bilici, A. and Kutucu, H., “A Fast and Simple computer aided lighting simulator”, *International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES 2018)*, Karabük, 639–641, (2018).

## BÖLÜM 2

### BİLGİSAYAR DESTEKLİ TASARIM

Birçok sektörde bir işi yapmadan önce tasarlamak gerekir. Özellikle inşaat sektöründe yapılacak tüm işlerin yapılmaya başlamadan önce projelendirilmesi gerekmektedir. Eskiden yapılan tasarımlar ve projeler kâğıt üzerine elle çizilmekteydi. Günümüzde inşaat, otomotiv, enerji, giyim, oyunlar gibi birçok sektörlerde özellikle mühendislik ve tasarım için bilgisayarlar kullanılmaktadır. Bilgisayar destekli tasarım, İngilizce’de bu işi yapmak için kullanılan programlara verilen genel bir ad olan “Computer Aided Design” sözcüklerinin baş harflerinden oluşan CAD terimi ile ifade edilir. Computer-aided design and drafting (CADD) olarak da bilinir. Bilgisayar destekli tasarımın en büyük avantajı, yapılan tasarımların ve projelerin kolayca yenilenebilmeleridir.

CAD ve uygulamaları, mühendislik ve mimari gibi tasarımla ilgili alanların en popüler konularındandır. CAD sistemleri, tasarımcıların üretkenliğini artırmak, tasarımların kalitesini iyileştirmek, dokümantasyon yoluyla iletişimi geliştirmek ve üretim için veri tabanı oluşturmak gibi birçok amaç için kullanılmaktadır. CAD sistemleri, proje çizimlerinde üretimi manuel yapmaya göre en az beş kat arttırmaktadır [50].

Yapı ve tasarımla ilgili tüm sektörlerde özellikle mühendislik, mimarlık ve tasarım hizmetleri için 2D/3D birçok CAD programı kullanılmaktadır. Autocad, Revit, Archicad, SolidWorks, SketchUp, StaCad, Idecad, 3DS Max bunların en popüler olanlarıdır. Günümüzde inşaat sektöründe en çok kullanılan CAD uygulaması AutoCAD programıdır. AutoCAD hem 2D hem de 3D tasarım yapabilen bir programdır. Proje ve tasarım sektöründe çalışan hemen hemen bütün mühendis ve mimarlar tarafından kullanılmaktadır. Birçok üniversitede bilgisayar yardımıyla tasarım derslerinde eğitim içeriği olarak kullanılmaktadır.

## 2.1. VEKTÖREL ÇİZİM

Vektörel çizimler, çözünürlükten bağımsız olarak uygulama programları tarafından çalışma anında matematiksel ifadelerle oluşturulan ve detay kaybetmeden istenen herhangi bir boyuta yeniden ölçeklendirilebilen grafik türleridir. Vektörel çizimler iki veya üç boyutlu olabilirler. Bilgisayar destekli tasarım programlarının tamamına yakını vektörel çizim tabanlı programlardır.

Vektörel çizimlerin en önemli özelliği, çalışma anında matematiksel ifadeler sonucunda oluşturulmalarıdır. Örneğin; bir çizgi parçası için iki nokta, bir çember için bir nokta ve bir yarıçap verileri bu çizimleri oluşturmak için yeterlidir. Vektörel çizimler oluşturulurken çizgi stilleri, çizgi renkleri ve çizgi kalınlıkları gibi bazı formatlama nesnelere ile biçimlendirilirler.

Yapı sektöründeki vektörel çizimler yaygın olarak DWG veya DXF gibi dosya formatlarında kaydedilerek paylaşmakta, saklanmakta veya CAD uygulamaları arasında veri alışverişi yapmak için kullanılmaktadır.

Vektörel çizimler grafik nesnelere ve biçimlendirme (format) nesnelere olmak üzere iki kısımdan oluşmaktadır: Grafik nesnelere; çizgi, çember, yay, elips, tarama, yazı, çokgenler, polyline, spline gibi matematiksel ifadeler sonucu oluşturulan çizim nesnelere. Format nesnelere ise çizim nesnelere biçimlendirmek için kullanılan çizgi tipleri, çizgi kalınlıkları, çizgi renkleri gibi biçimlendirme nesnelere.

Vektörel çizimler genellikle katmanlar halinde oluşturulur. Bir katmanda çizilmiş nesnelere aksi belirtilmediği sürece o katmanda belirlenen çizgi rengi, çizgi kalınlığı, çizgi tipinde biçimlendirilir. Katman özellikleri kullanılarak çizimler hızlıca biçimlendirilebilmektedir. Ayrıca katman ayarları ile çizimlerin belirli kısımlarını gösterme veya gizleme, yazdırma veya yazdırmama gibi işlemler de kolayca yapılabilmektedir.

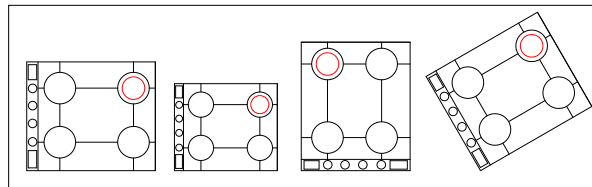
Mimari plan çizimleri de genellikle katmanlar kullanılarak hazırlanmaktadır. Katmanlar üst üste konulmuş çizimlere benzemektedir. Örneğin bir binanın tüm projesinde duvarlar, sıva, ölçüler, tefrişler, elektrik tesisatı planı, yangın alarm planı,

su tesisatı planı gibi birçok çizim ayrı katmanlarda tutulmaktadır. Bu katmanlar CAD uygulamalarında kolaylıkla görünür/görünmez yapılarak sadece istenen katmandaki nesnelerin görünebilir olması sağlanabilir. Aynı şekilde katmanlar yazdırılır/yazdırılmaz yapılarak sadece istenen çizim kısımlarının yazdırılması sağlanabilir. Projelerin amacına göre katman özellikleri kullanılarak çizimlerin belirli kısımlarının (örneğin duvarlar) vurgulanması için koyu renkler kullanılıp çizgi kalınlıkları arttırılabilir, bazı kısımların (örneğin döşemeler) daha silik kalması için açık renkler kullanılıp çizgi kalınlıkları azaltılabilir.

Mimari çizimler, bir yapının ölçekli olarak hazırlanmış teknik çizimidir. İnşa edilecek bir yapı, mimari çizime uygun olarak yapılır. Bina mimari çizimleri, genellikle mimarlar tarafından CAD programları kullanılarak hazırlanır. Bu çizimlerde genellikle çizgi, polyline, çember, tarama gibi temel çizim nesneleri kullanılmaktadır.

Mimari projelerde ana bileşen, kat planlarıdır. Kat planları, mimari projelerdeki mekânsal düzeni gösteren 2D ölçekli çizimlerdir. Kat planları; büyüklükler, koridorlar, iç özellikler, kapılar, pencereler, lavabolar ve mobilyalarla birlikte oda yerleşimlerini temsil etmektedir. Bir binanın statik, sıhhi, kalorifer, doğalgaz, elektrik, telefon, zil/diafon, televizyon, yangın alarm, kamera/güvenlik vb. tesisat projelerinin tamamı mimari kat planları baz alınarak vektörel formatta hazırlanmaktadır.

Mimari çizimlerde kapılar, pencereler, mobilyalar, mutfak ve banyo detayları gibi tefrişatlar sık kullanılan bileşenlerdir. Bu çizimler, çok sık kullanıldıkları için bir araya getirilip blok olarak kaydedilirler. Çizim blokları, birden fazla çizim nesnesinin farklı yerlerde tekrardan kolayca kullanılabilmesi için bir araya getirilmiş çizim gruplarıdır. Temel amaç, aynı şekilleri, farklı çizimlerde, farklı ölçülerde, farklı açılarla kullanabilmektir. Aynı blok nesnesinin farklı döndürme açıları (rotation angle) ve farklı büyüklükler (scale) ile kullanımı Şekil 2.1’te verilmiştir.



Şekil 2.1. Blok kullanımı.

## 2.2. VEKTÖREL İŞLEMLER

Bilgisayar destekli tasarım uygulamalarında öteleme, ölçekleme ve döndürme olmak üzere üç temel dönüşüm işlemi vardır. Bu temel işlemler kullanılarak grafik nesnelere üzerinde yansıtma, uzatma, kırpma, esnetme, paralel çıkarma, kesişim noktaları bulma gibi diğer vektörel işlemler yapılabilmektedir.

$P=[x,y,z,1]^T$  olmak üzere  $P$  noktası dönüşüm işlemlerine tabi tutulursa  $P'$  noktası Eşitlik 2.1'deki gibi hesaplanır.

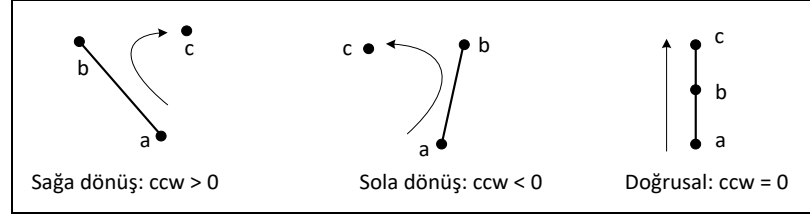
$$P' = T * P, \quad T = \begin{cases} \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, & \text{öteleme} \\ \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & \text{ölçekleme} \\ \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & \text{xy düzleminde döndürme} \end{cases} \quad (2.1)$$

2D CAD uygulamalarında vektörel işlemler, noktalar ve çizgi parçaları üzerinde yapılmaktadır. Çember, yay, elips, polyline ve spline gibi diğer nesnelere üzerinde yapılan işlemler de aslında bu nesnelere küçük doğru parçalarına dönüştürülerek çizgi parçaları üzerinde yapılan işlemlerdir.

Üç noktanın birbirine göre durumu *CCW* (counterclockwise) yöntemi ile bulunabilmektedir (Eşitlik 2.2).

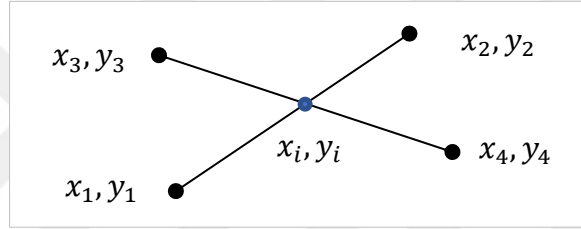
$$ccw(a, b, c) = \begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix} \quad (2.2)$$

Şekil 2.2'te gösterildiği gibi herhangi bir  $c$  noktasının  $a$  ve  $b$  noktasının oluşturduğu  $ab$  doğru parçasının sağında veya solunda olduğunu anlamak için determinant sonucuna bakılır. Eğer determinant sonucu sıfır ise  $a, b, c$  noktaları doğrusaldır. Sonuç sıfırdan küçük ise  $c$  noktası  $ab$  doğru parçasının solunda, sıfırdan büyük ise sağındadır.



Şekil 2.2. CCW yönteminin gösterimi [51].

Vektörel işlemlerde kullanılan doğru parçaları birbirilerine göre doğrusal, paralel, ayırık veya kesişen olabilirler. Bu doğru parçalarının birbirine göre durumları CCW ile hesaplanabilmektedir. Bu yöntem ile herhangi iki geometrik nesnenin birbirini kapsadığı veya kesiştiği de kontrol edilebilir.



Şekil 2.3. İki doğru parçasının kesişimi.

Şekil 2.3'te gösterilen ve başlangıç ile bitiş noktaları verilen iki doğru parçasının kesişim noktası Eşitlik 2.3'e göre hesaplanır.

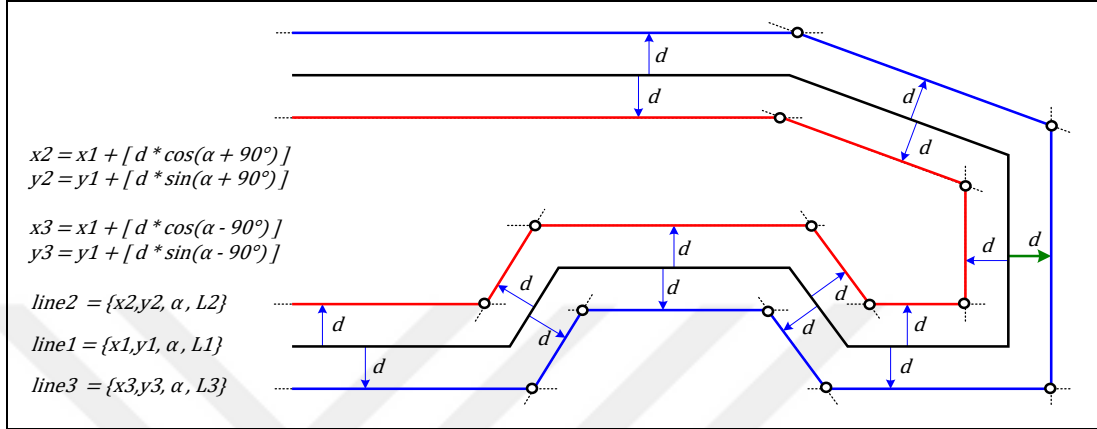
$$x_i = \frac{(x_1 y_2 - y_1 x_2)(x_3 - x_4) - (x_1 - x_2)(x_3 y_4 - y_3 x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)} \quad (2.3)$$

$$y_i = \frac{(x_1 y_2 - y_1 x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 y_4 - y_3 x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

CAD uygulamalarında sık kullanılan işlemlerden biri de paralel çıkarma (offset) işlemidir. Paralel çıkarma işlemi, linye ve sorti çizimlerinde kullanıldığından yapılan çalışmada otomatik tasarım işleminin önemli adımlarından biridir. Literatürde polyline nesnelere için birçok paralel çıkarma yöntemi bulunmaktadır [52–54].

Polyline nesnelere nokta dizilerinden oluşmaktadır. Art arda gelen iki nokta bir çizgi parçası oluşturur. Bir çizgi parçasının paralelini çıkarmak için çizginin açısına  $90^\circ$

ekleyerek veya açılarından  $90^\circ$  çıkararak  $d$  mesafesinde öteleme yapılır. Aynı işlem, tüm çizgi parçaları için yapılır ve daha sonra art arda gelen çizgi parçaları kesiştirilirse kesişim noktaları paralel polyline nesnesinin noktalarını oluşturur. Şekil 2.4'te bir polyline nesnesinin  $+90^\circ$  ve  $-90^\circ$  ile  $d$  mesafesinde paralelleri çıkarılmıştır.



Şekil 2.4. Polyline nesnesinin paralelini oluşturma.

## BÖLÜM 3

### ELEKTRİK TESİSAT PLANLARI

Elektrik enerjisini kaynaktan kullanılacak yerlere kadar taşınması için yapılan tesisatların plan çizimlerine *elektrik tesisat planı* denir. Elektrik enerjisinin üretim yerinden tüketim bölgesine kadar yapılan dağıtımlara *dış tesisat*, kapalı alanlarda sayaç panolarından priz veya aydınlatma armatürlerine kadar yapılan dağıtımlara ise *iç tesisat* denilmektedir. Bir binanın elektrik ihtiyacını sağlamak için uygulanması gereken kurulum projesinin şematik olarak gösterilmesine *elektrik iç tesisat planı* denilmektedir. Elektrik tesisat plan çizimlerinde uyulması gereken kurallar, yönetmelikler ve şartnameler ile düzenlenmektedir. Elektrik tesisat planları kuvvetli akım ve zayıf akım tesisatı olmak üzere ikiye ayrılmaktadır:

Kuvvetli akım tesisatı, insanlar, hayvanlar ve eşyalar için tehlikeli olabilecek elektrik akımı taşıyan (50V ve yukarısı) sistemlerin tesisatıdır. Bina elektrik tesisat projelerinde kuvvetli akım, AG (alçak gerilim – 1000V altı) ile çalışan priz, motor ve aydınlatma tesisleri için kullanılan bir terimdir. Kuvvetli akım plan çizimleri, priz, motor ve aydınlatma planlarını içermektedir.

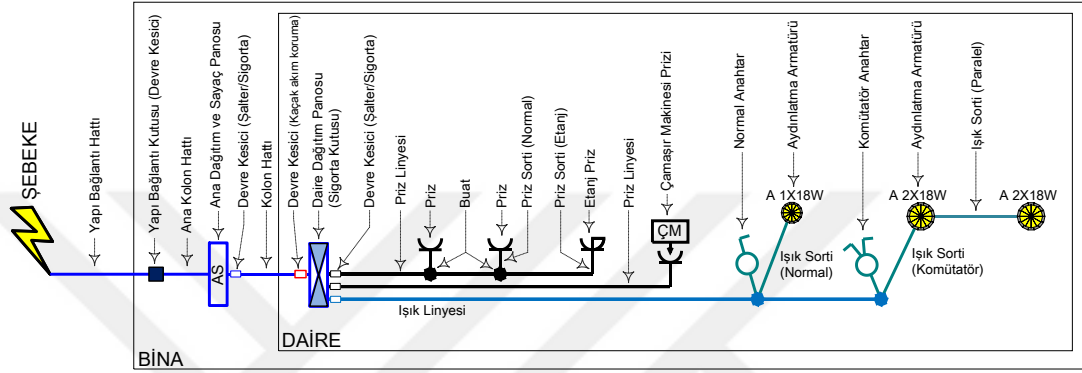
Zayıf akım tesisatı, insanlara zarar verebilecek elektrik akımı taşımayan (50V altı), zil/diyafon, telefon/data, televizyon uydu/anten, kamera, seslendirme, yangın alarm ve diğer bazı özel otomasyon sistemlerinin tesisatlarıdır. Zayıf akım tesisat planları bu sistemlerin bağlantı detaylarını içermektedir.

Bu tez çalışması konut binalarında kuvvetli akım elektrik tesisat planlarının otomatik oluşturulması ile ilgili olduğu için zayıf akım tesisat planlarından bahsedilmemiştir.

Kuvvetli akım elektrik tesisat projeleri, plan çizimi, hesaplamalar ve dökümlerden oluşmaktadır. Yapılan çalışmanın anlaşılması için elektrik tesisat planlarında yapılan işlemler ve kullanılan malzemeler bu bölümde özet bir şekilde açıklanmıştır.

### 3.1. ELEKTRİK MALZEMELERİ

Elektrik tesisat planlarında dağıtım panoları, prizler, anahtarlar, aydınlatma armatürleri, motorlar gibi malzemeler için semboller kullanılmaktadır. Elektrik projelerinde en sık kullanılan kuvvetli akım elektrik malzemeleri ve sembolleri Şekil 3.1’de gösterilmiştir.



Şekil 3.1. Kuvvetli akım elektrik malzemeleri.

Elektrik kaynağından (şebeke/trafo/jeneratör) gelen elektrik enerjisinin binaya bağlandığı noktaya *yapı bağlantı kutusu* veya *kofre* denir.

Elektrik enerjisini bina içinde katlara veya dairelere dağıtan dağıtım panoları vardır. Binada, enerjinin ilk girdiği pano *ana dağıtım panosudur*. İlk pano aynı zamanda sayaç panosu ise *ana dağıtım ve sayaç panosu* olur. Okul, hastane, yurt, alışveriş merkezleri gibi binalarda, enerjiyi aynı kata dağıtan panolara *kat dağıtım panosu* denir. Konut binalarında veya iş yerlerinde daire içi sigorta kutusuna *daire dağıtım panosu* denir.

Elektrik tesisatında kabloların birbirine bağlandığı ve dağıtımların yapıldığı noktalara *buat* denir. Elektrik tesisatı ağaç veri yapısı şeklinde modellenebildiği için buatlar düğüm olarak değerlendirilmektedir.

Kuvvetli akım plan çizimlerinde priz/motor ve aydınlatma plan çizimleri genellikle aynı kat planları üzerinde birlikte çizilmektedir. Bununla birlikte oteller, okullar, hastaneler, yurtlar ve alışveriş merkezleri gibi çok büyük yapılarda, planların daha iyi

okunabilmesi için priz/motor tesisatı ile aydınlatma tesisatı aynı kat planı kullanılarak ayrı paftalarda çizilebilmektedir.

### **3.1.1. Priz ve Motor Tesisatında Kullanılan Malzemeler**

Elektrik ile çalışan cihazları, elektrik şebekesine bağlamak için kullanılan araçlara *priz* denir. Genel amaçlı herhangi bir elektrikli ev aleti için kullanılan prizlere *normal priz*, ıslak zemin veya nemli yerlerde kullanılan prizlere *etanj priz*, üç fazlı araçlar için kullanılan prizlere *sanayi prizi* denilmektedir.

Bunların yanında çamaşır makinesi, bulaşık makinesi, fırın, şofben, kombi, klima ve ısıtıcı gibi sabit elektrikli ev aletleri için kullanılan prizlere elektrik planlarında *müstakil priz* denilmektedir. Bunların kurulu güçleri malzemesine göre 2000-2500W olacak şekilde projelerde belirtilmektedir. Müstakil priz, malzeme olarak aslında normal veya etanj prizdir. Ancak bu hatlara başka prizler bağlanmaz. Tek priz linyesi olarak değerlendirildiği için tek sigortadan bağımsız olarak çekilir. Elektrik projelerinde normal prizlerin kurulu güçleri belirtilmediği sürece 300 W olarak kabul edilmektedir.

Elektrik tesisat projelerinde asansörler, su pompaları, hidroforlar, havalandırma ve iklimlendirme sistemleri, garaj ve bahçe kapıları birer *motor* olarak değerlendirilir.

### **3.1.2. Aydınlatma Tesisatında Kullanılan Malzemeler**

Bir veya birden fazla lambayı ve çalışması için gerekli parçaları barındıran aydınlatma malzemelerine aydınlatma armatürü denir. Günümüzde farklı firmaların farklı amaçlarla farklı mahallerde kullanılmak üzere ürettiği binlerce farklı aydınlatma armatürü vardır. Genel olarak aydınlatma işlemi için mahallerin özelliklerine, fiziksel yapılarına ve olması gereken veya istenen minimum aydınlatma seviyesine göre farklı armatür türleri ve sayıları kullanılmaktadır. Aydınlatma hesabı yapılan profesyonel aydınlatma projelerinde ışık akıları ve ışık dağılım eğrileri bilinen aydınlatma armatürleri kullanılmaktadır.

Aydınlatma armatürlerini kumanda etmek için kullanılan tesisat malzemelerine *anahtar* denir. Tek bir armatürü kontrol eden anahtara *normal anahtar*, birden fazla armatür veya armatür grubunu kontrol eden anahtarlara *komütatör anahtar*, bir armatürü veya armatür grubunu aynı anda farklı noktalardan kontrol edebilen anahtarlara *vavien anahtar* denilmektedir.

### 3.2. HATLAR (BAĞLANTILAR)

Enerji dağıtımını, elektrik kaynağından son tüketilen malzemelere kadar ileten iletkenlere elektrik tesisatında hat denir. Elektrik tesisatında hatlar, havai hat, kablo, bara veya busbar şeklinde olabilir. Elektrik iç tesisat projelerinde çoğunlukla kablo kullanılmaktadır.

Şebekeden yapı bağlantı kutusunu besleyen bağlantıya *yapı bağlantı hattı* veya *yapı giriş hattı* denir. Yapı bağlantı kutusu ile binanın ilk panosu (ana dağıtım panosu, sayaç panosu) arasındaki bağlantı hattına *ana kolon hattı* denir. Binanın ilk panosundan diğer (kat dağıtım, daire dağıtım) panolarına kadar olan besleme hatlarına *kolon hattı* denir. Dağıtım panosundan aydınlatma veya prizlerin bağlandığı buata kadar olan kısma *linye hattı* denir. Buattan son malzemeye kadar olan kısma *sorti hattı* denir. Son malzeme, aydınlatma armatürü ise *ışık sorti* veya *aydınlatma sorti*, priz ise *priz sorti* olarak isimlendirilmektedirler.

Bir priz linyesine en fazla 7 priz sortisi bağlanabilir. Ancak çamaşır makinesi, bulaşık makinesi, fırın, şofben, kombi ve klima gibi sabit malzemeler için konulmuş priz linyelerine başka sortiler bağlanmaz. Aydınlatma linyelerine tavsiye olarak 9 aydınlatma sortisi bağlanabilir, fakat günümüz aydınlatma armatürlerinin tüketim güçleri çok düşük (5-15W) olduğu için gerilim düşümü hesaplamaları sonucuna göre bu sayı arttırılabilmektedir.

### 3.3. HESAPLAMALAR VE DÖKÜMLER

Elektrik tesisat planı, ağaç veri yapısı şeklindedir. Elektrik dağıtım ağacı, ana dağıtım ve sayaç panosundan daire dağıtım panolarına ve buradan da elektrik malzemelerine bağlantı şeklinde oluşmaktadır.

Elektrik dağıtım ağacı oluşturulduktan sonra her düğüm için kurulu güç, talep gücü ve talep akımı hesaplamaları yapılmaktadır. İkinci adımda bu hesaplamalara bağlı olarak kablo kesitleri ve devre koruma elemanları (şalter/sigorta) seçimleri yapılmaktadır. Üçüncü adımda bu hesaplamalar ve seçimlere göre gerilim düşümü ve kısa devre hesaplamaları [55–57] yapılarak seçilen elemanların uygunluk kontrolleri yapılmaktadır.

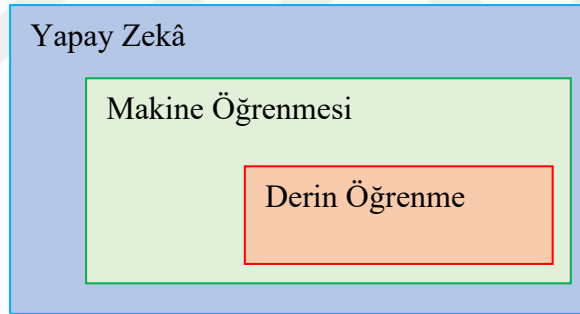
Projenin çizimi ve hesaplamaları sonucu projenin özeti olacak şekilde; aydınlatma tablosu, tablo yükleme cetveli, pano talep güçleri, gerilim düşümü, akım kontrolü, kısa devre hesapları, topraklama hesapları, kolon şeması, ana pano detayı ve kullanılan semboller tablosu, dökümler ve raporlamalar şeklinde elektrik projelerine eklenmektedir.

Akgül, yaptığı çalışmada kuvvetli akım elektrik tesisat planlarını ağaç veri yapısı şeklinde modelleyerek yapılması gereken tüm hesaplamaları ve dökümleri çizime entegre ederek otomatik olarak yapılmasını sağlamıştır [1,3].

## BÖLÜM 4

### MAKİNE ÖĞRENMESİ

Bilgisayarların insan davranışlarını taklit ederek zekâ gerektiren işleri gerçekleştirebilme becerilerine yapay zekâ denir [58]. Yapay zekanın kapsamı çok geniştir. Yapay zekânın alt konusu olan makine öğrenmesi, verilen bir problemi çözmek için probleme ait ortamdan elde edilen veriye göre model üreten algoritmalarının genel adıdır. Bir başka tanımla, yapay zekada karar verme adımlarına makine öğrenmesi denir. Yapay zekâ, makine öğrenmesi ve derin öğrenme kavramları birbirilerinin yerine çok kullanılmasına rağmen aralarındaki ilişki Şekil 4.1’de verilmiştir.



Şekil 4.1. Yapay zeka, makine öğrenmesi ve derin öğrenme arasındaki ilişki.

Makine öğrenmesi, çok yoğun çalışılan bir konu olduğundan günümüzde birçok yaklaşım ve algoritma önerilmiştir. Bunlar probleme yaklaşımlarına göre farklılık gösterebilir ve bu yüzden farklı problemlerde farklı başarılarla sahip olabilirler.

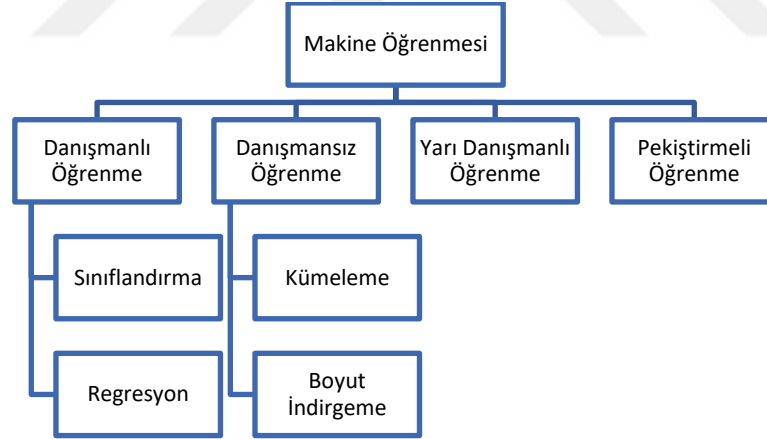
Makine öğrenmesi genel olarak danışmanlı, danışmansız, yarı danışmanlı ve pekiştirmeli (veya takviyeli) makine öğrenmesi olmak üzere dört ana grupta incelenmektedir. Şekil 4.2’de makine öğrenmesi yöntemleri genel hatlarıyla verilmiştir.

*Danışmanlı öğrenme*, bilinen girdi ve çıktılarla işaretlenmiş veriler üzerinde işlev üreten yöntemlerdir. Bu fonksiyonlar, eğri uydurma ve sınıflandırma gibi algoritmalar ile belirlenmektedir.

*Danışmansız öğrenmede*, işaretlenmemiş veriler aracılığıyla bilinmeyen bir yapıyı tahmin etmek için bir fonksiyon oluşturma beklenir. Verilerin mesafelere veya benzerliklere dayalı olarak kümelenmesi bu yapıya örnek olarak gösterilebilir.

İşaretlenmemiş çok fazla veriye karşılık işaretlenmiş az miktardaki veri ile eğitim yapılan yöntem *yarı danışmanlı öğrenme* denir. Yarı danışmanlı öğrenme danışmanlı ve danışmansız öğrenmenin arasında yer almaktadır.

*Pekiştirmeli öğrenmede* bir ceza-ödül puanı sistemi bulunmaktadır. İstenen amaca ulaşana kadar yapılan yanlışlardan çıkarımlar yapılarak öğrenme sağlanmaktadır. Bir bebeğin sıcak bir cisme dokunup tekrar dokunmaması gerektiğini öğrenmesi bu modele bir örnek olarak gösterilebilir.



Şekil 4.2. Genel hatlarıyla makine öğrenmesi yöntemleri.

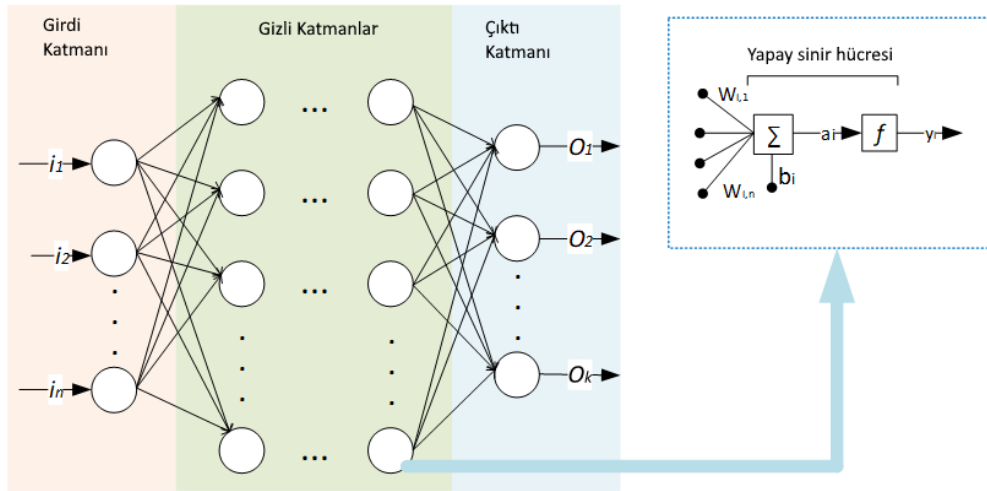
#### 4.1. YAPAY SİNİR AĞLARI

Makine öğrenmesinde yapay sinir ağları en çok kullanılan yöntemlerden biridir. Yapay Sinir Ağları (YSA) ilk olarak McCulloch-Pitts tarafından sinir sisteminden esinlenen sinirsel aktivitelerin çalışma prensibini mantıksal olarak hesaplayan matematiksel bir model olarak ortaya çıkarılmıştır [59]. 1990'lı yıllardan sonra YSA teknolojilerinde

oldukça hızlı gelişmeler görülmüştür. Yapay sinir ağları, geleneksel sinir ağları ve derin sinir ağları olmak üzere kabaca iki kısımda incelenebilir. Geleneksel yapay sinir ağlarında özellik çıkarımı kullanıcılara aittir. Özellik çıkarımı genellikle zor bir süreçtir ve çıkarılan özelliklerin verileri temsil etmesi beklenir. Çıkarılan özellikler sinir ağının eğitimi için kullanılacağından ağın başarısı doğrudan doğru özelliklerin çıkarılmasına bağlıdır [60].

YSA, insan beyninin biyolojik sinir yapısını taklit eden, doğrusal olmayan matematiksel modellerdir [61]. YSA'da verilen girdilere karşılık doğru sonuçların üretilmesi istenir. Ağın bunu yapabilmesi için önce sonucu bilinen örnekler ile eğitilir. Ağ istenen seviyede genelleme yapabilecek seviyeye ulaşıncaya kadar eğitim devam eder. Eğitim bitince ağ, sonucu bilinen bir kısım örnekler ile test edilerek doğruluk kontrolü yapılır. YSA'da veriler ağırlıklarda tutulmaktadır. Belirlenen öğrenme kuralı ve parametrelere göre sonuçları bilinen eğitim verileri için ağın ağırlık değerlerinin ayarlanmasına eğitim denir. Eğitimin tamamlanmasından sonra eğitilmiş olan ağ, ağırlık değerlerinin son durumuna göre daha önce ağa verilmemiş herhangi bir verinin sonucunu bir oran ile tahmin edebilmektedir.

Geleneksel YSA yapısı ve yapay sinir hücresi Şekil 4.3'de gösterilmiştir. Yapay sinir ağları, giriş katmanı, çıktı katmanı ve bir veya birden fazla gizli katmandan oluşmaktadırlar.



Şekil 4.3. Geleneksel yapay sinir ağı ve bir nöron yapısı.

Katmanlardaki her bir sinir hücresi; girdiler, ağırlıklar, toplam fonksiyonu, aktivasyon veya transfer fonksiyonu ve çıktı olmak üzere beş temel bileşenden oluşur.

Girdiler dış sistemden gelen veriler olabileceği gibi başka yapay sinir hücrelerinin çıktı değerleri de olabilir. Bir sinir hücresine birçok girdi verisi ulaşabilir. Her girdinin kendine bağlı bir ağırlık değeri vardır. Ağırlık, girdi değerinin toplam fonksiyonu üzerindeki etkisini belirler. Bu sayede bazı girdi değerleri, diğerlerinden daha önemli olabilir. Böylece sinir hücresi üzerinde daha fazla etkili olabilir. Bir yapay sinir hücresinde ilk işlem tüm girdilerin ağırlıklı toplamalarını bulmaktır.

Matematiksel olarak girdiler  $(x_1, x_2, \dots, x_n)$  ve karşılık gelen ağırlıklar  $(w_1, w_2, \dots, w_n)$  şeklinde gösterilebilen vektörlerdir. Toplam girdi sinyali genellikle bu iki vektörün iç çarpımı olarak hesaplanan bir değerdir. Toplam fonksiyonu, Eşitlik 4.1'de ve bir nöronun çıktısı Eşitlik 4.2'de gösterilmiştir. Burada  $y_i$  çıktıları,  $x_i$  girdileri,  $w_i$  ağırlıkları,  $b_i$  bias ve  $f$  aktivasyon fonksiyonunu ifade etmektedir.

$$NET_i = \sum_{i=1}^n x_i w_i \quad (4.1)$$

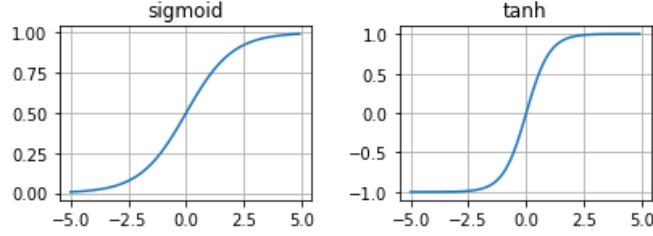
$$y_i = f(b_i + NET_i) \quad (4.2)$$

Toplam fonksiyonunun sonucu, aktivasyon fonksiyonuna aktararak hücrenin çıktı değeri üretilmektedir. Aktivasyon fonksiyonları probleme veya kullanılan öğrenme algoritmalarına göre değişiklik gösterebilir. Lineer, step, sinüs, eşik değer, sigmoid veya hiperbolik tanjant fonksiyonları olmak üzere değişik aktivasyon fonksiyonları bulunmaktadır. Tipik olarak bir nöronun çıktısı  $[0,1]$  veya  $[-1,1]$  aralığına normalleştirilir. En sık kullanılan aktivasyon fonksiyonları sigmoid ve hiperbolik tanjant fonksiyonlarıdır. Bunlar sırasıyla Eşitlik 4.3 ve Eşitlik 4.4'te gösterilmiştir.

$$\text{Sigmoid} \quad f(x) = \frac{1}{1 + e^{-x}} \quad (4.3)$$

$$\text{Tanh} \quad f(x) = \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4.4)$$

Şekil 4.4'te gösterildiği gibi sigmoid fonksiyonu toplam sonrası oluşan değeri [0,1], hiperbolik tanjant fonksiyonu ise [-1,1] aralığında normalleştirmektedir.



Şekil 4.4. Sigmoid ve Tanh fonksiyonları.

Aktivasyon fonksiyonunun ürettiği sonuç hücre çıktısı olarak kabul edilmektedir. Bu çıktı, hücrenin ağıdaki konumuna göre bağlı olduğu diğer hücelere girdi olarak iletilebilir veya ağı ürettiği sonuç olarak değerlendirilebilir.

Bir YSA modeli kurulduktan sonra istenen sonuçları vermesi için eğitim aşamasından geçer. Başlangıçta rastgele olarak verilmiş bias ve ağırlıkların, ağ istenen sonuçları verene kadar değiştirilmesine eğitim denir. Değiştirme işlemi, her adımda beklenen değerler ile ağı hesapladığı değerler arasında oluşan hata kareleri ortalamasının (Eşitlik 4.5) minimize edilmesi ile yapılır.

$$E = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.5)$$

Burada  $\hat{y}_i$  i'ninci çıkış nöronunun hesaplanan değeri,  $y_i$  ise beklenen değeri ifade etmektedir.

Yapay sinir ağının öğrenmesi geriye yayılım ile sağlanır. Elde edilen çıktılar ile gerçek sonuçlar arasındaki toplam hata, kabul edilebilir seviyeye gelene kadar eğitim devam eder. Eğitim bittiğinde YSA'daki ağırlıklar verilen girdiye göre en uygun sonucu üretmektedir.

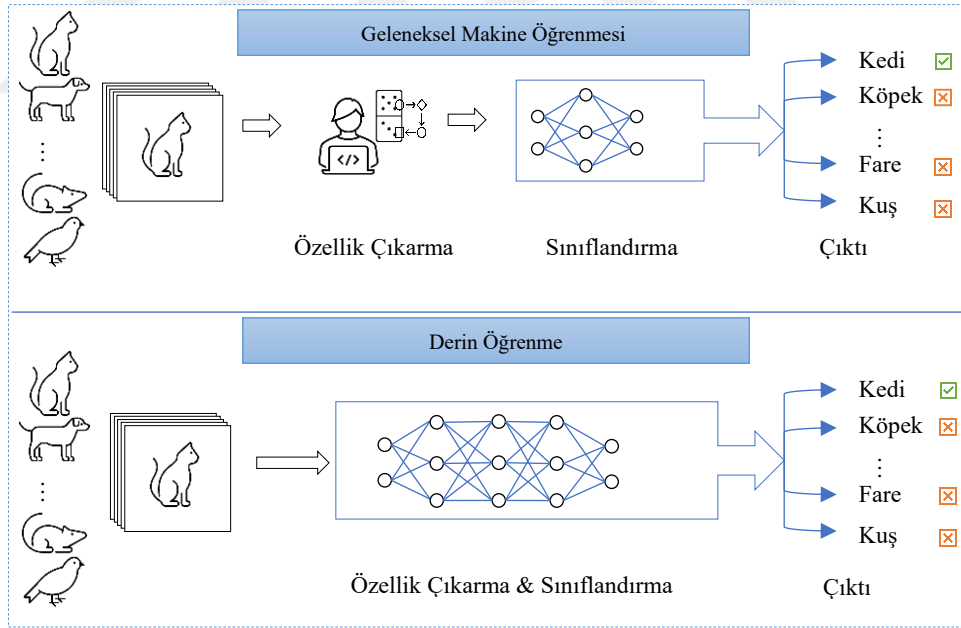
El yazısı tanıma [62,63], Optik karakter tanıma [64], imza tanıma [65,66], örüntü tanıma [67], ses tanıma [68,69], parmak izi tanıma [70,71], plaka tanıma [72], yüz

tanıma [73], resim veya videolardaki nesnelerin tespiti [74,75] ve benzeri birçok alanda geleneksel yapay sinir ağları kullanılmaktadır.

## 4.2. DERİN ÖĞRENME

Derin sinir ağları (Deep Neural Network) göreceli bir tanımdır. Önceleri YSA'nın katman sayısına göre kullanılan bir isimlendirme iken günümüzde katman sayısından ziyade otomatik özellik çıkarabilme özelliği için kullanılan bir isimlendirmedir.

Geleneksel makine öğrenmesi yöntemleri ile bir modelleme kurmak için ilk önce özellik vektörlerinin çıkarılması gerekmektedir. Özellik vektörlerinin çıkarılması için alanında uzman kişilere ihtiyaç duyulmaktadır. Bu işlemlerin zaman açısından maliyeti oldukça yüksektir. Geleneksel makine öğrenmesi teknikleri ile ham veriye ön işlem yapmadan istenen sonuç elde edilemez. Şekil 4.5'te geleneksel makine öğrenmesi ve derin öğrenme yaklaşımları şema üzerinde gösterilmiştir.



Şekil 4.5. Geleneksel makine öğrenmesi ve derin öğrenme.

Derin öğrenmenin gelişmesiyle birlikte makine öğrenmesi alanında uzun yıllar uğraşılan özellik çıkarma sorunu, ortadan kaldırılarak büyük gelişmeler sağlanmıştır. Geleneksel makine öğrenmesi tekniklerinde özellikler görüntü işleme teknikleri ile çıkarılırken derin öğrenme yaklaşımında bunun aksine öğrenme işlemi ham veri

üzerinde yapılmaktadır. Derin öğrenme, ham veriyi işlerken özellikleri farklı katmanlarda oluşturulan temsil öğrenme yöntemleri ile elde etmektedir. Temsil öğrenme, modelin ham verilerle beslenerek sınıflandırma işlemi için gereken özelliklerin otomatik olarak keşfedilmesine olanak tanıyan bir dizi yöntemdir [76].

Derin öğrenmede çok katmanlı hiyerarşik bir yapı söz konusudur. Bu katman hiyerarşisi sayesinde, üst seviye özellikler daha alt seviye özelliklerinden elde edilmektedir [77].

Derin öğrenmenin temeli, temsil öğrenme üzerine kurulmuştur. Örneğin; bir görüntü için temsil denildiğinde kenar şekilleri, özel şekiller, ortalama piksel yoğunluğu gibi özellikler düşünülebilir. Probleme göre bu özelliklerden bazıları veriyi daha iyi temsil edebilmektedir. Derin öğrenme yöntemleri bu aşamada avantaj olarak elle özellik çıkarmak yerine veriyi en iyi temsil eden hiyerarşik özelliklerin çıkarılması için etkin algoritmalar kullanmaktadır [78].

Derin öğrenme kavramı her ne kadar önceleri kullanılmış olsa da işlem maliyeti yüzünden pek tercih edilememiştir. Başarılı ilk derin sinir ağı çalışması Yann LeCun vd. tarafından 1998 yılında el yazısı ile yazılmış posta kodları tanımlanması üzerinde yapmıştır [79].

2000'li yıllardan sonra işlem hızlarının artması ve sonraki birkaç yılda GPU'ların hesaplamalarda kullanılması ile 10-15 yıllık sürede işlem hızlarının neredeyse binlerce kat artması dolayısıyla derin öğrenme, yapay zekâ alanında çok popüler olmaya başlamıştır.

Makine öğrenmesi alanındaki çalışmalar derin öğrenmeyle birlikte “*daha iyi özellik çıkarma*” işlemlerinden “*daha iyi özellik öğrenmek için daha iyi mimari tasarlama*” işlemine yönelmiştir. Araştırmacılar, önceleri problemin çözümü için veriyi temsil edecek en iyi özellikleri çıkarma ve bunlar içinde temsil kabiliyeti en iyi olanları seçme üzerine çalışmaktaydı. Derin öğrenme yaklaşımlarının gelişmesiyle birlikte özelliklerden ziyade artık çok katmanlı yapay sinir ağlarının nasıl tasarlanması gerektiği, kaç katmandan oluşacağı, her katmanda ne kadar nöron olacağı, hangi optimizasyon algoritmasının ya da aktivasyon fonksiyonunun kullanılması gerektiği

problem çözümünde daha önemli hale gelmiştir. Özellikle GPU'ların kullanımı ile işlem gücünün gelişmesi, araştırmacıları daha kompleks ağlar tasarlamaya yöneltmiştir.

Klasik sinir ağlarına nispeten derin öğrenme modelleri için çok daha fazla etiketlenmiş veriye ihtiyaç duyulmaktadır. Yüzbinler, milyonlar derin öğrenme için normal verilerdir. Günümüzde ulaşılabilir verilerin hızlı bir şekilde artması, problem çözümünde derin öğrenmenin rolünü oldukça önemli hale getirmiştir.

Derin öğrenme, günümüzde görüntü sınıflandırma [80], örüntü tanıma [81], e-ticaret web sayfalarında, sosyal ağlarda, film, müzik önerileri gibi içerik tabanlı önerilerde [77], doğal dil işleme [82], endoskopi görüntülerinden hastalık teşhisi [83], moda [84], oyunlar [85] ve daha birçok alanda kullanılmaktadır.

Derin öğrenmenin yakın gelecekte çok daha fazla başarıya ulaşacağı düşünülmektedir. Çünkü çok az manuel mühendislik işlemi gerektirmekte, böylece mevcut hesaplama ve veri miktarındaki artışlardan kolayca yararlanabilmektedir. Derin sinir ağları için şu anda geliştirilmekte olan yeni öğrenme algoritmaları ve mimarileri bu ilerlemeyi hızlandıracaktır [76].

#### **4.2.1. Evrimsel Sinir Ağları**

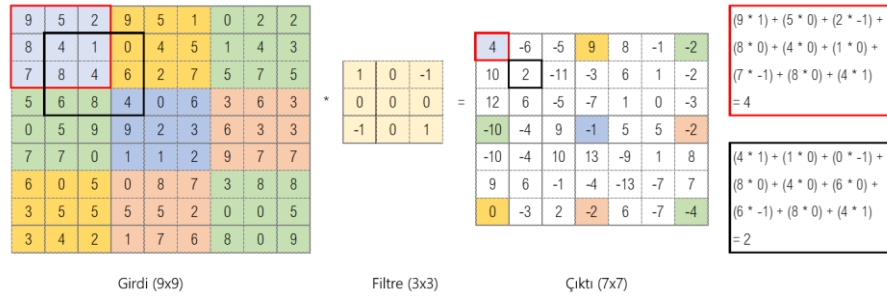
Evrimsel sinir ağları (ESA) derin öğrenmenin temel mimarisi olarak kabul edilmektedir. ESA, hayvanların görme merkezinden esinlenerek oluşturulmuş çok katmanlı ileri beslemeli bir YSA modelidir [79,86]. ESA, ilk olarak 1980'lerde duyurulmuş [86] ve AlexNet [80] modeli ile büyük ölçekli görüntü sınıflandırmasında dönemindeki mükemmel başarısı ile 2012'de popüler hale gelmiştir.

ESA'larda evrim ve havuzlama katmanlarında kullanılan filtreler ile özellik çıkarımı otomatikleştirilmektedir. ESA'larda toplam katman sayısı ağın derinliğini, evrim katmanlarındaki filtre sayıları ise ağın genişliğini ifade etmektedir.



oluşmaktadır. Dönüşüm işlemi  $n \times n$  (örneğin  $2 \times 2$ ,  $3 \times 3$ ,  $5 \times 5$ , vb.) gibi belirli boyuttaki filtrelerin (kernel) bütün görüntü üzerinde dolaştırılması ile yapılmaktadır. Filtreler önceden belirlenmiş olabileceği gibi rastgele de oluşturulabilir. Görüntü tanıma ve sınıflandırma uygulamalarında her bir filtre ile görüntüdeki yeni bir özellik öğrenilmiş olur. Bu yüzden ne kadar çok filtre varsa o kadar çok özellik öğrenilmiş olacaktır. Genellikle ilk katmanlarda büyük boyutlu filtreler, ilerleyen katmanlarda küçük boyutlu filtreler kullanılmaktadır. Dolayısıyla ilk katmanlarda görüntünün genel hatları, ilerleyen katmanlarda ise görüntünün detayları öğrenilmektedir [87]. Bununla birlikte katman sayısı arttıkça geri yayılım (back-propagation) etkisi zayıflayacağından bir noktadan sonra katman sayısını arttırmak öğrenme sürecine etki etmeyeceği gibi işlem süresini uzatacaktır. Bu yüzden katman ve filtre sayıları doğru ayarlanmalıdır.

$w \times h$  boyutunda bir görüntüye konvolüsyon işlemi uygulandığında elde edilecek çıktı görüntüsünün boyutu,  $(w - f_w + 1) \times (h - f_h + 1)$  olacaktır. Burada  $f_w$  filtre genişliği,  $f_h$  ise filtre yüksekliğidir. Örneğin  $32 \times 32$  piksel büyüklüğünde bir görüntüye  $5 \times 5$  boyutunda bir filtre ile konvolüsyon yapılırsa dönüşen yeni görüntü  $28 \times 28$  piksel olacaktır. Şekil 4.7’de örnek bir konvolüsyon işlemi gösterilmiştir.



Şekil 4.7. Örnek konvolüsyon işlemi.

### Düzeltilme Katmanı (Rectified Linear Unit – ReLU)

Konvolüsyon katmanında görüntüler dönüştürüldüğünde dönüşen görüntüler, orijinaleri ile doğrusal yapıda olmaktadır. Ağı, doğrusal olmayan bir yapıya sokmak için *ReLU* veya türevleri olan *ELU*, *LeakyReLU*, *AbsoluteReLU* gibi aktivasyon fonksiyonları kullanılmaktadır. Konvolüsyon katmanından hemen sonra kullanılan ve

aktivasyon katmanı olarak da bilinen bu katmanın amacı konvolüsyon işlemleri sonucu oluşmuş ve özellikle negatif değerleri sıfıra çekme (Eşitlik 4.6) veya olabildiğince sıfıra yaklaştırma (Eşitlik 4.7 veya Eşitlik 4.8) ya da mutlak değerini (Eşitlik 4.9) almaktır. Araştırmacılar, *ReLU* kullanarak ağın *Sigmoid* (Eşitlik 4.3) ve *Tanh* (Eşitlik 4.4) fonksiyonlarına göre daha hızlı öğrendiğini tespit etmişlerdir [88].

$$ReLU \quad f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (4.6)$$

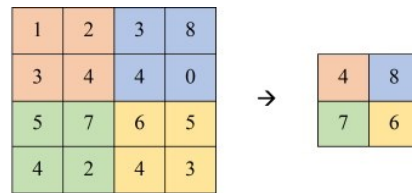
$$LeakyReLU \quad f(x) = \begin{cases} x, & x > 0 \\ 0.01x, & x \leq 0 \end{cases} \quad (4.7)$$

$$ELU \quad f(x) = \begin{cases} x, & x > 0 \\ e^x - 1, & x \leq 0 \end{cases} \quad (4.8)$$

$$AbsoluteReLU \quad f(x) = |x| \quad (4.9)$$

### Havuzlama Katmanı

Havuzlama veya ortaklama katmanı, *ReLU* katmanından hemen sonra kullanılır. Temel amacı, bir sonraki konvolüsyon katmanı için veri boyutunu azaltmaktır. Ortaklama ile görüntüden  $w \times h$  büyüklüğünde pencereler belirli bir adım ilerletilerek pencerenin içinde kalan en büyük değer (max-pooling) veya pencerenin ortalaması (average-pooling) alınarak yeni görüntü oluşturulur. Örneğin  $32 \times 32$  piksel boyutundaki bir görüntü üzerinde  $2 \times 2$  büyüklüğünde bir pencere ve 2 adım kaydırma ile hareket edilirse oluşacak yeni görüntü  $16 \times 16$  piksel büyüklüğünde olacaktır. Boyuttaki bu azalma, hem hesaplama yükü için hem de ağın ezberleme (overfitting) yapmaması için bir avantaj oluşturmaktadır. Şekil 4.8’de  $4 \times 4$ ’lük bir matris üzerinde  $2 \times 2$  filtre ve 2 adım kaydırma ile max-pooling uygulama örneği gösterilmiştir.



Şekil 4.8.  $2 \times 2$  filtre 2 adım kaydırma ile max-pooling.

## Tam Bağlı Katman

ESA'larda peş peşe gelen bir veya birden fazla evrişim + havuzlama katmanından sonra tam bağlı katman gelmektedir. Tam bağlı katman, kendinden önceki katmanın tüm nöronlarına tam bağlıdır. Tasarlanan modelin sınıflandırma hedefini optimize etmek için birden fazla tam bağlı katman kullanılabilir.

## Softmax (Sınıflandırma) Katmanı

Sınıflandırma için çıktıların belirli bir sınıfa ait olma olasılığını [0-1] aralığında değerler üreterek belirleyen katmandır. Softmax fonksiyonu Eşitlik 4.10'da verilmiştir. Softmax katmanındaki nöron sayısı ( $K$ ) sınıf etiketi sayısı kadardır.

$$S(y_i) = \frac{e^{y_i}}{\sum_{j=1}^K e^{y_j}}, \quad i = 1, \dots, K \quad (4.10)$$

### 4.2.1.2. ESA Parametreleri

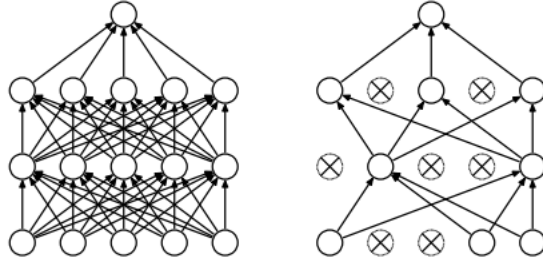
ESA'da sınıflandırmanın başarısı bazı parametrelere bağlıdır. Veri büyüklüğü ve çeşitliliği bu parametrelerin en önemlileridir. Bunların yanında ESA'da ezberlemeyi önlemek ve performansı artırmak için bazı ince ayarlar (Fine Tuning) yapılmaktadır. Konvolüsyon ve havuzlama (ortaklama) katmanlarındaki filtre (kernel) büyüklükleri ve adım sayıları, seçilen eğitim ve aktivasyon fonksiyonları, öğrenme oranı, dropout (seyreltme) oranı, eğitim tur sayısı (epoch), ağırlık başlangıç değerleri, katmanlardaki nöron sayıları ESA'da kullanılan parametrelerdir.

### Veri Büyüklüğü ve Çeşitliliği

ESA'da veri setinin büyük olması, hesaplama ve kaynak maliyetini artırsa da başarı için önemlidir. İyi bir model için veri setinin büyüklüğü tek başına yeterli değildir. Başarı için çeşitlilik de önemlidir. Bazı durumlarda veri seti yeteri kadar büyük değilse veri artırma (data augmentation) yöntemi ile veriler artırılmalıdır. Görüntüler döndürülerek, yansıtılarak veya bazı dezenformasyonlar uygulanarak veri artırımı yapılabilmektedir [81,89,90].

## Seyreltme (Dropout)

ESA'da büyük veriler ile eğitim yapıldığı için bazen ağ ezberleme (overfitting) yapabilmektedir. Bunun önüne geçmek için tam bağlı katmanda belirli bir oranda (genellikle 0.5 kullanılır) Şekil 4.9'de gösterildiği gibi rastgele bazı düğümler bağlantıları ile birlikte yok edilir [91,92].

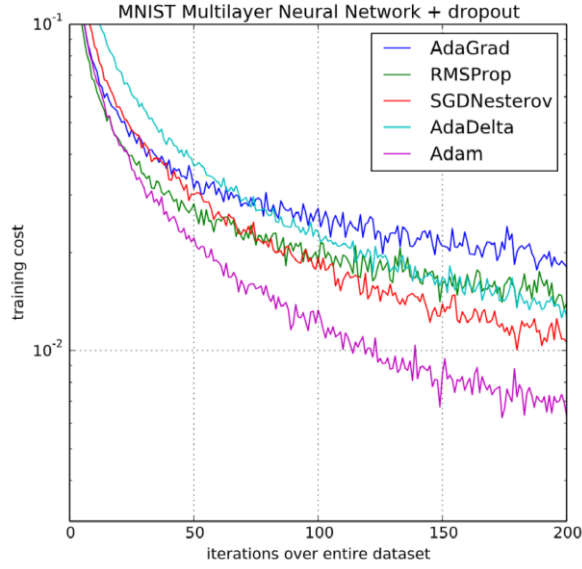


Şekil 4.9. Standart ve dropout uygulanmış sinir ağı [92].

Çok sayıda parametreye sahip derin sinir ağları çok güçlü makine öğrenme sistemleridir. Bu ağlarda en büyük problem aşırı uyumdur. Aynı zamanda büyük ağların kullanımı yavaştır. *Dropout* bu sorunu çözmek için ortaya konulmuş bir tekniktir. Temel fikir, eğitim sırasında sinir ağından bazı birimleri bağlantılarıyla birlikte rastgele yok etmektir. Böylece aşırı uyum, önemli ölçüde azaltılmış olur. Araştırmacılar danışmanlı öğrenmede, *dropout* kullanımı ile bilgisayarlı görü, konuşma tanıma, belge sınıflandırma gibi birçok görevde ağlarının performansının iyileştiğini ve birçok veri setinde daha iyi sonuçların alındığını göstermişlerdir [93].

## Optimizasyon Algoritması

YSA'da öğrenme, temelde bir optimizasyon problemidir. Derin öğrenme uygulamalarında yaygın olarak *stochastic gradient descent (SDG)*, *adagrad*, *adadelat*, *adam*, *adamax* gibi optimizasyon algoritmaları kullanılmaktadır. Bu algoritmalar arasında başarımlar ve hız bakımından farklılıklar bulunmaktadır [94]. Günümüzde en popüler yöntem öğrenme hızına uyum sağlayan *Adam* (Adaptive Moment estimation) metodudur [95,96]. Şekil 4.10'da optimizasyon algoritmalarının MNIST veri seti için kullanım karşılaştırmaları gösterilmiştir.



Şekil 4.10. Optimizasyon algoritmalarının karşılaştırılması [95].

### Öğrenme Oranı

Öğrenme oranı (Learning Rate), ağırlıkların hangi hızda güncelleneceğini göstermektedir. Öğrenme oranı, sabit veya uyarlanabilir şekilde ayarlanabilmektedir. Öğrenme oranı çok küçük seçilirse öğrenme hızı yavaş olacağından eğitim süresi çok uzun olacaktır. Eğer büyük seçilirse geniş atlamalar olabileceğinden öğrenme gerçekleşmeyebilir. Öğrenme oranının başlangıçta büyük tutulup sonraki adımlarda küçültülmesi tavsiye edilmektedir [97,98]. Adaptive algoritmalar öğrenme oranını dinamik olarak ayarlamaktadır. Eşitlik 4.11’de bir sonraki adımdaki ağırlık değerinin  $w(t + 1)$  mevcut durumdaki ağırlık değerine  $w(t)$  göre güncellenmesi gösterilmiştir. Burada  $\alpha$  öğrenme oranı,  $L$  ise kayıp fonksiyonudur.

$$w(t + 1) = w(t) - \alpha \frac{\partial L}{\partial w} \quad (4.11)$$

### Kayıp Fonksiyonu

Kayıp fonksiyonu (loss function), önerilen modelin nasıl performans gösterdiğini ölçmek için kullanılan ve olması gereken değerlere karşılık tahmin edilen değerlerin ne kadar farklı olduğunu gösteren bir fonksiyondur.

Makine öğrenmesinde, kayıp fonksiyonu için sıklıkla kullanılan *log-loss* ve *cross-entropi* aynı şeyleri ifade eden ve çok sınıflı sınıflandırma problemlerinde varsayılan olarak kullanılan kayıp fonksiyonudur [99]. Eşitlik 4.12’de *cross-entropi loss* fonksiyonu gösterilmiştir. Burada  $y$  gerçek çıktıları,  $z$  ise model çıktılarını ifade etmektedir.

$$L(z, y) = -[y \log(z) + (1 - y) \log(1 - z)] \quad (4.12)$$

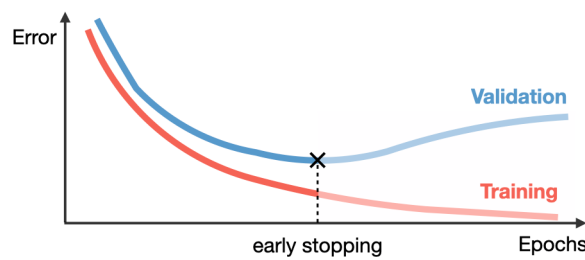
### Maliyet Fonksiyonu

Maliyet fonksiyonu (cost function), önerilen bir modelin performansını değerlendirmek için kullanılan bir fonksiyondur. Maliyet, tüm kayıpların toplamına eşittir ve Eşitlik 4.17’teki gibi tanımlanmaktadır.

$$J = \sum_{i=1}^n L(z_i, y_i) \quad (4.13)$$

### Eğitim adımı

Bir eğitim modelinde, modelin ağırlıklarını güncellemek için tüm eğitim setinin geçtiği her yinelemeye bir adım (epoch) denir. Eğitim aşamasında, doğrulama kaybı (validation loss) Şekil 4.11’de gösterildiği gibi düzleştiğinde veya artmaya başladığında eğitim için ayarlanan adım sayısı kadar beklemeden bitirilmelidir. Buna literatürde *early-stopping* denilmektedir. Bu noktadan sonra eğitim bitirilmezse ağ, ezberleme (overfitting) yapacaktır.



Şekil 4.11. Eğitimin erken bitirilmesi [99].

## Mini-Batch Büyüklüğü

YSA'da öğrenmenin her adımında ağırlık değerlerini güncellemek için ağı üzerinde gradyan hesaplaması ile geriye yayılım işlemi yapılmaktadır. Bu işlemlerde veri sayısı ne kadar fazla ise hesaplama da o oranda uzun sürmektedir. Veri setinde bulunan tüm verileri aynı anda işlemek zaman ve bellek açısından maliyetlidir. Özellikle bellek maliyetinin yetersiz kaldığı durumlarda veriler küçük kısımlara ayrılarak öğrenme işlemi daha küçük gruplar üzerinde gerçekleştirilmektedir. Bu şekilde birden fazla girdinin parçalar halinde işlenmesi *mini-batch* olarak adlandırılmaktadır. Mini-batch aynı anda modelin kaç veriyi işleyeceğini ifade etmektedir.

### 4.2.1.3. Eğitim ve Test Veri Setlerinin Ayarlanması

Bir ESA modelinin eğitilmesi için eğitim veri seti ve doğruluğu kontrol etmek için test veri seti olmak üzere iki grup veriden bahsedilir. Eğitim verisi arttıkça daha iyi öğrenme sağlanmaktadır. Test verisi arttıkça modelin hata yapma olasılığı daha iyi hesaplanabilmektedir. İyi bir model için eğitim ve test verileri birbirinden farklı olmalıdır. Bir model oluşturulurken veri setini, eğitim ve test veri seti olarak ikiye ayırmak için farklı yöntemler kullanılmaktadır.

## Hold-Out

Eğer kullanılacak veri seti yeterince büyük ise veri seti belli bir oranlarda eğitim ve test için ayrılır. Örneğin %80 eğitim için, %20 test için ayrılır. Test için ayrılan kısım kesinlikle eğitimde kullanılmaz. Eğitim sonucu, oluşan modelde eğitim setinde yüksek doğruluk oranına sahipken test için düşük doğruluk çıkıyorsa model, iyi öğrenememiş demektir. Bu şekilde oluşturulan modelde modelin başarımı test veri setinde oluşan doğruluk oranıdır.

## Çoklu Rastgele Örneklem

Eğer veri seti çok küçük ise test veri seti çok daha küçük olacağından sonuç güvenilir olmayabilir. Bu durumda  $n$  defa rastgele eğitim ve test veri setleri oluşturulur. Eğitim

ve test  $n$  defa yapılacağı için sonuç doğruluk değeri, her defasında hesaplanan doğruluk oranlarının ortalaması olacak şekilde hesaplanır.

### Çapraz Doğrulama

“Cross-validation” veya “k-fold cross validation” olarak isimlendirilen bu yöntemde veri seti rastgele  $k$  tane kümeye bölünür. Bu durumda eğitim  $k$  defa yapılır. Her eğitim döngüsünde  $(k-1)$  fold eğitim için, kalan bir fold test için ayrılır. İşlem sonucunda  $k$  tane doğruluk oranı ortaya çıkar. Bu durumda sonuç doğruluk değeri her fold için oluşan doğruluk değerlerinin ortalaması olarak hesaplanmaktadır [100].

#### 4.2.2. Sınıflandırma Ölçüm Metrikleri

Makine öğrenmesi alanında sınıflandırma problemleri, dünyanın en çok araştırılan alanlarından biridir. Konuşma/ses tanıma, yüz tanıma, yazı tanıma, nesne tanıma gibi neredeyse her alanda sınıflandırmalar yapılmaktadır. Sınıflandırma metrikleri, bir modelde sınıflandırmanın ne kadar iyi olduğunu gösteren performans değerlendirmesidir. Makine öğrenmesi alanında birçok performans ölçütü önerilmiş olsa da bir sınıflandırıcının performansını değerlendirmek için hangi metriğin seçileceğine dair genel bir kılavuz bulunmamaktadır [101].

##### 4.2.2.1. Doğruluk Oranı

Doğruluk oranı (accuracy) çoğunlukla temel değerlendirme ölçütü olarak kullanılmaktadır. Basit bir şekilde doğru tahmin edilen örnek sayısının toplam örnek sayısına oranını göstermektedir. Hesaplaması Eşitlik 4.14’da gösterilmiştir.

$$Accuracy = \frac{\text{doğru sınıflandırılan örnek sayısı}}{\text{veri setindeki toplam örnek sayısı}} \quad (4.14)$$

Doğruluk oranının dengesiz veri setlerinde kullanımı yanlıştır. Örneğin 990 örnekle A ve 10 örnekle B sınıflarından oluşan 1000 örnekle bir veri setinde, geliştirilen bir  $S(x)$  sınıflandırıcı tüm örnekleri A olarak sınıflandırırsa bile gerçekte hiçbir işlem yapılmadığı halde sınıflandırmanın doğruluk oranı  $990/1000 = \%99$  olacaktır.

*Doğruluk oranına* göre iyi bir başarı gibi görünmesine rağmen burada kullanılması yanlış bir yöntem olduğu için  $S(x)$  aslında iyi bir sınıflandırıcı değildir.

#### 4.2.2.2. Karışıklık Matrisi

Sınıflandırma sonucu örneklerin *tahmin edilen* ile *gerçek* sınıflarını karıştırıp karıştırmadığını göstermek için kullanılan karışıklık matrisi (confusion matrix), hataların oranını gösteren ve aynı zamanda hata matrisi olarak da bilinen bir çizelgedir [101,102]. Çizelge 4.1’de ikili bir sınıflandırıcı için karışıklık matrisi verilmiştir.

Çizelge 4.1. İkili sınıflandırma için karışıklık matrisi.

		Gerçek Sınıflar	
		Pozitif	Negatif
Tahmin Edilen Sınıflar	Pozitif	TP	FP
	Negatif	FN	TN

İkili sınıflandırmada genellikle *pozitif* ve *negatif* şekilde isimlendirilmek üzere iki sınıf vardır. Örneğin kanserli ve normal hücrelerin bulunduğu bir veri setinde kanserli hücreler *pozitif*, normal hücreler *negatif* olsun.

- **True Positive (TP)**: Eğer pozitif bir örnek, pozitif olarak sınıflandırılırsa buna *True Positive* denir. Örneğin, sınıflandırıcının kanserli bir hücreyi kanserli olarak sınıflandırdığı durum. (Doğru sınıflandırma)
- **True Negative (TN)**: Eğer negatif bir örnek, negatif olarak sınıflandırılırsa buna *True Negative* denir. Örneğin, sınıflandırıcının normal bir hücreyi normal olarak sınıflandırdığı durum. (Doğru sınıflandırma)
- **False Positive (FP)** : Eğer negatif bir örnek, yanlışlıkla pozitif olarak sınıflandırılırsa buna *False Positive* denir. Örneğin, sınıflandırıcının normal bir hücreyi kanserli olarak sınıflandırdığı durum. (Yanlış sınıflandırma)
- **False Negative (FN)**: Eğer pozitif bir örnek, yanlışlıkla negatif olarak sınıflandırılırsa buna *False Negative* denir. Örneğin, sınıflandırıcının kanserli bir hücreyi normal olarak değerlendirdiği durum. (Yanlış sınıflandırma)

### 4.2.2.3. F1-puanı

*Accuracy* (doğruluk), bir modelin başarısını ölçmek için en çok kullanılan ancak özellikle dengesiz veri setlerinde tek başına işe yaramayan bir metriktir (Eşitlik 4.15).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.15)$$

*Accuracy*, güvenilirmez olduğu için yerine *F1-puanı* (*F1-score*) kullanılmaktadır. *F1-puanı*, bir sınıflandırıcının başarısını değerlendirmek için yaygın olarak kullanılan bir kriterdir. *F1-puanı* hesaplamak için önce kesinlik (*precision*) ve duyarlılık (*recall*) değerlerinin hesaplanması gerekmektedir.

*Kesinlik* (*precision*), pozitif olarak işaretlenmiş örneklerin gerçekte de pozitif olanlarıdır ve matematiksel gösterimi Eşitlik 4.16'da gösterilmiştir.

$$Precision = \frac{TP}{TP + FP} \quad (4.16)$$

*Duyarlılık* (*recall*), gerçek pozitif örneklerin pozitif olarak işaretlenen kısmıdır ve matematiksel gösterimi Eşitlik 4.17'de gösterilmiştir.

$$Recall = \frac{TP}{TP + FN} \quad (4.17)$$

*F1-puanı*, *kesinlik* ve *duyarlılık* değerlerinin harmonik ortalamasıdır ve Eşitlik 4.18'te gösterildiği gibi hesaplanmaktadır.

$$F_1score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4.18)$$

İkili sınıflandırmalardan farklı olarak çok sınıflı sınıflandırıcılarda *F1-puanı* her sınıf için ayrı ayrı hesaplanmaktadır. Çizelge 4.2’de çok sınıflı örnek bir karışıklık matrisi verilmiştir.

Çizelge 4.2. Çok sınıflı sınıflandırıcı için karışıklık matrisi.

		Gerçek Sınıflar			
		C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>
Tahmin Edilen Sınıflar	C <sub>1</sub>	TP	FP		
	C <sub>2</sub>	FN	TN		
	C <sub>3</sub>				
	C <sub>4</sub>				

*F1-puanı*, kesinlik (*precision*) ve duyarlılık (*recall*) değerlerinin harmonik ortalaması olduğuna göre her sınıf için bu değerlerin ayrı ayrı hesaplanması gerekmektedir. Bunları hesaplamak için her bir sınıf için ayrı ayrı *TP*, *FP* ve *FN* değerlerinin hesaplanması gerekmektedir. Çizelge 4.2’de gösterildiği gibi, *C<sub>1</sub>* sınıfında olmayıp yanlış olarak *C<sub>1</sub>* olarak tahmin edilen tüm örnekler hangi sınıfta olursa olsun *C<sub>1</sub>* için *FP*’dir. Benzer şekilde *C<sub>1</sub>* sınıfında olduğu halde yanlış olarak hangi sınıfta sınıflandırılmış olursa olsun *C<sub>1</sub>* için *FN*’dir. Bu şekilde her bir sınıf için *F1-score* hesaplanır. Sınıflandırıcının genel *F1-score* değeri *F1-macro-score* ve *F1-micro-score* olmak üzere iki farklı şekilde hesaplanır. *F1-macro-score* hesaplamak için tüm sınıfların ayrı ayrı hesaplanmış *F1-score* değerlerinin aritmetik ortalaması hesaplanır (Eşitlik 4.19). Çok sınıflı sınıflandırıcılarda genellikle başarımlı ölçütü olarak *F1-macro-score* değeri kullanılır.

$$F_1^{macro} = \frac{1}{n} \sum_{class=1}^n F_1(class) \quad (4.19)$$

*F1-micro-score* hesaplamak için tüm sınıflar için hesaplanmış *micro-averaged-precision* ( $Precision^{micro}$ ) ve *micro-averaged-recall* ( $Recall^{micro}$ ) değerleri sırasıyla Eşitlik 4.20 ve Eşitlik 4.21’de gösterildiği gibi hesaplanır. Daha sonra *F1-micro* Eşitlik 4.22’de gösterildiği gibi  $Precision^{micro}$  ve  $Recall^{micro}$  değerlerinin harmonik ortalaması olarak hesaplanır.

$$Precision^{micro} = \frac{\sum_i^n TP_i}{\sum_i^n TP_i + \sum_i^n FP_i} \quad (4.20)$$

$$Recall^{micro} = \frac{\sum_i^n TP_i}{\sum_i^n TP_i + \sum_i^n FN_i} \quad (4.21)$$

$$F_1^{micro} = \frac{2 * Precision^{micro} * Recall^{micro}}{Precision^{micro} + Recall^{micro}} \quad (4.22)$$

Çok sınıflı sınıflandırıcılarda *micro-averaged-precision* ve *micro-averaged-recall* her zaman eşittir. Bu yüzden ikisinin harmonik ortalamaları da kendilerine eşit olur. Dahası *F1-micro-score* her zaman sınıflandırıcının *accuracy* (doğruluk) değerine eşit olur.

### 4.2.3. Öğrenme Kontrolü

Oluşturulan model, eğitim setinde yeterince düşük bir hata değeri elde edemiyorsa yetersiz/eksik öğrenme (underfitting) oluşmuş demektir. Eğer eğitim hatası ile test hatası arasındaki fark çok büyük ise bu durumda aşırı öğrenme/ezberleme (overfitting) oluşmuş demektir.

*Eksik öğrenme*, sinir ağlarında eğitim setlerinin özelliklerini yakalayamadığında ve hedef eşlemelere iyi uyamadığında oluşmuş demektir. Eksik öğrenme, düşük doğruluk oranlarına ve yüksek kayıp değerlerine yol açmaktadır. Buna sebep olan olası faktörler, tasarlanan ağ mimarilerinin çok basit olması ve çok az gizli katmana veya eğitilebilir parametreye sahip olması, dolayısıyla karmaşık özelliklerin yakalanması için yeterince güçlü olamamasıdır. Bir başka faktör ise ağın öğrenecek kadar eğtilememiş olmasıdır [93,103].

Model aşırı derecede eğitim veri setine bağımlı olup, doğrulama ve test veri setlerinde çok düşük sonuç veriyorsa *aşırı öğrenme* meydana gelmiş demektir. Bunun olası sebepleri, tasarlanan ağ mimarisinin kompleks olmasına rağmen yeteri kadar eğitim verisi olmamasıdır. Bu durumda ağ, belirli örneklere çok iyi odaklanır ancak ortak

özellikler görmezden gelinir. Bir başka faktör ise eğitim ve test verilerinin birbirinden çok farklı olduğu durumlarda aşırı öğrenme oluşabilir [92,93].

Makine öğrenmesinde eksik öğrenme, olması gereken ve aşırı öğrenme durumları ve muhtemel çözüm önerileri Çizelge 4.3'te karşılaştırmalı olarak verilmiştir.

Çizelge 4.3. Eksik öğrenme, olması gereken ve aşırı öğrenme [99].

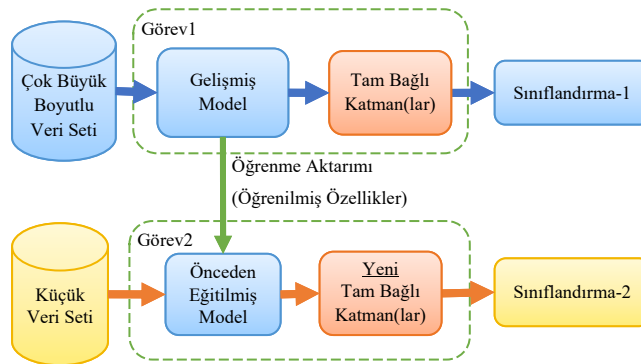
	Yetersiz/Eksik Öğrenme	Olmaması Gereken	Aşırı Öğrenme/ Ezberleme
<b>Belirtiler</b>	<ul style="list-style-type: none"> <li>Eğitim hatası yüksek,</li> <li>Eğitim hatası ile test hatası birbirine yakın,</li> <li>Yüksek bias (verilen veri noktaları için beklenen tahmin ile tahmin etmeye çalıştığımız doğru model arasındaki fark)</li> </ul>	<ul style="list-style-type: none"> <li>Eğitim hatası, test hatasından birazcık daha düşük.</li> </ul>	<ul style="list-style-type: none"> <li>Çok düşük eğitim hatası,</li> <li>Eğitim hatası, test hatasından çok daha düşük,</li> <li>Yüksek varyans (verilen veri noktaları için model tahmininin değişkenliği)</li> </ul>
<b>Eğri Uydurma Gösterimi</b>			
<b>Sınıflandırma Gösterimi</b>			
<b>Derin Öğrenme Gösterimi</b>			
<b>Muhtemel Çözümler</b>	<ul style="list-style-type: none"> <li>Modeli geliştir,</li> <li>Daha fazla özellik ekle,</li> <li>Daha uzun süreli eğitim yap.</li> </ul>		<ul style="list-style-type: none"> <li>Model karmaşıklığını azalt,</li> <li>Veri setini büyüt,</li> <li>Eğitim süresine dikkat et.</li> </ul>

### 4.3. TRANSFER ÖĞRENME

Transfer öğrenme (TL-Transfer Learning), makine öğrenmesi sürecinde bir görevden kazanılan bilgiyi benzer başka bir görevde kullanmayı amaçlayan bir yaklaşımdır. Klasik makine öğrenmesi tekniklerinde, sınıflandırma için manuel olarak çıkarılan özellikler kullanılmaktadır. Ama ESA'da sınıflandırma, evrişim ve havuzlama katmanları aracılığıyla otomatik çıkarılan özellikler ile yapılmaktadır. Öğrenme için otomatik çıkarılan özellikler, eğitim veri seti boyutuna büyük ölçüde bağımlıdır. Eğitim verilerinin miktarı yeterince büyük değilse aşırı öğrenme (overfitting) oluşabilmektedir. Bunu önlemek için transfer öğrenme yaygın olarak kullanılmaktadır [104].

ESA için en yaygın sorun, araştırmacıların genellikle büyük bir veri seti oluşturmak için yeterli veri bulamamasıdır. Transfer öğrenme, özellikle eğitim verilerinin toplanmasının maliyetli veya zor olduğu durumlarda kullanılan iyi bir yöntemdir [105]. Ayrıca zamandan tasarruf etmek ve daha az eğitim süresi ile daha iyi başarımlar elde etmek için kullanılmaktadır. Derin bir ESA modelini sıfırdan eğitmek, büyük miktarda etiketlenmiş veri ve büyük miktarda işlem kaynağı gerektirmektedir. Ancak transfer öğrenme yöntemiyle önceden eğitilmiş bir ESA modeli kullanılarak kısa yoldan bir eğitim süreci gerçekleştirilmiş olur.

Deneyle, transfer öğrenmenin, rastgele ağırlık dağılımına kıyasla ESA'da daha iyi doğruluk sonuçları verdiğini göstermektedir [106]. Şekil 4.12'de transfer öğrenme modeli diyagram üzerinde gösterilmiştir.



Şekil 4.12 Transfer öğrenme diyagram gösterimi.

Transfer öğrenmeli bir ESA modeli oluşturup kullanmak için önceden benzer alanda eğitilmiş bir ESA bulup onun parametrelerinin aktarılması gerekmektedir. Bunu yapmak için;

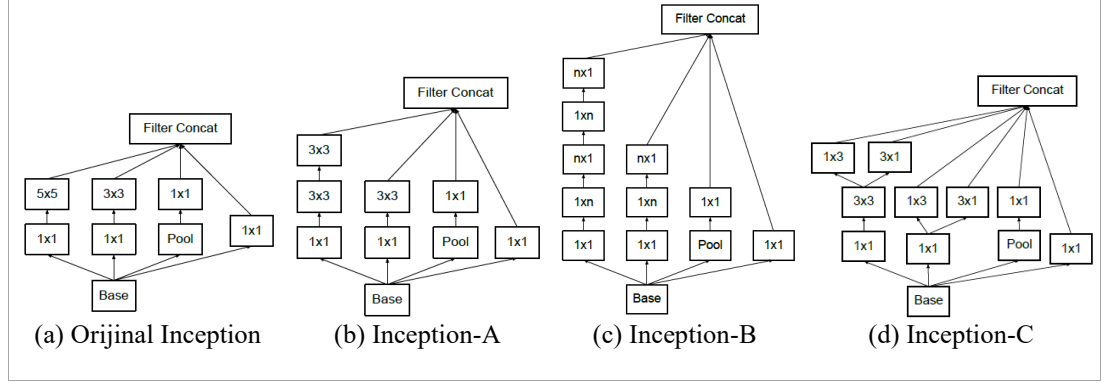
- Ağ, ImageNet gibi büyük miktarda etiketlenmiş veri ile kaynak görev üzerinde eğitilir.
- Ardından, evrişim ve havuzlama katmanları dondurulur.
- Sondaki tam bağlı katmanlar ve sınıflandırıcı kaldırılır.
- Bu önceden eğitilmiş modeli yeni görevde kullanmak için bir veya daha fazla yeni tam bağlı katman –adaptasyon katman(lar)– eklenir.
- Hedef veri seti ile eşleşen yeni bir sınıflandırıcı eklenir.
- Son olarak ağ, tam bağlı yeni katmanların ağırlıklarının güncellenmesi için yeniden eğitilir.

Transfer öğrenme, günümüzde birçok alanda özellikle görüntü sınıflandırma işlemlerinde ve özellikle ImageNet veri setinden elde edilen öğrenme aktarımı ile çok yaygın bir şekilde kullanılmaktadır. ImageNet, nesne algılama ve sınıflandırma araştırmaları için tasarlanmış yaklaşık 22.000 kategoride 14 milyondan fazla yüksek çözünürlüklü veri bulunduran çok büyük bir veri setidir. ILSVRC (The ImageNet Large Scale Visual Recognition Challenge) yarışması 2010 yılından beri nesne algılama ve sınıflandırma alanında standart bir kıyaslama veri seti olmuştur. ImageNet yarışmalarında 1000 kategoride eğitim için 1.2 milyon, test için 100.000 ve doğrulama için 50.000 görüntü kullanılmaktadır [107].

Son birkaç yılda, çok başarılı derin öğrenme modelleri yayınlanmıştır. Inception-v3 [108], ResNet [109,110], Inception-v4/Inception-ResNet [111], ResNeXt [112], Xception [113], DenseNet [114], MobileNet [115–118], VGGNet [119] ve bunların geliştirilmiş sürümleri gibi birçok profesyonel derin ESA mimarilerinin farklı platformlarda transfer öğrenme için kullanılacak modelleri araştırmacılar için mevcuttur. Bu modellerin her biri, özellikle ImageNet sınıflandırma yarışlarında kendi başarılarını kanıtlamışlardır. Bu çalışmada Inception-v3 [108] mimarisinin ImageNet veri seti üzerinde öğrendiği özelliklerin aktarımı ile tefriş sembollerinin sınıflandırılması yapılmıştır.



farklı şekilde yeniden şekillendirilmiştir. Orijinal inception ve yeniden şekillendirilmiş üç inception modülleri Şekil 4.15'te gösterilmiştir.



Şekil 4.15. Orijinal ve yeniden şekillendirilmiş inception modülleri.

Inception-A modülünde, her  $5 \times 5$  evrişim filtresi, iki adet  $3 \times 3$  evrişim filtresi ile değiştirilmiştir. Bu, parametre sayısında  $5 \times 5 = 25$ 'i  $2 \times (3 \times 3) = 18$ 'e düşürmektedir. Inception-B modülünde,  $n \times n$  filtre boyutunun evrişim filtrelerini  $1 \times n$  ve  $n \times 1$  kombinasyonuna ayırmaktadır. Inception-C modülünde, her  $3 \times 3$  evrişim filtresi yerini  $1 \times 3$  ve  $3 \times 1$  evrişim filtreleri almıştır. Bu da parametre sayısında  $3 \times 3 = 9$ 'u  $1 \times 3 + 3 \times 1 = 6$ 'ya düşürmüştür. Bu şekilde ağın derinliği artmasına rağmen parametre miktarı azaltılmıştır. Inception-v3 modeli ile ImageNet sınıflandırmada top-5 hata oranı %3.58'e düşürülmüştür.

#### 4.4. NESNE ALGILAMA VE SINIFLANDIRMA

Bilgisayarlı görü, son yıllarda birçok alanda ilgilenilen disiplinler arası bir konu olmuştur. Bilgisayarlı görünün en temel amacı nesne tanıma iken bir diğer önemli amacı da nesne algılamadır. Nesne algılama algoritmaları, bir görüntüdeki her bir nesneyi bir çerçeve içine almak ile ilgilenmektedir. Görüntüde birden fazla nesne varsa birden fazla çerçeve oluşması gerekecektir. Nesne algılama, otonom araçlar, kamera ile gözetleme, pozlama gibi özellikle gerçek zamanlı görüntülerde nesne tespiti ve sınıflandırmanın sıcak konusudur.

Çeşitli bilgisayarlı görü projeleri için COCO (Common Objects in Context) veri seti yaygın olarak kullanılmaktadır. 80 kategoride 1.5 milyon nesne içeren COCO veri seti,

Microsoft tarafından yayınlanan büyük ölçekli bir nesne algılama, segmentasyon ve etiketleme veri setidir [122].

Standart bir ESA mimarisi ile birden fazla nesnenin olduğu bir görüntü doğru bir şekilde sınıflandırılmaz. Bunun yerine görüntüdeki farklı alanların bulunup sınıflandırılması gerekmektedir. Bir görüntüdeki nesnelerin konumları ve ebatları farklı olabilir. Dolayısıyla bir görüntüden çok fazla bölüm elde edilebilir.

Ross Girshick vd. bir görüntüden seçici arama algoritması [123] ile sadece 2000 bölgeyi alıp sınıflandıran ve R-CNN (Regions with CNN) ismini verdiği bir yöntem önermişlerdir [124]. R-CNN (bölgesel evrimsel sinir ağı) yapısındaki temel mantık, ilk önce görüntüyü seçici arama ile 2000 bölgeye ayırır ve her bölgeyi ayrı ayrı evrimsel sinir ağından geçirir. Bu yöntemin en büyük problemi zamandır. Zaman problemini aşmak için R-CNN'nin türevleri farklı teknik ve süreçlerden geçilerek Fast R-CNN [125], Faster R-CNN [126] ve Mask R-CNN [127] mimarileri geliştirilmiştir. Geliştirilen bu modeller ile gerçek zamanlı görüntülerde ve videolarda nesne tespiti yapıp sınıflandırabilecek hıza ulaşılmıştır.

Nesne algılamada farklı bir yaklaşım olan YOLO (You Only Look Once) ile görüntüyü parçalara bölüp sınıflandırmak yerine tüm görüntüden nesnelere direkt olarak bulabilen daha hızlı bir model geliştirilmiştir [128–131]. YOLO algoritmasının R-CNN sürümlerinden daha hızlı olmasının sebebi, görüntüyü  $n \times n$ 'lik gridlere bölerek her grid içinde nesne olup olmadığını ölçen bir güven skoru üretmesidir. Böylece sadece nesne olduğu düşünülen gridler, ESA ağına verilerek nesne tespiti yapılmaktadır.

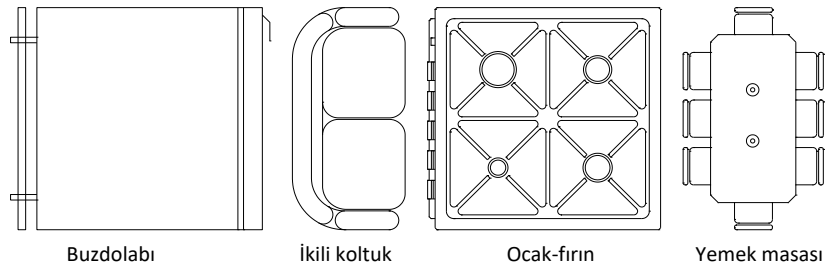
## BÖLÜM 5

### DERİN ÖĞRENME İLE TEFRİŞ SINIFLANDIRMA

Mimari kat planlarda tefriş sembolleri, buzdolabı, televizyon, ocak-fırın, çamaşır makinesi, bulaşık makinesi; koltuklar, masalar, yataklar, dolaplar, kapılar ve pencereler gibi malzemelerin çizim sembollerine verilen genel isimlerdir. Tefriş sembolleri, çizilen mimari projelerde oluşturulan odaların kullanım sırasında gerekli olan eşyaların yerleşip yerleşmediğini göstermesi bakımından önemlidir.

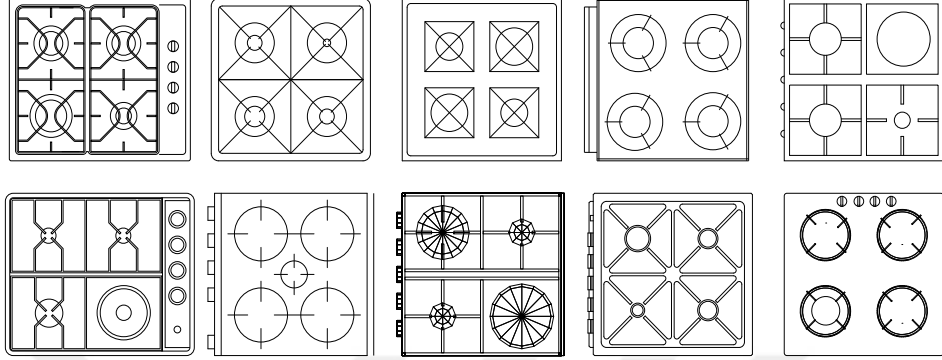
Tefriş sembolleri, vektörel çizim ortamında blok şeklinde tanımlanmaktadır. Bloklar, vektörel çizim ortamında birden fazla geometrik nesnenin farklı açılar, ölçekler ve konumlarda tekrardan kullanabilmek için bir araya getirilmiş çizim gruplarıdır. Projeciler, farklı projelerde tekrar eden nesne sembollerini (pencereler, kapılar, dolaplar, koltuklar, banyo ve mutfak tefrişleri, vb.) blok şeklinde birleştirerek oluşturdukları blok kütüphanelerinde saklarlar. Böylece yeniden çizilmesine gerek kalmadan istedikleri zaman istedikleri projede kullanabilmektedirler.

Tefriş blokları, mimara, mimariye ve kullanıldığı plana göre farklılık gösterse de kat planlarında genellikle temsil ettikleri malzemenin üstten görünüşü şeklinde hazırlanmaktadır. Şekil 5.1’de kat planlarında kullanılan örnek bazı tefriş sembolleri gösterilmiştir.



Şekil 5.1. Örnek tefriş sembolleri.

Mimarlar tarafından kullanılan her tefriş sınıfı için yüzlerce farklı sembol bulunabilir. Bu semboller genellikle mimara ve/veya kültüre özgüdür. Örneğin, Şekil 5.2’de bir ocak-fırın sembolü için on farklı şekil gösterilmiştir.



Şekil 5.2. Mimarlar tarafından kullanılabilen farklı ocak-fırın sembolleri.

Bu çalışmanın omurgasını tefriş sınıflandırma oluşturmaktadır. Çünkü diğer tüm işlemler tefriş tanımaya göre yapılmaktadır.

## 5.1. VERİ SETİ OLUŞTURMA

Veri seti oluşturmak için onlarca farklı konut binası kat planı çizimlerinden 23 farklı sınıfa ait 1020 adet tefriş sembolü bloğu toplanmıştır. Sınıf isimleri, her sınıftan örnek bir görsel ve her sınıftaki örnek sayıları Çizelge 5.1’de verilmiştir.

Toplanan bu semboller daha sonra, araştırmacılar tarafından kamuya açık olarak kullanılmak ve/veya genişletilmek üzere bir DXF dosyasına kaydedilmiştir. DXF, özellikleri Autodesk firması tarafından belirlenen ve CAD programları arasında çizim verilerini aktarmak için kullanılan, sayısal veya metin tabanlı olabilen bir dosya formatıdır [132].

CAD dosyalarındaki semboller, vektör formatında olduğundan oluşturulan ESA modelinde kullanılmak üzere belirli çözünürlüklerde resim formatına dönüştürülmüştür. Hangi çözünürlüğün daha iyi sonuç vereceğini araştırmak için farklı çözünürlük senaryoları denemek gerekmektedir. Vektörel verilerde ölçeklerin büyütülüp küçültülmesi kenar çizgilerinin kalınlıklarını değiştirmez. Ancak piksel tabanlı resimler için aynısını söylemek mümkün değildir. Piksel tabanlı resim

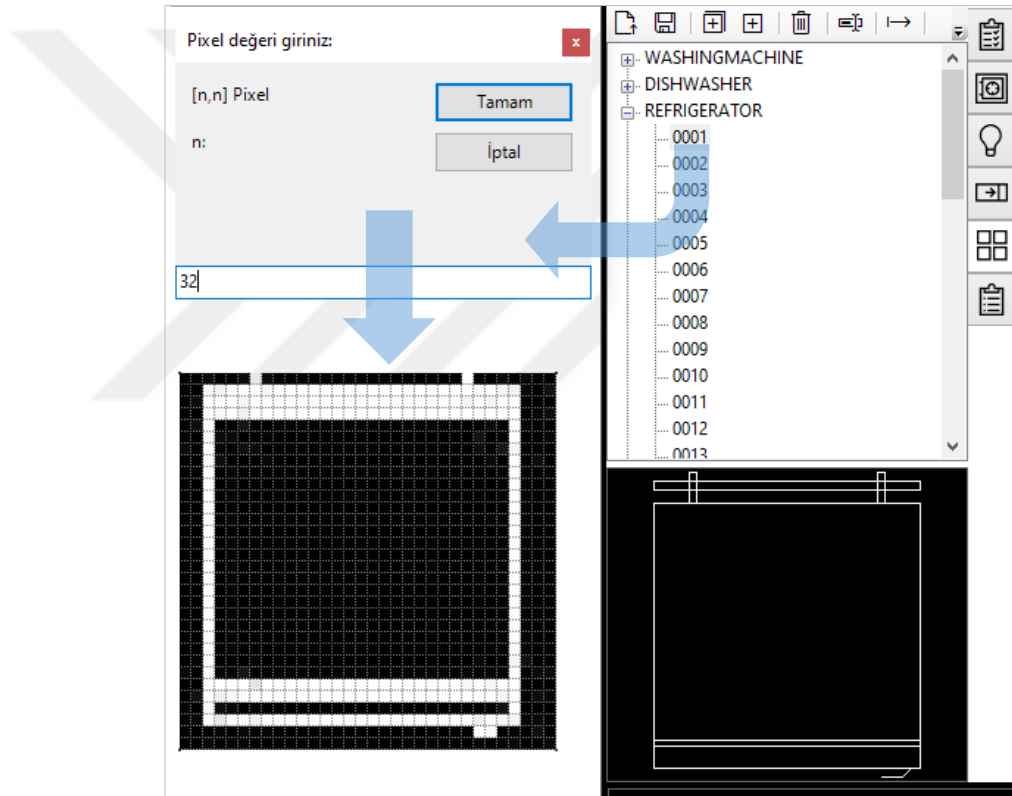
dosyalarında ölçek deęiştirilince kenarlarda kalınlaşma veya incelme olabilir. Hatta eęer resim çok küçültülürse örneğin orijinal resimde 1-piksel kalınlığındaki bazı çizgiler kaybolabilir. Bunu önlemek için tefriş blokları, vektörel formatta istenilen ölçeklere getirdikten sonra resim formatına dönüştürülmüştür. Bu şekilde resim çözünürlüğü ne olursa olsun, çizgi kalınlıkları 1-piksel olmuştur.

Çizelge 5.1. Veri setindeki tefriş sınıfları.

No	Tefriş Sınıfı	Örnek Sembol	Örnek Sayısı	Açıklama	No	Tefriş Sınıfı	Örnek Sembol	Örnek Sayısı	Açıklama
1	BED-DOUBLE		65	Çift kişilik yatak	13	SOFA-TWO		36	İkili koltuk
2	BED-SINGLE		45	Tek kişilik yatak	14	STOVE-OVEN		36	Ocak-Fırın
3	DISHWASHER		30	Bulaşık makinesi	15	TABLE-DINNER		111	Yemek masası
4	DOOR-DOUBLE		38	İki kanatlı kapı	16	TABLE-STUDY		30	Çalışma masası
5	DOOR-SINGLE		68	Tek kanatlı kapı	17	TELEVISION		13	Televizyon
6	DOOR-WINDOWED		19	Pencereli kapı	18	TOILET		67	Tuvalet
7	REFRIGERATOR		19	Buzdolabı	19	WARDROBE		54	Gardırop
8	SHOWER		60	Duş kabini	20	WASHBASIN		64	Lavabo
9	SINK		38	Evye	21	WASHBASIN-CABINET		36	Dolaplı Lavabo
10	SOFA-CORNER		17	Köşe takımı koltuk	22	WASHINGMACHINE		30	Çamaşır makinesi
11	SOFA-ONE		52	Tekli koltuk	23	WINDOW		50	Pencere
12	SOFA-THREE		42	Üçlü koltuk					

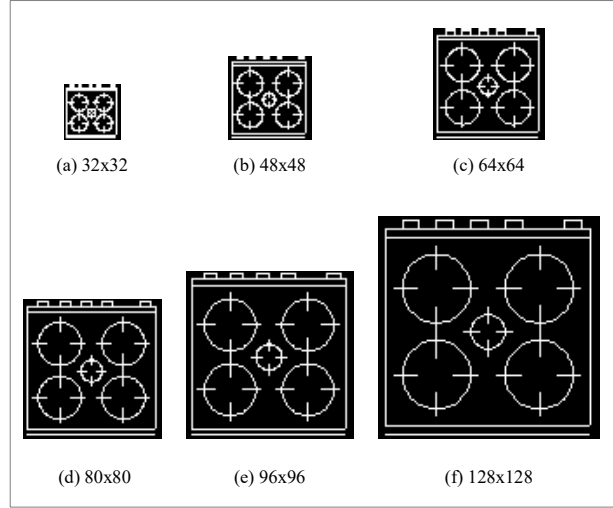
Vektörel çizimlerde nesnelerin konulduğu koordinat, açı gibi geometrik ve bunları biçimlendirmek için çizgi stili, kalınlığı, rengi gibi biçimlendirme özellikleri vardır. Her mimarın tefriş sembolleri için kullandığı blokların temel çerçeveleri birbirine benzemekle beraber, çizgi stili, kalınlığı veya rengi ile kendi özgün biçimini kullanabilmektedirler. Tasarlanan modelde tefriş sembollerinin sadece geometrik şekillerine bakıldığından standart bir çizgi rengi, çizgi kalınlığı ve arka plan rengi

kullanılmıştır. Bu şekilde veri setini oluşturmak için boyut (çözünürlük), döndürme açısı, çizgi kalınlığı, çizgi stili, çizgi rengi ve arka plan rengi gibi belirtilen parametrelere dayalı olarak DXF bloklarını otomatik olarak bitmap görüntülere dönüştürebilen bir uygulama geliştirildi. Böylece tefriş sembolü için hangi renk, çizgi stili veya çizgi kalınlığı kullanılırsa kullanılsın, sistem otomatik olarak arka plan rengini siyah, çizgi rengini beyaz ve çizgi kalınlığını 1-piksel olacak şekilde tefriş sembollerini bitmap görüntülere dönüştürmektedir. Şekil 5.3'te vektör formattaki tefriş sembollerinin belirlenen ebat ve özelliklerde pikselleştirilmesine yönelik bir ekran görüntüsü gösterilmiştir.



Şekil 5.3. Vektörel tefriş sembolü pikselleştirme.

Şekil 5.4'te örnek olarak ocak-fırın sembolünün altı farklı ölçekte bitmap resme dönüştürülmüş gösterimleri verilmiştir. Örnekte görüldüğü gibi farklı ölçeklerde farklı çözünürlüklerde görüntüler elde edilmiş olsa bile oluşan görüntülerdeki çizgi kalınlıkları aynıdır.

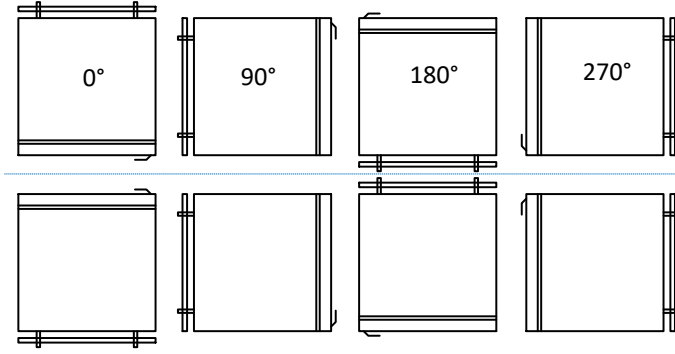


Şekil 5.4. Farklı ölçeklerde bitmap yapılmış ocak-fırın sembolü.

### 5.1.1. Veri Seti Çoğaltma ve Çeşitlendirme

Derin öğrenmede, veri setlerinin büyüklüğü, hesaplama ve kaynak maliyetini artırır da başarı için önemlidir. İyi bir model için veri setinin büyüklüğü tek başına yeterli değildir. Aynı zamanda veri çeşitliliği de önemlidir. Bazı durumlarda, bir veri seti yeterince büyük değilse veri artırma (data augmentation) yöntemleriyle veriler artırılmalıdır. Veri setini büyütmek için görüntülerde yeniden boyutlandırma, döndürme, yansıtma veya bazı dezenformasyonlar uygulanabilir [81,89,90].

Toplanan 1020 tefriş blok sembolünün her biri  $112 \times 112$ ,  $224 \times 224$  ve  $299 \times 299$  piksel çözünürlükte üç ayrı görüntüye dönüştürüldü. Daha sonra oluşan her bir görüntüden Şekil 5.5'te gösterildiği gibi döndürme ve yansıtma teknikleri ile sekiz ayrı görüntü elde edilmiş oldu. Orijinal görüntünün ( $0^\circ$ ) yanı sıra, sırasıyla  $90^\circ$ ,  $180^\circ$  ve  $270^\circ$  açılar ile döndürülmüş görüntüler, x-eksenine göre yansıtılarak bu sekiz görüntü elde edilmiş oldu. Bu işlem, hazırlanan sistem için bir avantaj olarak değerlendirilebilir. Çünkü tefriş sembolleri kat planlarına herhangi bir açıda konulabilmektedir. Bu şekilde oluşturulan üç farklı çözünürlükteki görüntülerden üç farklı veri seti oluşturulmuştur. Oluşturulan veri setleri, “*Furnishing Dataset*”’in kısaltması olan FDS ön eki eklenerek isimlendirilmiştir. Daha sonra bu veri setlerinin birleşimlerinden ayrıca bir veri seti daha oluşturulmuştur. Bu şekilde *FDSx112*, *FDSx224*, *FDSx299* ve dördünün birleşiminden oluşan *FDS\_Full* olmak üzere toplamda dört ayrı veri seti elde edilmiştir.



Şekil 5.5. Bir sembolün döndürülmüş ve yansıtılmış halleri.

Her bir veri setinin her sınıfından dengeli olacak şekilde %25'i daha sonra test sürecinde kullanılmak üzere ayrıldı. Oluşturulan veri seti isimleri, görüntü ebatları, eğitim ve test için ayrılan veri oranları ve sayıları Çizelge 5.2'de gösterilmiştir.

Çizelge 5.2. Tefriş sembolleri eğitim ve test veri setleri.

Veri Seti Adı	Görüntü Ebatları (piksel)	Adet ve Oran		
		(%100) Toplam (Adet)	(%75) Eğitim (Adet)	(%25) Test (Adet)
<i>FDSx112</i>	112×112	8160	6120	2040
<i>FDSx224</i>	224×224	8160	6120	2040
<i>FDSx299</i>	299×299	8160	6120	2040
<b><i>FDS_Full</i></b>	<b>Karışık</b>	<b>24480</b>	<b>18360</b>	<b>6120</b>

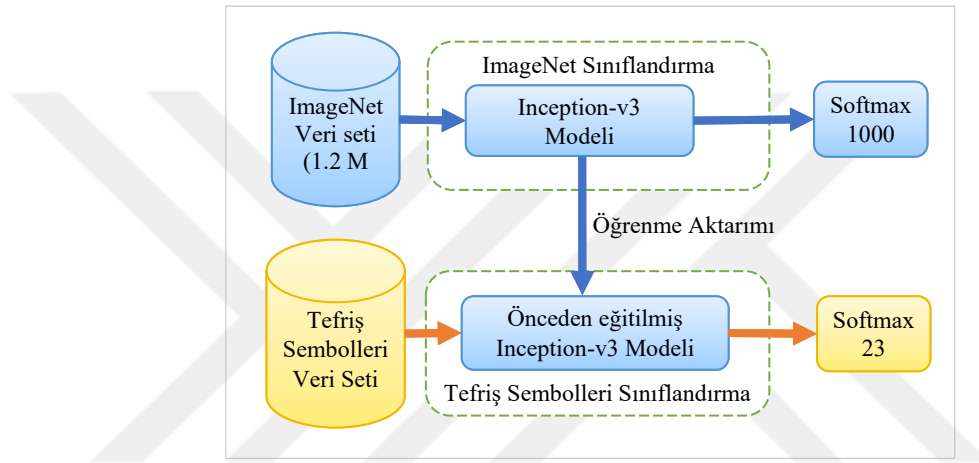
Veri seti eğitim ve test olarak ayrılırken örneğin *FDSx112*'de test için ayrılan aynı sembolün rastgele iki varyantı, *FDSx224* ve *FDSx299* setlerinde de test için ayrıldı. Dolayısıyla *FDS\_Full* veri setindeki eğitim ve test verileri diğer üç veri setlerindeki eğitim ve test verilerinin birleşiminden oluşmaktadır.

## 5.2. TEFRİŞ SINIFLANDIRMA İÇİN ÖNERİLEN METOD

Tefriş sembollerinin çizimden ayıklandıktan sonra sınıflandırılmaları artık bir görüntü sınıflandırma problemi olmaktadır. Görüntü sınıflandırması, bir görüntüyü girdi olarak alan ve onu belirlenmiş bir sınıfa atayan bir bilgisayarlı görü problemidir. Geçmişten günümüze sınıflandırma yapmak için birçok yöntem kullanılmasına rağmen günümüzde en iyi yöntemin derin öğrenme ile sınıflandırma olduğu yapılan çalışmaların sonuçlarından anlaşılmaktadır.

Derin öğrenme mimarilerine göre büyük bir veri setimiz olmadığından sınıflandırma yapmak için transfer öğrenme ile önceden eğitilmiş derin bir ESA mimarisi olan Inception-v3 modeli kullanılmıştır. Transfer öğrenmesi ayrıca zamandan tasarruf etmek ve daha az eğitim süresi ile daha iyi başarımlar elde etmek için kullanılmaktadır.

Bu çalışmada, “öğrenme aktarımı”, Inception-v3 modelinin ImageNet sınıflandırmasından edinilen özellik çıkarma bilgileri, tefriş sembolleri sınıflandırmasına aktarılmıştır. Önerilen model Şekil 5.6'da gösterilmektedir.



Şekil 5.6. Tefriş sembolleri sınıflandırması için önerilen metod.

### 5.2.1. Modelin Oluşturulması ve Eğitim

Model, Microsoft .NET ortamında ML.NET görüntü sınıflandırma API'si kullanılarak oluşturulmuştur. ML.NET görüntü sınıflandırma API'si, açık kaynaklı bir makine öğrenmesi platformu olan TensorFlow'u kullanmaktadır. Bu şekilde oluşturulan veri setleri için dört ayrı sınıflandırıcı oluşturulmuştur.

Eğitim sürecinde, eğitim veri setleri sırasıyla %80 ve %20 eğitim ve doğrulama setlerine bölünmüştür. Mini-batch boyutu 10'a ve öğrenme oranı başlangıçta 0.01 olup her iterasyonda adaptif olarak azalacak şekilde ayarlanmıştır. Böylece eğitim süreci her bir veri setinde 27 ile 33 iterasyon arasında erken bitirme (early stopping) kuralı ile bitirilmiştir. Veri setlerinin test için ayrılan kısımları hiçbir şekilde eğitime dahil edilmemiştir.

## 5.2.2. Deneysel Sonuçlar

Inception-v3 modelinde görüntüler otomatik olarak 299×299 piksel boyuta ayarlanmaktadır. Oluşturulan modelde veri artırma (data augmentation) yöntemleri ile her tefriş sembolü için üç farklı görüntü boyutu oluşturduk: 112×112, 224×224 ve 299×299. ML.NET ile Inception-v3 modeli kullanırken minimum girdi boyutunun 75×75 piksel olması dışında bir boyut şartı yoktur. ML.NET, ağ için görüntüleri otomatik olarak yeniden boyutlandırmaktadır.

Çizelge 5.3'te her bir veri seti için eğitim ve test sonucunda hesaplanan doğruluk ve kayıp değerleri verilmiştir. Çizelgeden de anlaşılacağı gibi test veri setinde en iyi başarımlar *FDS\_Full* veri setindedir. Yine eğitim ve doğrulamada en düşük kayıp oranı *FDS\_Full* veri setindedir.

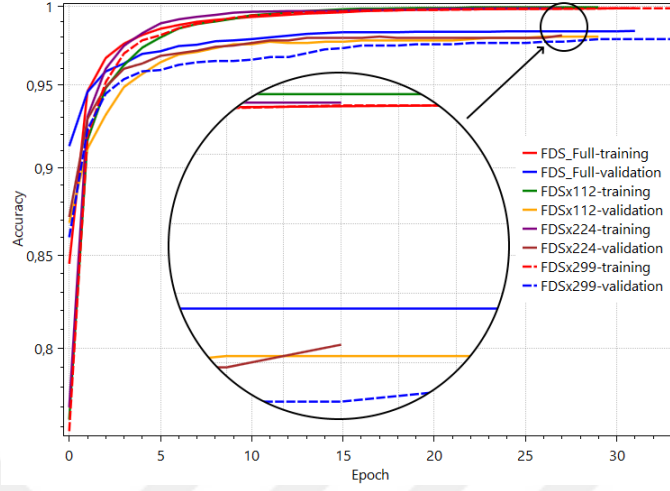
Çizelge 5.3. Eğitim ve test sonucu doğruluk ve kayıp sonuçları.

Dataset	Eğitim		Doğrulama		Test
	Accuracy	Loss	Accuracy	Loss	Accuracy
<i>FDSx112</i>	0.9993	0.0362	0.9805	0.1087	0.9828
<i>FDSx224</i>	0.9987	0.0327	0.9813	0.0851	0.9838
<i>FDSx299</i>	0.9985	0.0322	0.9789	0.0865	0.9779
<b><i>FDS_Full</i></b>	<b>0.9988</b>	<b>0.0261</b>	<b>0.9842</b>	<b>0.0577</b>	<b>0.9864</b>

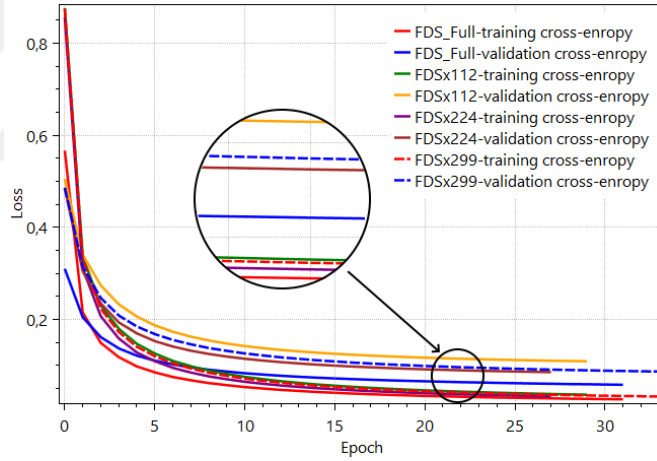
Çizelge 5.3'te gösterildiği gibi eğitim veri setindeki en iyi doğruluk *FDSx112* veri seti için oluşturulmuş sınıflandırıcıya ait olmasına rağmen doğrulama veri setinde üçüncü sırada kalmaktadır. Yine Çizelge 5.3'te gösterildiği gibi en kötü kayıp oranı da *FDSx112* için oluşturulan sınıflandırıcının doğrulama veri setine aittir. Bu da *FDSx112* veri seti için oluşturulmuş sınıflandırıcıyı diğer veri setleri için oluşturulmuş sınıflandırıcılara nispeten daha kötü yapmaktadır. Bu durumda doğrulama ve test veri setlerinde en iyi durumdaki *FDS\_Full* veri seti için hazırlanmış sınıflandırıcı en az kayıp ve en iyi doğruluk oranını verdiği için önerilen modelimiz için sınıflandırıcı olarak kullanılmıştır.

Şekil 5.7 ve Şekil 5.8'de eğitim ve doğrulama veri setlerinde sırasıyla doğruluk ve kayıp oranları grafik üzerinde karşılaştırmalı olarak gösterilmiştir. Şekil 5.9'de oluşturulan

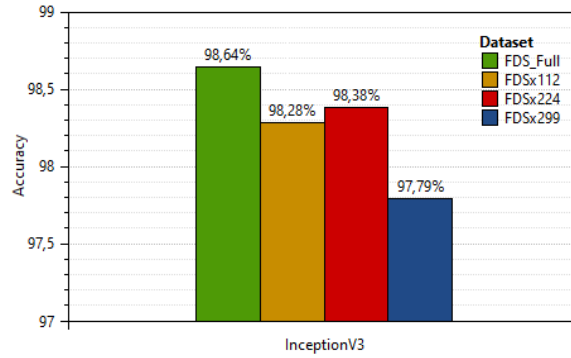
modellerin test veri setleri üzerinde doğruluk oranları grafik üzerinde karşılaştırmalı olarak verilmiştir.



Şekil 5.7. Eğitim ve doğrulama veri setlerindeki doğruluk oranları.

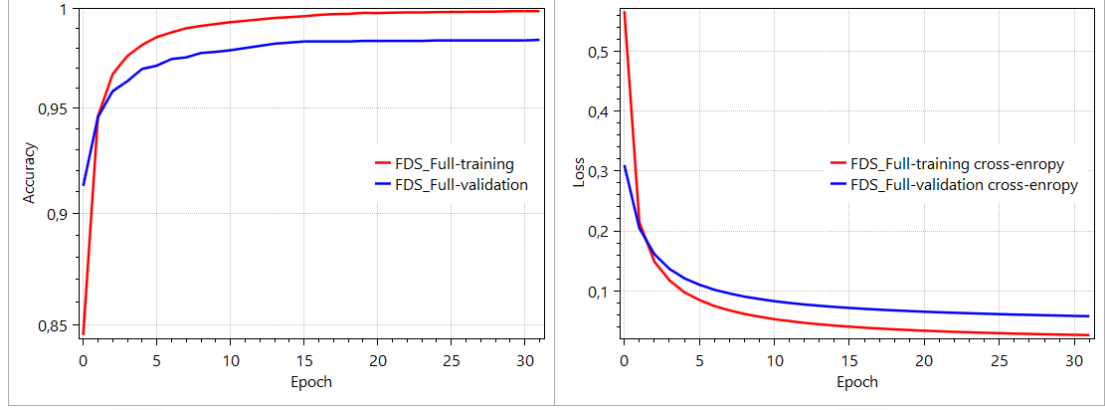


Şekil 5.8. Eğitim ve doğrulama veri setlerindeki kayıp oranları.



Şekil 5.9. Test veri setinde doğruluk oranları.

*FDS\_Full* veri seti için oluşturulmuş sınıflandırıcının eğitim ve doğrulama veri setleri üzerin gösterdiği doğrulama ve kayıp oranları Şekil 5.10'da gösterilmiştir.



Şekil 5.10. *FDS\_Full* veri seti için hazırlanmış sınıflandırıcının eğitim ve doğrulama veri setleri üzerindeki doğruluk ve kayıp oranları.

Şekil 5.10'da verilmiş kayıp grafiğine bakıldığında doğrulama kayıp oranı ( $0.0577$ ) eğitim kayıp oranından ( $0.0261$ ) birazcık fazladır. Bu da bize tasarlanan modelin olması gerektiği gibi öğrenme yaptığını göstermektedir. Çizelge 4.3'te de gösterildiği gibi eğer ağ, eksik öğrenme yapmış olsaydı eğitim ve doğrulama kayıp oranları çok yüksek olurdu. Aynı şekilde eğer aşırı öğrenme gerçekleşmiş olsaydı eğitim kayıp oranı çok düşük olmasına rağmen doğrulama kayıp oranı çok yüksek olurdu.

*FDS\_Full* veri seti ile hazırlanmış sınıflandırıcının test veri seti için karışıklık matrisi (confusion matrix) Çizelge 5.4'te verilmiştir. Çizelgede, sütunlarda gerçek sınıflara karşılık satırlarda o sınıf için sınıflandırılan (tahmin edilen) örnek adetleri bulunmaktadır. Karışıklık matrisi, sınıflandırıcının tefriş sembollerini birbirinden iyi derecede ayırdığını göstermektedir.

Her tefriş sınıfındaki veri sayıları aynı değildir. Veri sayıları bu şekilde dengesiz olduğundan sınıflandırmanın performansına bakmak için doğruluk oranına bakmak doğru değildir. Bunun yerine F1-puanı (F1-score) kullanılmalıdır. F1-puanı, özellikle dengesiz veri setlerinde bir sınıflandırıcının başarısını ölçmek için yaygın olarak kullanılan bir kriterdir. Çok sınıflı sınıflandırıcılarda F1-puanı her sınıf için ayrı ayrı hesaplanmaktadır. Her sınıf için F1-puanı'nı hesaplamak için önce her sınıf için kesinlik (precision) ve hassasiyet (recall) değerlerinin hesaplanması gerekir.

Çizelge 5.4. FDS\_Full test veri seti için karışıklık matrisi.

		GERÇEK SINIFLAR																						
		BED-DOUBLE	BED-SINGLE	DISHWASHER	DOOR-DOUBLE	DOOR-SINGLE	DOOR-WINDOWED	REFRIGERATOR	SHOWER	SINK	SOFA-CORNER	SOFA-ONE	SOFA-THREE	SOFA-TWO	STOVE-OVEN	TABLE-DINNER	TABLE-STUDY	TELEVISION	TOILET	WARDROBE	WASHBASIN	WASHBASIN-CABINET	WASHINGMACHINE	WINDOW
TAHMİN EDİLEN SINIFLAR	BED-DOUBLE	372	7	0	0	0	0	0	2	0	1	1	0	0	0	3	0	0	0	0	0	0	0	0
	BED-SINGLE	13	262	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0
	DISHWASHER	0	0	177	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	DOOR-DOUBLE	0	0	0	228	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	DOOR-SINGLE	0	0	0	0	407	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	DOOR-WINDOWED	0	0	0	0	0	114	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	REFRIGERATOR	0	0	1	0	0	0	114	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	SHOWER	0	0	0	0	0	0	0	356	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	SINK	0	0	0	0	0	0	0	0	228	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	SOFA-CORNER	1	0	0	0	0	0	0	0	0	95	0	0	0	0	0	0	0	0	0	0	0	0	0
	SOFA-ONE	1	0	0	0	0	0	0	0	0	0	311	0	0	0	0	0	0	3	0	0	0	0	0
	SOFA-THREE	0	0	0	0	0	0	0	0	0	1	0	245	10	0	1	0	0	0	0	0	0	0	0
	SOFA-TWO	0	0	0	0	0	0	0	0	0	2	0	4	205	0	0	0	0	0	0	0	0	0	0
	STOVE-OVEN	0	0	0	0	0	0	0	0	0	0	0	0	0	216	0	0	0	0	0	0	0	0	0
	TABLE-DINNER	2	0	0	0	0	0	0	0	0	2	0	0	0	0	658	4	0	1	0	0	0	0	0
	TABLE-STUDY	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2	175	1	0	0	0	0	0	0
	TELEVISION	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	77	0	0	0	0	0	0
	TOILET	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	398	0	0	3	0	0
	WARDROBE	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	324	0	0	0	0
	WASHBASIN	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	384	2	0	0
WASHBASIN-CABINET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	211	0	0	
WASHINGMACHINE	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180	0	
WINDOW	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	300	

Çok sınıflı sınıflandırıcılarda *kesinlik* (precision), bir sınıfa ait olduğu tahmin edilen örneklerin kaç tanesi gerçekten o sınıfa ait olduğunu gösteren bir metriktir ve Eşitlik 5.1’de gösterildiği gibi hesaplanır.

$$Precision = \frac{TP}{TP + FP} = \frac{372}{372 + 14} = 0.964 \quad ( 5.1 )$$

Örneğin Çizelge 5.4’teki karışıklık matrisine bakılırsa 386 tane sembol *BED-DOUBLE* olarak sınıflandırılmış, ancak bunlardan gerçekte 372 tanesi *BED-DOUBLE* sınıfına aittir. Bu durumda *BED-DOUBLE* için kesinlik 0.964 olur.

Çok sınıflı sınıflandırıcılarda *hassasiyet* (recall), bir sınıftaki örneklerin kaç tanesinin o sınıf için sınıflandırıldığını gösteren bir metriktir ve Eşitlik 5.2’de gösterildiği gibi

hesaplanır. Örneğin Çizelge 5.4'teki karışıklık matrisine bakılırsa 390 tane *BED-DOUBLE* sınıfına ait olan sembolden 372 tanesi *BED-DOUBLE* olarak sınıflandırılmıştır. Bu durumda *BED-DOUBLE* için hassasiyet 0.954 olur.

$$Recall = \frac{TP}{TP + FN} = \frac{372}{372 + 18} = 0.954 \quad ( 5.2 )$$

Her sınıf için F1-puanı (F1-score), her sınıf için hesaplanan kesinlik ve hassasiyet değerlerinin harmonik ortalaması olarak Eşitlik 5.3'te gösterildiği gibi hesaplanır. Buna göre *BED-DOUBLE* sınıfı için F1-puanı 0.959 olur.

$$F_1score = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * (0.964 * 0.954)}{(0.964 + 0.954)} = 0.959 \quad ( 5.3 )$$

Sınıflandırıcının genel F1-puanı değeri *F1-macro-score* ve *F1-micro-score* olmak üzere iki farklı şekilde hesaplanır:

*F1-macro-score* hesaplamak için tüm sınıflar için ayrı ayrı hesaplanmış *F1-puanı* değerlerinin aritmetik ortalaması hesaplanır (Eşitlik 5.4).

$$F_1^{macro} = \frac{1}{n} \sum_{class=1}^n F_1(class) = 0.9859 \quad ( 5.4 )$$

*F1-micro-score* hesaplamak için tüm sınıflar için hesaplanmış *micro-averaged-precision* (precision-micro) ve *micro-averaged-recall* (recall-micro) değerleri sırasıyla Eşitlik 5.5 ve Eşitlik 5.6'da gösterildiği gibi hesaplanır.

$$Precision^{micro} = \frac{\sum_i^n TP_i}{\sum_i^n TP_i + \sum_i^n FP_i} = \frac{6037}{6037 + 83} = 0.9864 \quad ( 5.5 )$$

$$Recall^{micro} = \frac{\sum_i^n TP_i}{\sum_i^n TP_i + \sum_i^n FN_i} = \frac{6037}{6037 + 83} = 0.9864 \quad ( 5.6 )$$

Precision-micro ve recall-micro deęerleri birbirine eřit olduęu iin harmonik ortalamaları da kendilerine eřit olacaktır. ok sınıflı sınıflandırıcılarda *micro-averaged-precision* ve *micro-averaged-recall* her zaman birbirine eřittir. Bu yzden *F1-micro-score* her zaman sınıflandırıcının doęruluk (*accuracy*) oranına eřit olur.

*FDS\_Full* veri seti iin oluřturulan sınıflandırıcı iin her sınıfın kesinlik, hassasiyet ve F1-puanı hesaplamaları izelge 5.5'te verilmiřtir.

izelge 5.5. *FDS\_Full* test veri setinde her sınıf iin hesaplanmıř F1-puanı

No	TEFRİŐ SINIFI	Sembol Sayısı	TP	FP	FN	Precision	Recall	F1 Score
1	BED-DOUBLE	390	372	14	18	0.964	0.954	0.959
2	BED-SINGLE	270	262	17	8	0.939	0.970	0.954
3	DISHWASHER	180	177	0	3	1.000	0.983	0.992
4	DOOR-DOUBLE	228	228	1	0	0.996	1.000	0.998
5	DOOR-SINGLE	408	407	0	1	1.000	0.998	0.999
6	DOOR-WINDOWED	114	114	0	0	1.000	1.000	1.000
7	REFRIGERATOR	114	114	1	0	0.991	1.000	0.996
8	SHOWER	360	356	0	4	1.000	0.989	0.994
9	SINK	228	228	0	0	1.000	1.000	1.000
10	SOFA-CORNER	102	95	1	7	0.990	0.931	0.960
11	SOFA-ONE	312	311	4	1	0.987	0.997	0.992
12	SOFA-THREE	252	245	12	7	0.953	0.972	0.963
13	SOFA-TWO	216	205	6	11	0.972	0.949	0.960
14	STOVE-OVEN	216	216	0	0	1.000	1.000	1.000
15	TABLE-DINNER	666	658	9	8	0.987	0.988	0.987
16	TABLE-STUDY	180	175	4	5	0.978	0.972	0.975
17	TELEVISION	78	77	2	1	0.975	0.987	0.981
18	TOILET	402	398	5	4	0.988	0.990	0.989
19	WARDROBE	324	324	2	0	0.994	1.000	0.997
20	WASHBASIN	384	384	4	0	0.990	1.000	0.995
21	WASHBASIN-CABINET	216	211	0	5	1.000	0.977	0.988
22	WASHINGMACHINE	180	180	1	0	0.994	1.000	0.997
23	WINDOW	300	300	0	0	1.000	1.000	1.000
<b>Toplam:</b>		<b>6120</b>						<b>0.9859</b>

ok sınıflı sınıflandırıcılarda genellikle bařarım lüt olarak *F1-macro-score* deęeri kullanılır. Dolayısıyla tefriř sınıflandırma iin nerilen sınıflandırıcının bařarım lüt %98.64'tr.

### 5.3. GERÇEK KAT PLANLARINDA TEST

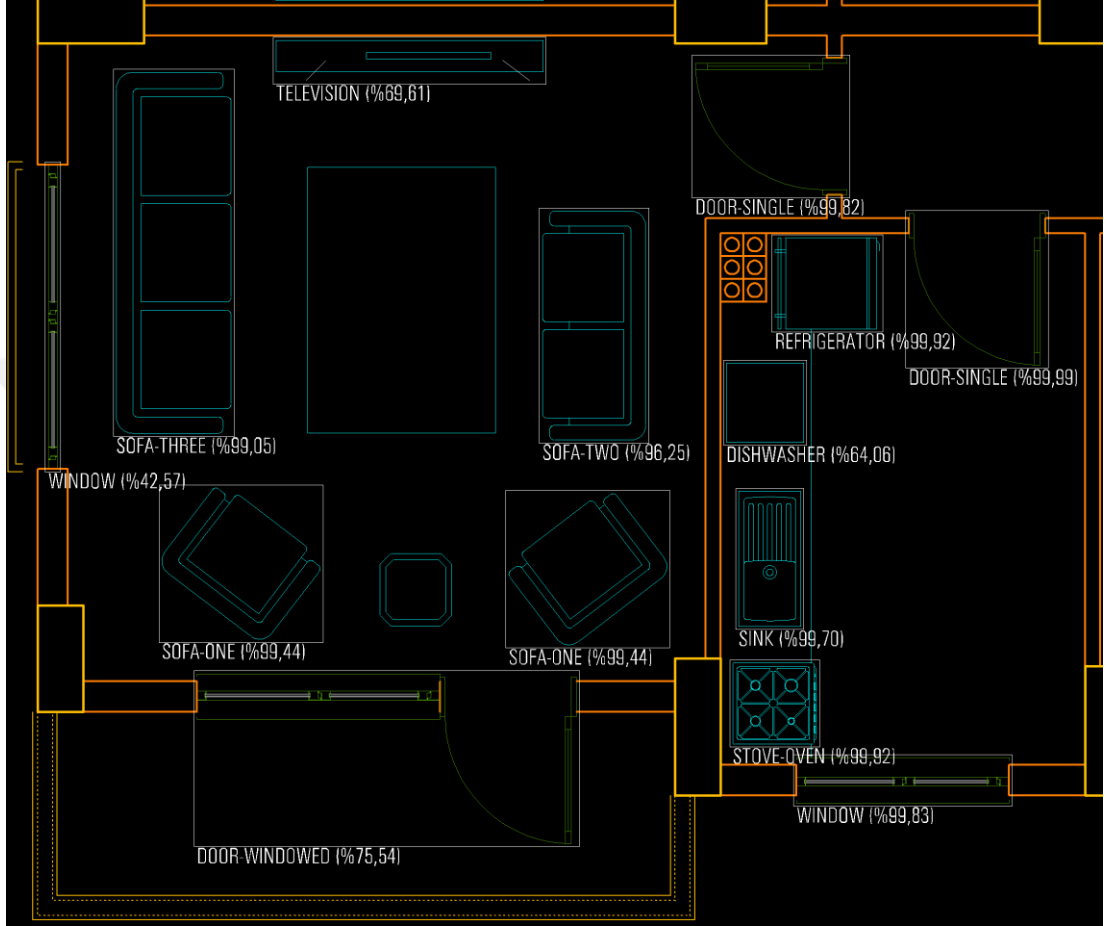
Oluşturulan sınıflandırma modeli, 20 gerçek kat planı üzerinde test edilmiştir. Bu kat planlarında toplam 2139 adet tefriş sembolü tespit edilmiş olup bunların 2076 tanesi doğru bir şekilde sınıflandırılmıştır.

Bu blok sembolleri, başlangıçta vektör formatında olduğundan sınıflandırma modelinde kullanabilmek için bitmap görüntü formatına dönüştürülmüştür. Her blok için üç farklı (112×112, 224×224 ve 299×299) piksel çözünürlükte bitmap görüntü uygulama tarafından otomatik olarak belirlenen özellikler ile (çizgi kalınlığı:1-piksel, çizgi-rengi: beyaz, arka-plan: siyah, çizgi-tipi: sürekli) oluşturulmuştur. Her blok için oluşturulan üç ayrı görüntü modele ayrı ayrı iletilmiş ve üç görüntüden en az ikisi için tahmin edilen sınıf, modelin tahmini olarak kabul edilmiştir. Bu işlem aynı zamanda üçlü bir kontrol yapılmasını da sağlamış olmaktadır. Çizelge 5.6’da her kat planında tespit edilen tefriş sembollerinin adedini ve doğruluk oranlarını göstermektedir. Çizelge 5.6’da görüldüğü gibi, önerilen model, tefriş sembollerini sınıflandırırken gerçek kat planlarında %97.05 doğruluğa ulaşmıştır. Bu da önerilen model için tatmin edici bir oran olarak değerlendirilebilir.

Çizelge 5.6. Gerçek kat planlarında test sonuçları.

Dosya Adı (Kat Planı)	Dosyada Bulunan Sembol Sayısı	Doğru Sınıflandırılan Sembol Sayısı	Doğruluk Oranı
M001	125	121	96.80%
M002	125	125	100.00%
M003	151	142	94.04%
M004	64	64	100.00%
M005	131	128	97.71%
M006	122	120	98.36%
M007	62	62	100.00%
M008	117	107	91.45%
M009	152	149	98.03%
M010	114	110	96.49%
M011	90	88	97.78%
M012	52	51	98.08%
M013	174	168	96.55%
M014	130	126	96.92%
M015	119	115	96.64%
M016	49	48	97.96%
M017	67	65	97.01%
M018	125	123	98.40%
M019	38	36	94.74%
M020	132	128	96.97%
<b>Toplam:</b>	<b>2139</b>	<b>2076</b>	<b>97.05%</b>

Şekil 5.11’de gerçek bir daire kat planı kesitinde bulunan tefriş sembollerinin sınıflandırması gösterilmiştir. Bu kat planı kesitinde bulunan tefrişlerin tamamı doğru olarak sınıflandırılmıştır.



Şekil 5.11. Örnek bir daire kat planında tefriş sınıflandırma.

## BÖLÜM 6

### ELEKTRİK PLAN ÇİZİMİNİN OTOMATİK OLUŞTURULMASI

Elektrik tesisat projeleri çizimler, hesaplamalar ve dökümlerden oluşur. Hesaplamalar çizimlere bağlı olarak yapılabilmektedir. Dökümler de hesaplamaların tablolar veya şemalar şeklindeki özetleridir. Elektrik tesisat planları çizildikten sonra hesaplamalar ve dökümler çizime bağlı otomatik olarak oluşturulabilmektedir.

Bu çalışmada konut binalarında elektrik iç tesisat plan çizimleri hazırlanırken priz ve aydınlatma armatürleri gibi elektrik malzeme sembollerinin mimari kat planları üzerindeki tefriş sembolleri baz alınarak uygun yerlere otomatik olarak yerleştirilmelerini sağlamak ve bağlantı şemalarının mekânsal ilişkiler göz önüne alınarak oluşturulmasını sağlamak için akıllı bir model geliştirilmiştir. Geliştirilen model, mimari kat planlarındaki tefriş sembollerinin bulunması, tanımlanması ve bulunduğu mahal bölgelerinin birbirleri ile ilişkilendirilmesi esasına dayanmaktadır.

Bu bölümde kat planları üzerinde yapılan elektrik tesisat plan çizimlerinin otomatik oluşturulması için yapılan işlemler açıklanmıştır. Birbirlerine bağımlı olarak yapılan bu işlemler üç ana başlık altında detaylandırılmıştır:

- Elektrik malzeme sembollerinin uygun yerlere yerleştirilmesi için yapılan işlemler,
  - Mahal bölgesi seçimi,
  - Seçilen mahal bölgesi içindeki tefriş sembollerinin bulunması ve sınıflandırılması,
  - Sınıflandırılan tefriş sembollerine göre uygun elektrik malzeme sembollerinin uygun duvarlara göre konumlandırılması,
- Aydınlatma tasarımı için yapılan işlemler,
  - Bulunan tefriş sembollerine göre mahal tiplerinin tespiti,
  - Tespit edilen mahale göre aydınlatma tasarımı ve armatürlerin yerleştirilmesi,

- Bağlantı hatlarının oluşturulması için yapılan işlemler,
  - Mekânsal ilişkilerin oluşturulması,
  - Yerleştirilen tüm elektrik malzemelerinin dağıtım panolarına bağlanması için line ve sorti bağlantılarının oluşturulması.

## 6.1. ELEKTRİK MALZEME SEMBOLÜ YERLEŞTİRME

Elektrik tesisat plan çizimleri yapılırken elektrik malzeme sembollerinin doğru yerlere yerleştirilmesi, işlemlerin ilk adımındır. Bu bölümde, bu işlemin tefriş tabanlı olarak otomatik yapılması açıklanmıştır. Elektrik malzeme sembollerinin kat planlarında olması gereken yerlere otomatik olarak yerleştirilmesi Algoritma 6.1’de verilen adımlara göre yapılmaktadır.

### Algoritma 6.1: Otomatik Elektrik Malzemesi Sembolü Yerleştirme

**Girdi:** Tefriş sembolleri içeren mimari kat planı.

**Çıktı:** Elektrik malzeme sembolleri yerleştirilmiş mimari kat planı

**Her bir** mimari kat planındaki **Mahal** için:

**Mahal** bölgesini içine alacak **Mahal-Çokgeni** oluştur.

*Not: Çokgen kenarları mahal alanının duvarlarıdır.*

**Mahal-Çokgeni** içinde kalan tüm **Blok** nesnelerini **Mahal-Blok-Listesi** içine ekle.

*Not: Tefriş sembollerinin mimari kat planlarında blok şeklinde konulduğu biliniyor.*

**Her bir Mahal-Blok-Listesi** içindeki **Blok** sembolü için:

- **Blok** sembolünü ağ için belirlenen çözünürlüklerde (112×112, 224×224 ve 299×299) ve özelliklerde (çizgi-kalınlığı: 1 piksel, çizgi-rengi: beyaz, çizgi-tipi: sürekli, arka-plan: siyah) resim nesnelere dönüştür.
- Resimleri önceden eğitilmiş ağa vererek tefriş sembolünü sınıflandır.
- Sınıflandırılan tefriş sembolüne göre tefriş-malzeme tablosundan (Çizelge 6.1) konulması gereken malzemeleri belirle.
- Belirlenen elektrik malzemelerinin sembolünü tefrişin olduğu duvara referanslı olarak yerleştir.

**Bitir**

Tanımlanan tefriş sembolleri için Çizelge 6.2’ye göre seçilen alan için mahal puanlaması yap ve puanlamaya göre **Mahal-Tipi** belirle

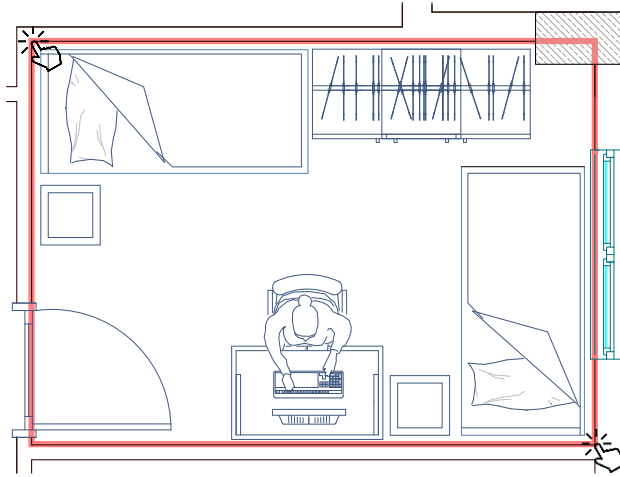
Belirlenen **Mahal-Tipi**’ne göre uygun aydınlatma tasarımı yap.

**Bitir**

### 6.1.1. Mahal Bölgesi Seçimi

Mimari kat planlarında odalar veya mahaller, çizimlerde açıkça belirlenmektedir. Kat planı çizimlerinde mahal bölgeleri çokgen (poligon) olarak değerlendirilebilir. Bir çokgen, art arda gelen uç noktalarının seçimi veya karşılıklı iki köşe noktasının belirlenmesi ile oluşturulabilir. Mimari kat planlarında odaların büyük bir çoğunluğu dikdörtgen veya dörtgen formda olmalarına rağmen koridor veya salonlar gibi bazı mahaller farklı şekillerde (örneğin L şeklinde) olabilir. Oda şekilleri nasıl olursa olsun elektrik malzeme sembolleri (prizler) her zaman duvardadır. Dolayısıyla *mahal çokgeni* doğru bir şekilde oluşturulmalıdır. **Mahal** bölgesini içine alacak **Mahal-Çokgeni** oluşturma işlemi üç farklı şekilde yapılabilir:

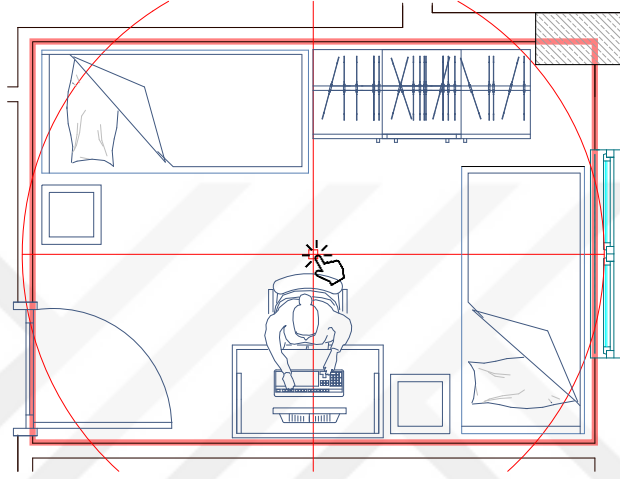
Mimari çizimlerde odaların büyük bir kısmı dikdörtgen formdadır. Dikdörtgen formdaki bir mahal bölgesini (oda) seçmek için iki nokta seçmek yeterlidir. Örneğin Şekil 6.1’de gösterildiği gibi odanın sol-üst ve sağ-alt köşeleri seçilerek odayı içine alan mahal çokgeni oluşturulabilir. Bu yöntem sadece dikdörtgen odalarda işe yaramaktadır.



Şekil 6.1. İki nokta ile mahal çokgeni oluşturma.

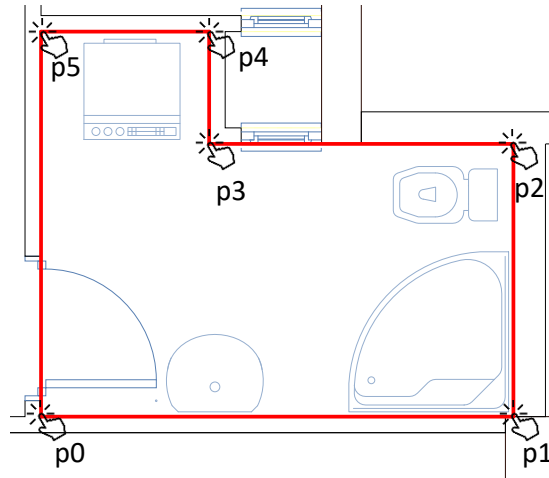
Dörtgen veya dikdörtgen formdaki mahal bölgeleri (dört duvar odalar), imlecin sol-sağ-alt-üst konumuna en yakın duvar çizgilerin kesiştirilmesi yardımı ile mahal bölgesinin içinden aktif olarak bir tıklama ile tespit edilebilir. Bunu yapmak için imlecin bulunduğu konuma en yakın soldaki, sağdaki, üstteki ve alttaki duvar çizgi

parçaları kesiştirilerek bir dörtgen oluşturulur. Dikdörtgen olmasına gerek kalmadan dört duvar olan tüm odalarda bu yöntem işe yaramaktadır. İmlece yakın duvar çizgilerini bulmak için kat planındaki tüm çizgi parçalarına bakmak işlem yükünü arttırır. Bunun yerine sadece imlece belirli bir r mesafesindeki çizgi parçalarına bakılarak işlem yükü azaltılabilir. Şekil 6.2’de tek noktaya tıklama ile odayı içine alan mahal çokgeninin oluşturulması gösterilmiştir.



Şekil 6.2. Tek nokta ile mahal çokgeni oluşturma.

Seçilecek mahal bölgesi dikdörtgen veya dörtgen değilse, örneğin L şeklinde bir oda ise, odanın uç noktaları sıralı tıklamalar yardımıyla belirlenerek odayı içine alan mahal çokgeni oluşturulabilir. Şekil 6.3’te L şeklinde bir banyonun sınırları belirlenerek mahal çokgeni oluşturulmuştur.



Şekil 6.3. Sınırlarını belirleyerek mahal çokgeni oluşturma.

### 6.1.2. Tefriş Sembollerinin Bulunması

Bu çalışmada, sadece tefriş sembollerini sınıflandırmaya değil, elektrik malzemelerinin yerleştirilmesi ve mekânsal ilişkilerin oluşturulabilmesi için aynı zamanda tefriş sembollerinin konumlarına da ihtiyaç vardır. Genel nesne algılamanın amacı, bir görüntüdeki mevcut nesnelere bulmak ve sınıflandırmak ve bunları sınırlayıcı kutularla etiketlemektir [133]. Çalışmamızdaki kat planları resim görüntülerden oluşsaydı, öncelikle R-CNN veya YOLO gibi klasik nesne algılama algoritmalarını kullanarak tefriş sembollerini konumlarıyla birlikte tespit etmemiz gerekecekti.

Vektörel çizim dosyalarında koordinatlar, açılar ve benzeri bilgilerden oluşan geometrik veriler bulunmaktadır. Vektör dosyalarındaki nesnelere, uygulama programı tarafından çalışma anında görselleştirilmektedir. Bu çalışmada kullanılan kat planları vektör formatında olduğu için klasik nesne algılama algoritmalarının kullanılmasına gerek kalmamıştır. Vektör çizimlerde tüm blok sembolleri zaten konumlarıyla beraber bir listededir. Ayrıca dosyadaki geometrik veriler sayesinde bu blok nesnelere sınırları hesaplanabilmektedir.

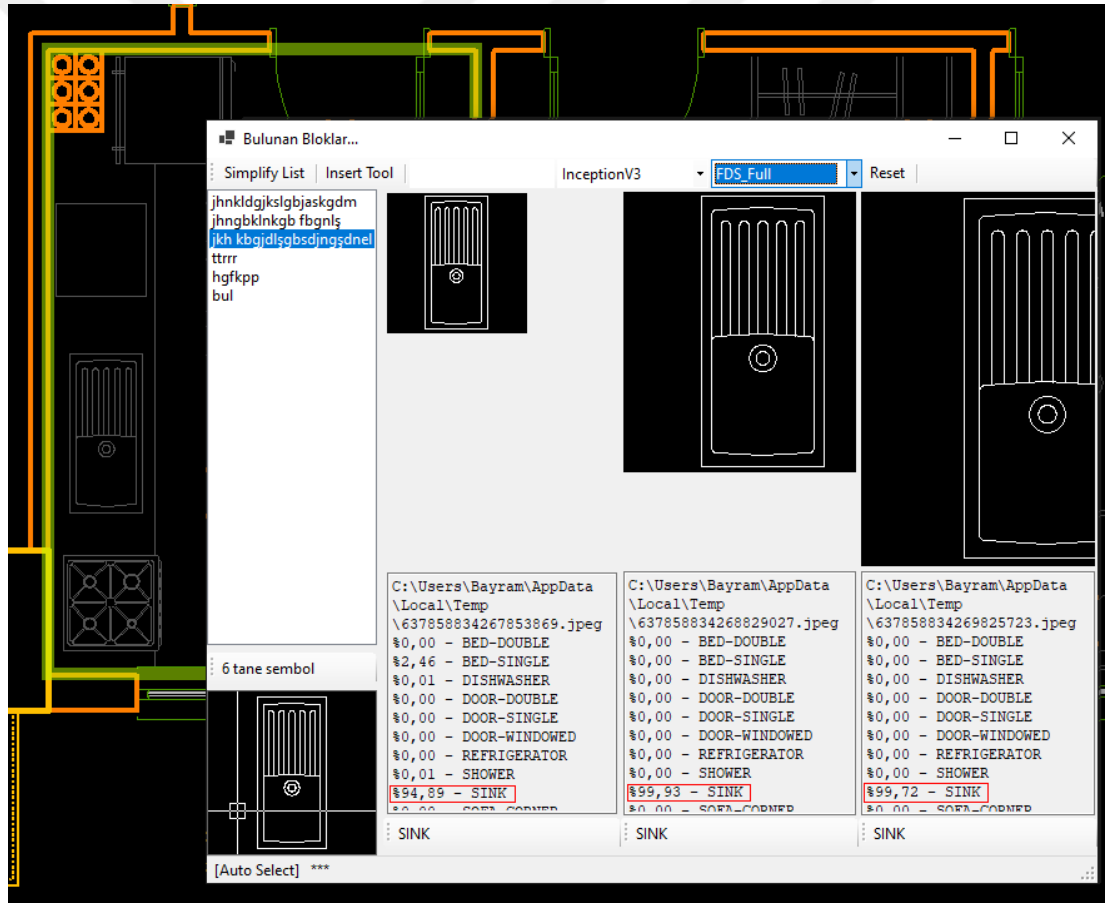
*Mahal-Çokgeni* içinde kalan tüm *Blok* nesnelere *Mahal-Blok-Listesi* içine eklemek için oluşturulmuş *mahal çokgeni* içinde koordinatları bilinen tüm grafik nesnelere (çizgi, çember, yay, blok, vb.) konumlarına göre belirlenmiş ve ardından blok olanlar ayrı bir *Mahal-Blok-Listesi* olarak gruplandırılmıştır. Kat planı çizimindeki tüm blok nesnelere potansiyel tefriş sembolleri olarak kabul edilmektedir. Böylece *Mahal-Çokgeni* oluşturulmuş bir alandan tefriş sembolleri konumlarına göre belirlenerek *Mahal-Blok-Listesi* olarak listelenmiş olur.

### 6.1.3. Tefriş Sembollerinin Sınıflandırılması

Seçilen odanın *Mahal-Çokgeni* içindeki çizim bloklarının hangi tefriş sembolü olduğunu sınıflandırmak için, *Mahal-Blok-Listesi* içindeki her bir *Blok* sembolünün orijinal renk, kalınlık ve çizgi stili özellikleri göz ardı edilerek daha önce ağ için belirlenen çözünürlüklerde (112×112, 224×224 ve 299×299) ve özelliklerde (çizgi-

kalınlığı:1-piksel, çizgi-rengi: beyaz, çizgi-tipi: sürekli, arka-plan: siyah) resim nesnelere dönüştürülmektedir.

Bu şekilde, her blok için oluşturulan üç ayrı görüntü modele ayrı ayrı iletilmektedir. Üç görüntüden en az ikisi için tahmin edilen sınıf, modelin tahmini olarak kabul edilmektedir. Şekil 6.4'te seçilen mutfak alanında 6 tane blok (kapı, pencere, buzdolabı, bulaşık makinesi, evye ve ocak-fırın) listelenmiştir. Bu bloklardan her biri için 112×112, 224×224 ve 299×299 piksel çözünürlükte ve belirlenen özelliklerde üç ayrı resim görüntüsü oluşturulmuştur. Örneğin bulunan bir evye bloğu için oluşturulan görüntüler ayrı ayrı eğitilmiş ağa verilir ve *SINK* (evye) olarak sınıflandırılmışlardır.



Şekil 6.4 Bloklardan görüntü oluşturma ve sınıflandırma.

#### 6.1.4. Tefriş Sembolüne Göre Elektrik Malzemesi Önerme

Elektrik tesisat planı çizilirken elektrik malzeme sembollerini önermek için proje tasarımcılarının deneyimleri kullanılarak Çizelge 6.1'de verilen *Look-up Table*

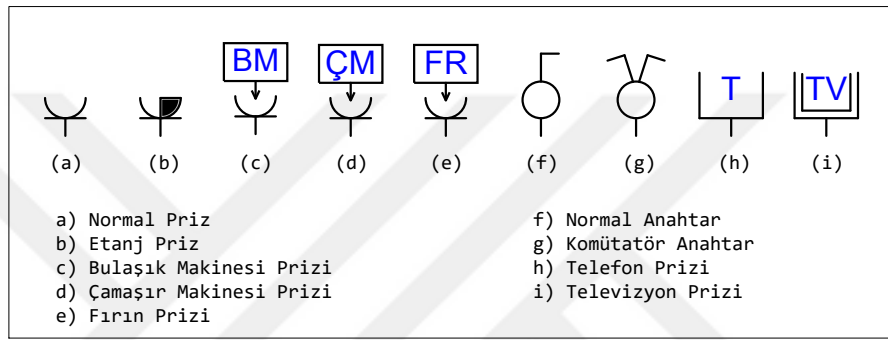
oluşturulmuştur. Bu tablo ülke ve/veya bölgeye göre özelleştirilebilir. Bu çizelge tamamen projecilerin tecrübelerinden faydalanılarak hazırlandığı için bazı tefriş sembolleri çizelgede bulunmamaktadır. Örneğin duş kabini, tekli koltuk veya dolaplar için herhangi bir elektrik malzemesi önerilmemiştir. Bu tefrişler genellikle elektrik malzemeleri ile ilişkili değildir. Ancak yine de bu tefriş sembolleri bulunduğu mahalin tipini belirlemesi bakımından önemli tefrişlerdir.

Çizelge 6.1. Tefrişe göre elektrik malzemesi önerme tablosu.

Tefriş Sembolü	Tefriş	#	Elektrik Malzemesi	Gücü (Watt)	Açıklama
STOVE-OVEN	Fırın-Ocak	1	Fırın Prizi	2500	Fırın için
		2	Normal Priz	300	Ocak için
		3	Normal Priz	300	Davlumbaz için
WASHINGMACHINE	Çamaşır Makinesi	1	Çamaşır Makinesi Prizi	2500	
DISHWASHER	Bulaşık Makinesi	1	Bulaşık Makinesi Prizi	2000	
REFRIGERATOR	Buzdolabı	1	Normal Priz	300	
SINK	Evye	1	Normal Priz	300	Çeşitli mutfak eşyaları için (Su ısıtıcı, karıştırıcı, öğütücü, ...)
		2	Etanj Priz	300	
TABLE-DINNER	Yemek Masası	1	Normal Priz	300	
TELEVISION	Televizyon	1	Normal Priz	300	TV İçin
		2	Normal Priz	300	Uydu alıcı için
		3	Anten Prizi	–	
		4	Telefon Prizi	–	
TABLE-STUDY	Çalışma Masası	1	Normal Priz	300	Bilgisayar, Masa lambası, ...
		2	Telefon Prizi	–	
SOFA-TWO	İkili Koltuk	1	Normal Priz	300	Arkasına
SOFA-THREE	Üçlü Koltuk	1	Normal Priz	300	Sağ tarafa
		2	Normal Priz	300	Sol tarafa
SOFA-CORNER		1	Normal Priz	300	Sağ tarafa
		2	Normal Priz	300	Sol tarafa
BED-DOUBLE	Çift Kişilik Yatak	1	Normal Priz	300	Sağ tarafa
		2	Normal Priz	300	Sol tarafa
BED-SINGLE	Tek Kişilik Yatak	1	Normal Priz	300	
WASHBASIN-CABINET	Banyo lavabosu	1	Etanj Priz	300	Tıraş makinesi, saç kurutma...
DOOR-SINGLE		1	Işık Anahtarı	–	Tüm kapılarda

### 6.1.5. Elektrik Malzeme Sembolü Konumlandırma

CAD programları kullanılarak elektrik tesisat planları çizilirken, elektrik malzeme sembollerinin doğru yerlere uygun açılar ile yerleştirilmesi ilk adımdır. Tefriş sembolleri bulunup sınıflandırıldıktan sonra yerleştirilecek elektrik malzeme sembolleri Çizelge 6.1'den seçilmektedir. Kullanılan bazı elektrik malzeme sembolleri Şekil 6.5'da gösterilmiştir. Bu semboller CAD çizimlerinde blok olarak kullanılmaktadır.

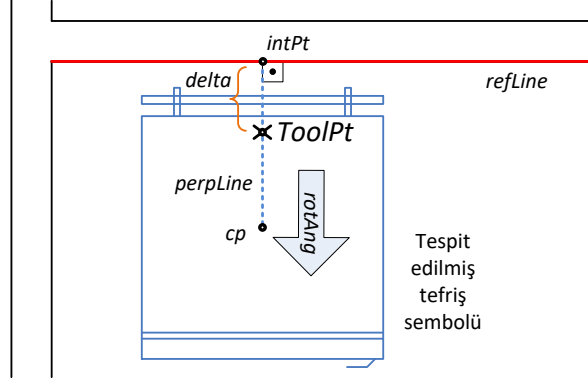


Şekil 6.5. Bazı elektrik malzeme sembolleri

Bir tefriş sembolü tanımlandıktan sonra ona uygun elektrik malzeme sembolünü yerleştirmek için en uygun duvar çizgisini tespit etmek gerekir. Seçilen elektrik malzeme sembolü, tefriş sembolünün bulunduğu geometrik konuma en yakın duvar çizgisine yerleştirilir. En yakın duvar çizgisini tespit etmek ve elektrik malzeme sembolünü uygun şekilde yerleştirmek için sırasıyla aşağıdaki işlemler yapılmaktadır [1]:

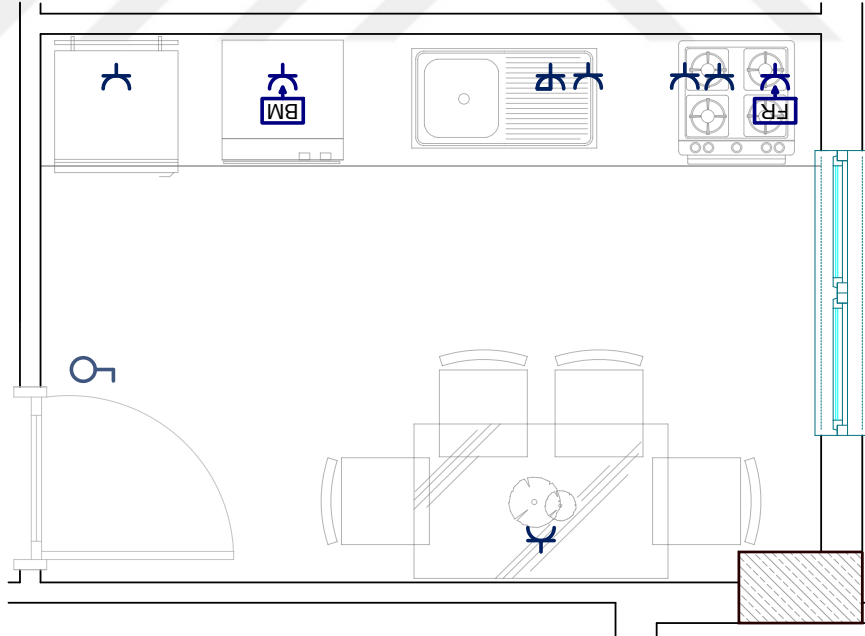
- Tanımlanan tefriş sembolünün merkez noktasına ( $cp$ ) en yakın çizgi parçası (duvar çizgisi) seçilir. Bu çizgi parçası referans çizgisi ( $refLine$ ) olarak kabul edilecektir.
- $cp$  noktasından  $refLine$  doğru parçasına dik bir çizgi ( $perpLine$ ) çizilir ve kesişim noktası ( $intPt$ ) bulunur.
- $perpLine$  doğru parçasının açısı, elektrik malzeme sembolü için koyma açısı ( $rotAng$ ) olarak kabul edilecektir.
- $intPt$  noktasından  $perpLine$  doğru parçası doğrultusunda belirtilen  $delta$  mesafe kadar ilerisi elektrik malzeme sembolü için koyma noktası ( $ToolPt$ ) olacaktır.

Elektriksel malzeme sembollerinin yukarıdaki adımlara göre uygun çizgi parçası (duvar çizgisi) üzerindeki yerleşimi Şekil 6.6'de gösterilmiştir.



Şekil 6.6. Uygun duvara elektrik sembolünün otomatik konulması.

Şekil 6.7'de gösterildiği gibi, seçilen mahal bölgesinde bulunan tefriş sembollerine uygun elektrik malzeme sembolleri Çizelge 6.1'den seçilerek uygun duvarlara yerleştirilmiştir.



Şekil 6.7. Elektrik malzemesi sembollerinin, tanımlanmış tefriş sembollerine göre otomatik yerleştirilmesi.

## 6.2. MAHAL TİPİNE GÖRE AYDINLATMA TASARIMI

Aydınlatma tesisat planları her ne kadar 2D ortamda çizilse bile bir odaya konulması gereken aydınlatma armatürlerinin uygun aydınlatma seviyesi sağlayıp sağlamadığını 3D ortamda test etmek gerekmektedir. Sektörde bir aydınlatma armatürünün 3D kapalı alanda sağladığı aydınlatma seviyesini modelleyerek ölçen Dialux [134] ve Relux [135] gibi uygulamalar mevcuttur. Ancak bu uygulamalarda odaya uygun armatür tavsiyesi bulunmamaktadır.

Bu çalışma kapsamında 3D kapalı alanlarda mahal tipine, büyüklüğüne ve kullanım amacına uygun, istenen minimum aydınlatma düzeyini sağlayacak şekilde aydınlatma armatürü tipi, sayısı ve konulması gereken yerler için öneri yapan bir aydınlatma simülasyon çalışması yapılmıştır. Yapılan uygulama sayesinde ışık dağılımı bilinen bir aydınlatma armatürü için 3D kapalı bir alanda istenen bir düzlem üzerinde aydınlatma şiddeti hesaplanabilmektedir. Yapılan uygulamanın avantajı sisteme entegre olmasıdır.

### 6.2.1. Tefriş Sembollerinden Mahal Tipi Belirleme

Tefriş sembolleri, çizilen mimari projelerde oluşturulan odaların kullanım amacını göstermektedir. Örneğin bir mahalde evye veya ocak tefrişi varsa o mahalin mutfak olduğu kolaylıkla anlaşılır.

Bu çalışmada mahal tipinin belirlenmesindeki temel amaç, mahale uygun aydınlatma tasarımı yapmaktır. Seçilen mahal çokgeni içinde bulunan tefriş sembollerinden, **Mahal-Tipi** çözümlmek için Çizelge 6.2'deki tabloya göre bir puanlama sistemi geliştirilmiştir. Çizelgede bulunan tefrişlere göre puan verilerek **Mahal-Tipi** belirlenmektedir.

Çizelge 6.2'ye göre örneğin bir mahalde “televizyon” ve “yemek masası” tefriş sembolleri bulunursa; televizyon için *salon* mahal tipine +1 puan, yemek masası için hem salon hem de mutfak mahal tiplerine +1'er puan verilecektir. Böylece mutfak 1, salon 2 puan alacağından bu mahalin "salon" olduğu anlaşılmiş olacaktır. Böyle bir senaryoda "salon" mahal tipine göre aydınlatma kuralları uygulanacaktır. Benzer

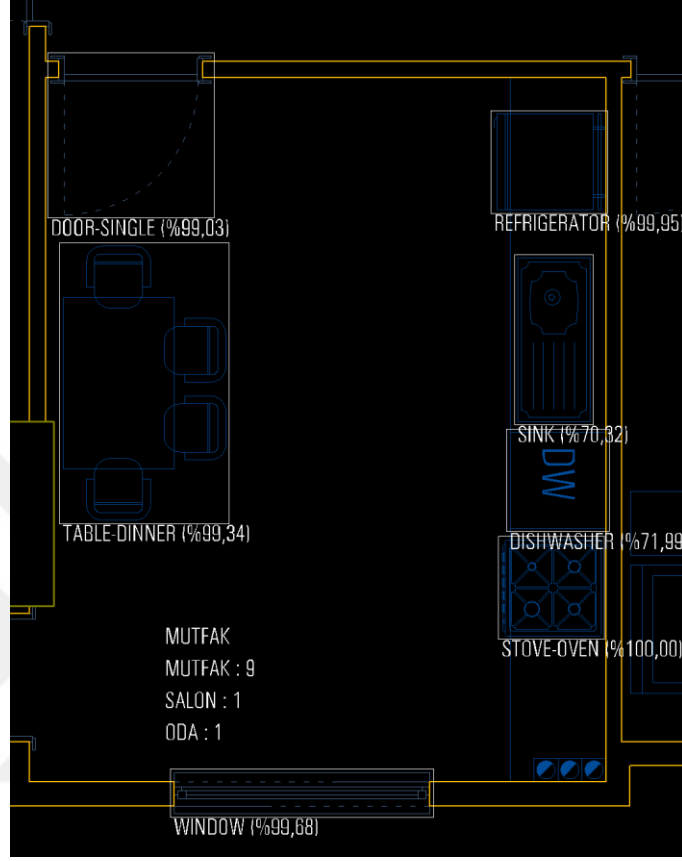
şekilde örneğin bir mahalde "tekli yatak" bulunursa "çocuk odası" adındaki mahale 1, "çalışma masası" bulunursa "çocuk odası", "salon" ve "çalışma odası" adındaki mahallere 1'er puan verilecektir. "Çocuk odası" adındaki mahalın toplamda 2 puanı olacağı için bu mahalın "çocuk odası" olduğu tespit edilmiş olacak ve "çocuk odası" mahal tipine göre aydınlatma kuralları uygulanacaktır.

Çizelge 6.2. Tefriş sembolüne göre mahal-tipi puanlama tablosu.

No	Tefriş Sınıfı	Açıklama	Mahal-Tipi	Puan
1	BED-DOUBLE	Çift kişilik yatak	Yatak Odası	3
2	BED-SINGLE	Tek kişilik yatak	Çocuk Odası	1
3	DISHWASHER	Bulaşık makinesi	Mutfak	2
4	DOOR-DOUBLE	İki kanatlı kapı	Salon	1
5	DOOR-SINGLE	Tek kanatlı kapı	Oda	-
6	DOOR-WINDOWED	Pencereli kapı	Balkon	1
			Mutfak	1
7	REFRIGERATOR	Buzdolabı	Mutfak	2
8	SHOWER	Duş kabini	Banyo	2
9	SINK	Evye	Mutfak	2
10	SOFA-CORNER	Köşe takımı koltuk	Salon	2
11	SOFA-ONE	Tekli koltuk	Salon	2
12	SOFA-THREE	Üçlü koltuk	Salon	2
13	SOFA-TWO	İkili koltuk	Salon	2
14	STOVE-OVEN	Ocak-Fırın	Mutfak	2
15	TABLE-DINNER	Yemek masası	Mutfak	1
			Salon	1
16	TABLE-STUDY	Çalışma masası	Çocuk Odası	1
			Çalışma Odası	1
			Salon	1
17	TELEVISION	Televizyon	Salon	1
18	TOILET	Tuvalet	Banyo	1
			Tuvalet	1
19	WARDROBE	Gardırop	Yatak Odası	1
			Çocuk Odası	1
20	WASHBASIN	Lavabo	Banyo	1
			Tuvalet	1
21	WASHBASIN-CABINET	Dolaplı Lavabo	Banyo	2
22	WASHINGMACHINE	Çamaşır makinesi	Banyo	1
23	WINDOW	Pencere	Oda	1

Yapılan uygulamada örnek olarak seçilen ve Şekil 6.8'da gösterilen mahal bölgesinde bulunan tefriş sembollerine göre mahal puanlama tablosu Çizelge 6.3'te verilmiştir.

Bu puanlamaya göre mahal tipi “Mutfak” olarak belirlenmiştir. Dolayısıyla aydınlatma kuralları “Mutfak” için uygulanacaktır.



Şekil 6.8. Tefriş puanlama ile mahal tipi tespiti.

Çizelge 6.3. Örnek mahalde tefriş sembollerine göre mahal tipi belirleme.

Tefriş Sınıfı	Açıklama	Mahal-Tipi	Puan
DISHWASHER	Bulaşık makinesi	Mutfak	2
DOOR-SINGLE	Tek kanatlı kapı	Oda	-
REFRIGERATOR	Buzdolabı	Mutfak	2
SINK	Evye	Mutfak	2
STOVE-OVEN	Ocak-Fırın	Mutfak	2
TABLE-DINNER	Yemek masası	Mutfak	1
		Salon	1
WINDOW	Pencere	Oda	1
Mutfak = 9, Salon = 1, Oda=1			
<b>SONUÇ: Mutfak</b>			

## 6.2.2. Aydınlatma Hesabı

Noktasal bir ışık kaynağından R düzlemi üzerindeki bir noktada ışık şiddeti Eşitlik 6.1'deki gibi hesaplanır. Burada E hesaplanana ışık şiddetini (lx), I noktasal ışık akısını (lm), r ışık kaynağı ile düzlemdeki nokta arasındaki öklit mesafesini (m),  $\theta$  düzlemin normali ile ışık noktası arasındaki açıyı ifade eder.

$$E = \frac{I \cos \theta}{r^2} \quad ( 6.1 )$$

Ancak gerçekte noktasal ışık kaynağı yoktur. Aydınlatma armatürleri uzayda homojen olmayan bir şekilde ışık yaymaktadır. Dolayısıyla bir armatür ile aydınlatma hesabı yapmak için o armatürün uzayda yaydığı ışık dağılım bilgilerinin bilinmesi gerekmektedir. Bir aydınlatma armatürünün ışık dağılım bilgileri armatürün *eulumdat* dosyasından alınmaktadır.

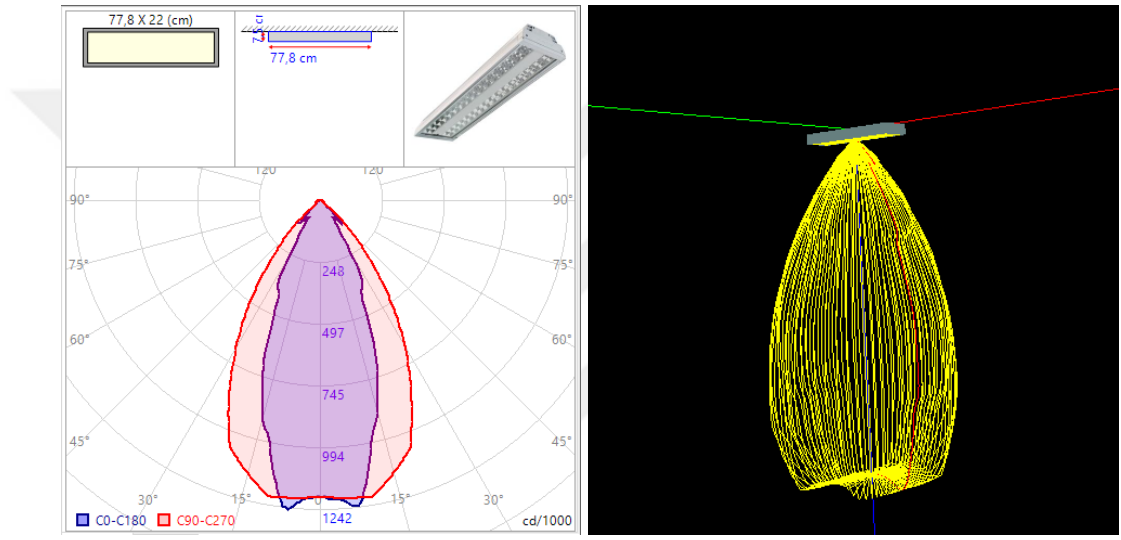
General	Luminaire	Lamps	Luminous intensities							Light distribution						
	C0°	C5°	C10°	C15°	C20°	C25°	C30°	C35°	C40°	C45°	C50°	C55°	C60°	C65°	C7°	
0°	439.9	439.9	439.9	439.9	439.9	439.9	439.9	439.9	439.9	439.9	439.9	439.9	439.9	439.9	439.9	
0.5°	439.7	439.6	439.6	439.5	439.5	439.4	439.4	439.3	439.3	439.3	439.2	439.2	439.2	439.2	439.2	
1°	439.5	439.4	439.3	439.2	439.1	439	438.9	438.8	438.8	438.7	438.7	438.6	438.6	438.6	438.6	
1.5°	439.2	439.1	438.9	438.8	438.6	438.5	438.4	438.3	438.2	438.1	438.1	438	438	437.9	437.9	
2°	439	438.8	438.6	438.4	438.2	438	437.9	437.7	437.6	437.5	437.4	437.4	437.3	437.3	437.3	
2.5°	438.5	438.3	438	437.8	437.6	437.4	437.3	437.1	437	436.9	436.8	436.7	436.7	436.7	436.7	
3°	438.1	437.8	437.5	437.2	437	436.8	436.6	436.4	436.3	436.2	436.2	436.1	436	436	436	
3.5°	437.4	437.1	436.7	436.5	436.2	436	435.9	435.7	435.6	435.5	435.4	435.4	435.3	435.3	435.3	
4°	436.7	436.3	436	435.7	435.4	435.2	435	434.9	434.8	434.7	434.7	434.6	434.6	434.6	434.6	
4.5°	436	435.5	435.1	434.8	434.5	434.3	434.1	434	433.9	433.9	433.9	433.9	433.9	434	434	
5°	435.2	434.7	434.2	433.9	433.6	433.3	433.2	433.1	433	433	433.1	433.1	433.2	433.2	433.2	
5.5°	434.3	433.7	433.3	432.9	432.6	432.4	432.2	432.1	432.1	432.1	432.2	432.3	432.4	432.5	432.5	
6°	433.3	432.7	432.2	431.8	431.5	431.3	431.2	431.1	431.1	431.2	431.3	431.4	431.6	431.7	431.7	
6.5°	432.2	431.6	431.1	430.7	430.4	430.2	430.1	430	430.1	430.2	430.4	430.6	430.8	431	431	
7°	431.1	430.4	429.9	429.4	429.1	429	429	429	429	429.2	429.4	429.7	430	430.3	430.3	

Dimension unit: cd / 1000 lm

Şekil 6.9. Eulumdat dosyası içeriği.

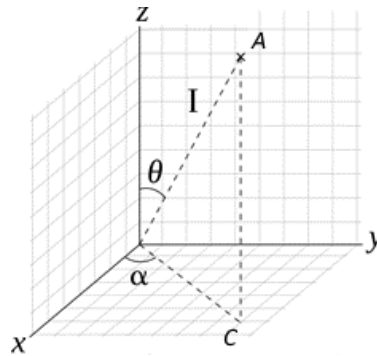
Eulumdat, lambalar ve armatürler gibi ışık kaynaklarından gelen fotometrik ışık dağılım verilerinin belirlenmesi için kullanılan bir veri dosyası formatıdır. Şekil 6.9'da bir eulumdat dosyasının içeriği gösterilmiştir. Şekil 6.9'daki tabloda, sütunlar armatürün yatayda  $\alpha$  açısı ile, satırlar ise armatürün dikeyde  $\theta$  açısı ile yaydığı ışık akısını ifade eder.

Bir armatürün eulumat dosyasında armatürün uzayda ışık dağılım bilgilerinin yanında üretici bilgileri, armatürün ebatları, ışıklık ebatları, armatürün gücü, lümeni, ışık renk sıcaklığı, lamba tipi ve adedi gibi bilgiler bulunmaktadır [136]. Armatür üreticileri her armatüre ait eulumat dosyasını armatürün üretimi ile oluştururlar. Bir eulumat dosyasındaki veriler genellikle bir gonyofotometre (goniofotometre) kullanılarak ölçülür. Bir gonyofotometre, farklı açılarda bir nesneden yayılan ışığın ölçülmesi için kullanılan bir cihazdır [137]. Şekil 6.10’da örnek bir armatürün eulumat bilgileri kullanılarak 2D ve 3D ortamda ışık dağılım eğrileri gösterilmiştir.



Şekil 6.10. Bir aydınlatma armatürünün 2D ve 3D ortamda ışık dağılım eğrileri.

Uzayda ışık dağılım eğrileri bilinen ve konumu Şekil 6.11’de gösterilen A noktasındaki bir aydınlatma armatürünün R düzlemi üzerindeki bir C noktasında aydınlatma şiddetini (lux) hesaplamak için Eşitlik 6.2 kullanılır.



Şekil 6.11. A noktasındaki armatür ile C noktasında aydınlatma şiddeti hesaplamak için gerekli olan  $\alpha$  ve  $\theta$  açıları.

$$E = \frac{I_{\alpha,\theta} \cos \theta}{r^2} \quad (6.2)$$

Burada  $I_{\alpha,\theta}$  aydınlatma armatürünün yatayda  $\alpha$  ve dikeyde  $\theta$  açısındaki ışık akısını (lm) ifade eder ve Şekil 6.9'da gösterildiği gibi tablodan alınır. AC noktaları arasındaki  $r$  mesafesi (m) öklit mesafesi olup Eşitlik 6.3'e göre hesaplanır.  $\alpha$  ve  $\theta$  açıları sırasıyla Eşitlik 6.4 ve Eşitlik 6.5 kullanılarak bulunur.

$$r = \|AC\| = \sqrt{(C_x - A_x)^2 + (C_y - A_y)^2 + (C_z - A_z)^2} \quad (6.3)$$

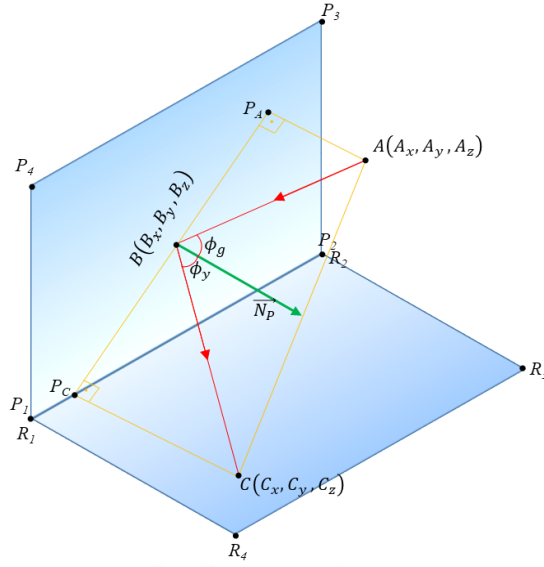
$$\alpha = \tan^{-1} \left[ \frac{(C_y - A_y)^2}{(C_x - A_x)^2} \right] \quad (6.4)$$

$$\theta = \cos^{-1} \left[ \frac{(C_z - A_z)}{r} \right] \quad (6.5)$$

Bu şekilde yansımalar dikkate alınmadan bir mahalde aydınlatma hesaplanabilir. Ancak kapalı alanlarda, aydınlatmaya doğrudan gelen ışınların yanında diğer düzlemlerden (duvarlardan) yansıyor gelen ışınlar da dâhil edilmelidir. Mahal çokgenleri oluşturulurken çokgenlerin her bir kenarı bir duvar olarak düşünülmektedir.

Bir noktadaki ışık şiddetini hesaplarken herhangi bir duvardan gelen yansımayı hesaplamak için; Şekil 6.12'te gösterildiği gibi, A noktasından çıkıp P düzlemi üzerindeki B noktasından yansıdıktan sonra R düzlemi üzerindeki C noktasına varan ışının  $\alpha$  ve  $\theta$  açılarını ( $\alpha_{\overline{AB}}$  ve  $\theta_{\overline{AB}}$ ) ve A ve C noktası arasındaki ışının toplam aldığı yolu ( $\sum r = r_{\overline{AB}} + r_{\overline{BC}}$ ) bulmak gerekecektir.

$A(A_x, A_y, A_z)$  noktasından gelen ışın,  $P_1, P_2, P_3$  ve  $P_4$  köşe noktalarına sahip düzlemde  $B(B_x, B_y, B_z)$  noktasına  $\phi_g$  açısıyla gelip  $\phi_y$  açısıyla yansıdıktan sonra,  $R_1, R_2, R_3$  ve  $R_4$  köşe noktalarına sahip düzlemde  $C(C_x, C_y, C_z)$  noktasına düşmektedir. A (armatür noktası) ve C (aydınlatma hesabının yapılacağı nokta) bilindiğine göre B noktası (yansıma noktası) aşağıdaki şekilde bulunabilir.



Şekil 6.12. Işığın bir düzlemde yansımaları.

B noktasının kartezyen koordinatlarda bileşenleri  $(B_x, B_y, B_z)$ 'dir. Bu bileşenleri bulabilmek için, A ve C noktalarının bu düzlemdeki izdüşümü olan  $P_A$  ve  $P_C$  noktalarının bulunması gerekir. Birbirine dik iki düzlem üzerinde yatan herhangi iki vektörün skaler çarpımı 0'a eşit olduğundan oluşturulacak  $(P_A P_C C A)$  dik yamuk düzlemi üzerindeki vektörler için Eşitlik 6.6 ve Eşitlik 6.7'deki bağıntılar kurulur.

$$\overrightarrow{P_A A} \cdot \overrightarrow{P_1 P_2} = 0$$

$$\overrightarrow{P_A A} \cdot \overrightarrow{P_2 P_3} = 0 \quad (6.6)$$

$$\overrightarrow{P_A P_1} \cdot \overrightarrow{N_P} = 0$$

$$\overrightarrow{P_C C} \cdot \overrightarrow{P_1 P_2} = 0$$

$$\overrightarrow{P_C C} \cdot \overrightarrow{P_2 P_3} = 0 \quad (6.7)$$

$$\overrightarrow{P_C P_1} \cdot \overrightarrow{N_P} = 0$$

Kurulan bu üçlü denklem sistemlerinden  $P_A$  ve  $P_C$  noktaları bulunur. Burada  $\overrightarrow{N_P}$ , P düzlemi üzerindeki normal vektördür. Gelen ve yansıyan ışın açıları  $(\phi_g, \phi_y)$  Şekil 6.12'de görüldüğü gibi birbirine eşit olduğundan  $(P_C C A)$  ve  $(P_A A B)$  üçgen benzerlikleri kullanılarak  $B(B_x, B_y, B_z)$  noktasını bulmak için Eşitlik 6.8 yazılabilir.

$$\frac{|P_A A|}{|CP_C|} = \frac{B_x - P_{Ax}}{P_{Cx} - B_x} = \frac{B_y - P_{Ay}}{P_{Cy} - B_y} = \frac{B_z - P_{Az}}{P_{Cz} - B_z} \quad (6.8)$$

Bu şekilde B noktası da bilindiğine göre bir ışının A noktasından çıkıp B noktasından yansdıktan sonra C noktasına varana kadar toplam aldığı yol öklit bağıntısından sırasıyla Eşitlik 6.9, Eşitlik 6.10 ve Eşitlik 6.11'e göre hesaplanmaktadır.

$$r_{\overline{AB}} = \|\overline{AB}\| = \sqrt{(B_x - A_x)^2 + (B_y - A_y)^2 + (B_z - A_z)^2} \quad (6.9)$$

$$r_{\overline{BC}} = \|\overline{BC}\| = \sqrt{(C_x - B_x)^2 + (C_y - B_y)^2 + (C_z - B_z)^2} \quad (6.10)$$

$$\sum r = r_{\overline{AB}} + r_{\overline{BC}} \quad (6.11)$$

B noktasından yansıyan ışının  $\alpha_{\overline{AB}}$  ve  $\theta_{\overline{AB}}$  açılarını hesaplamak için yine üçgen bağıntılarından Eşitlik 6.12 ve Eşitlik 6.13 yazılabilir.

$$\alpha_{\overline{AB}} = \tan^{-1} \left[ \frac{(B_y - A_y)^2}{(B_x - A_x)^2} \right] \quad (6.12)$$

$$\theta_{\overline{AB}} = \cos^{-1} \left[ \frac{(B_z - A_z)}{r_{\overline{AB}}} \right] \quad (6.13)$$

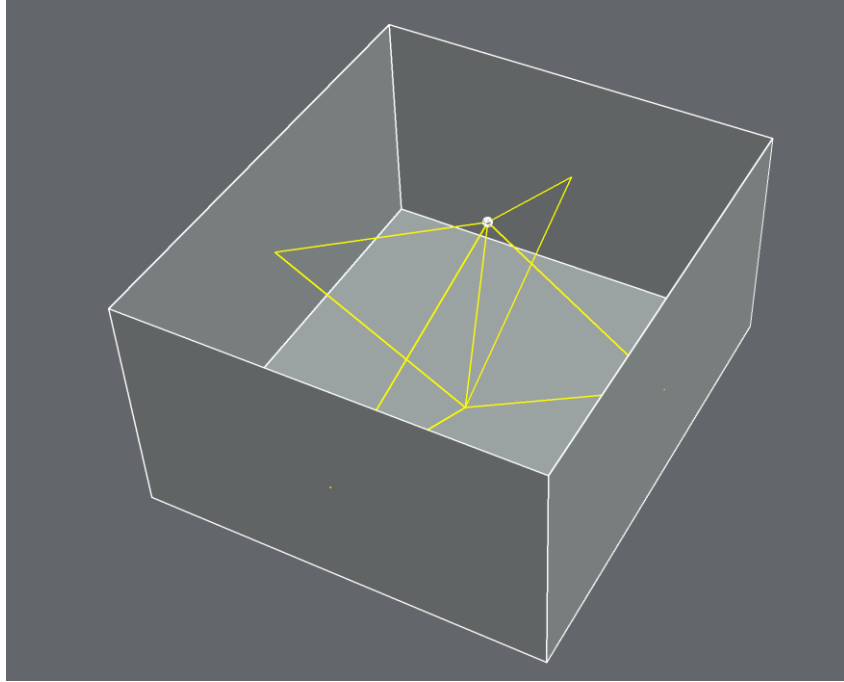
Yansıyan ışığın şiddeti yansımadan dolayı belli bir oranda ( $\rho$ ) azalır. Dolayısıyla A noktasından çıkıp P düzlemi üzerindeki B noktasından yansdıktan sonra C noktasına varan ışının C noktasındaki aydınlatma hesabı Eşitlik 6.14'teki gibi olacaktır.

$$E_C = \frac{\rho I_{\alpha_{\overline{AB}}, \theta_{\overline{AB}}} \cos \theta}{(r_{\overline{AB}} + r_{\overline{BC}})^2} \quad (6.14)$$

Burada  $E_C$ , C noktasındaki aydınlatma şiddeti (lx),  $\rho$  yansıtma katsayısı,  $I_{\alpha_{\overline{AB}},\theta_{\overline{AB}}}$  armatürün  $\alpha$  ve  $\theta$  açılarındaki ışık akısı (lm),  $r_{\overline{AB}} + r_{\overline{BC}}$  ışığın yansıma boyunca aldığı toplam yolu (m) ifade eder.

Yansıtma katsayısı ( $\rho$ ) her renk ve malzeme için farklılık göstermektedir [138,139]. Varsayılan hesaplamalarda yansıtma katsayıları, tavanlar için 0.8 (açık beyaz), duvarlar için 0.5 (alçı sıva) ve zeminler için 0.3 (parke) olarak değerlendirilmiştir.

Bir odada herhangi bir C noktasında toplam aydınlatma hesabı yapılırken önce Eşitlik 6.2’de verildiği gibi doğrudan aydınlatma hesabı yapılır. Daha sonra C noktasına odadaki her bir duvardan gelen yansıma aydınlatmaları Eşitlik 6.14’e göre yapılarak toplam aydınlatmaya eklenir. Bu işlemler her bir armatür için tekrarlanır. Tamamı toplanıp C noktasındaki toplam aydınlatma hesabı yapılmış olur. Şekil 6.13’te bir odadaki tüm duvarlardan yansıyan ışık yansıması temsili olarak gösterilmiştir. Bir odadaki ortalama aydınlatma hesabını  $E_{avg}$  hesaplamak için odada n tane noktada (örneğin 1000 nokta) bu işlemler tekrarlanır ve bunların ortalaması alınarak odanın ortalama aydınlatma hesabı yapılmış olur.



Şekil 6.13. Işığın duvarlardan yansıması.

### 6.2.3. Aydınlatma Tasarımı

Aydınlatma tasarımında amaç, bir mahale uygun aydınlatmayı sağlayacak armatür tipi, sayısı ve konumlarını belirlemektir. Mahal tipi, sınırları (eni-boyu) ve yüksekliği bilinen bir oda için aydınlatma tasarımı Algoritma 6.2’de verilen adımlara göre yapılmaktadır.

#### Algoritma 6.2: Mahal Tipine Göre Aydınlatma Tasarımı

**Girdi:** Mahal-Tipi, En-Boy-Yükseklik, İstenen Aydınlatma Seviyesi

*Mahal-Tipi tefriş sembollerinden puanlama vasıtasıyla öğrenilmektedir.*

*En-Boy oluşturulan Mahal-Çokgeninden ortalama olarak alınmaktadır.*

*Konutlarda varsayılan mahal yüksekliği 280 cm’dir. (Parametre değişebilir)*

**Çıktı:** Mahal için uygun aydınlatma tasarımı

#### Başla

Belirlenmiş Mahal-Tipine göre aydınlatma armatürü tipini ve istenen en az aydınlatma seviyesini Çizelge 6.4’e göre belirle

Öneri seçmek için nokta sayısı=100 olacak şekilde aydınlatma hesabı yapılacak  $C_{i,j}$  noktalarını belirle. (Algoritma 6.3)

Mahal ebatlarına ve armatür tipine göre **Adet-öneri-listesi** oluştur. (Algoritma 6.4)

*Not: Adet-öneri-listesi armatür sayısına göre küçükten büyüğe sıralıdır.*

#### Tekrarla (ikili arama)

*Seçilen-öneri = Adet-öneri-listesi’nin ortasından bir öneri al*

*Sol-Liste = Seçilen-öneri indeksinin solunda kalan öneriler*

*Sağ-Liste = Seçilen-öneri indeksinin sağında kalan öneriler*

*Seçilen-öneri’deki her bir armatür için aydınlatma hesabı yapılacak her bir  $C_{i,j}$  noktasında Eşitlik 6.2’ye göre aydınlatma hesabı yap ve  $E_{avg}$  (ortalama aydınlık seviyesini hesapla).*

*Eğer hesaplanan ortalama aydınlık seviyesi ( $E_{avg}$ ), istenen en az aydınlık seviyesini ( $E_{min}$ )  $\pm\%20$  tolerans ile sağlıyorsa **Aydınlatma Hesabı**’na git.*

**Eğer** ( $E_{avg} < E_{min}$ )

| *Adet-öneri-listesi = Sağ-Liste*

**Değilse**

| *Adet-öneri-listesi = Sol-Liste*

**Eğer-son**

#### Tekrara git

#### Aydınlatma Hesabı:

Nokta sayısı=1000 olacak şekilde aydınlatma hesabı yapılacak noktaları belirle. (Algoritma 6.3)

*Not: Daha hassas hesaplama için nokta sayısı artırılabilir.*

*Seçilen-öneri’deki her bir armatür için aydınlatma hesabı yapılacak her bir  $C_{i,j}$  noktasında Eşitlik 6.14’e göre aydınlatma hesabı yap ve  $E_{avg}$  (ortalama aydınlık seviyesini hesapla).*

#### Bitir

#### 6.2.4. Aydınlatma Armatürü Seçimi

Çizelge 6.4'te mahal tipine göre kullanılması gereken armatür tipi ve minimum aydınlatma seviyesi için bir "Look-Up Table" oluşturulmuştur. Kullanılan armatür tipleri projecilerin tecrübelerinden faydalanılarak hazırlanmıştır. Konut mahallerinde olması gereken en az aydınlık seviye önerileri elektrik mühendisleri odası tarafından belirlenmiştir [140].

Çizelge 6.4. Mahal-Tipine göre aydınlatma armatürü tipi belirleme.

Mahal-Tipi	Aydınlatma Armatürü Tipi	Minimum Aydınlatma Seviyesi
Salon	N	100 lx
Yatak Odası	N	50 lx
Mutfak	CIR	150 lx
Çocuk Odası	N	100 lx
Çalışma Odası	N	200 lx
Banyo-Tuvalet-Lavabo	C	100 lx
Koridor	J	50 lx
Balkon	B	50 lx
...	...	...

#### 6.2.5. Aydınlatma Hesabının Yapılacağı Noktaların Belirlenmesi

Aydınlatma hesabı yapılırken mahal bölgesi (mahal çokgeni), kullanıcının önceden belirlediği nokta sayısı kadar alana bölünür. Eğer armatür sayısı için öneri yapılacaksa bu sayı yaklaşık olarak 100 seçilir. Adet önerisi yapıldıktan sonra gerçek hesaplama yapılırken bu durumda nokta sayısı mahal büyüklüğü de dikkate alınarak yaklaşık olarak 1000 seçilir. Daha hassas hesaplama yapılmak istenirse nokta sayısı birkaç katına kadar arttırılabilir.

Zeminde  $z = 0$  ve tavanda  $z = \text{oda yüksekliği}$  (genellikle 280 cm) olduğu varsayılmaktadır. Çalışma alanında aydınlatma yüksekliği genellikle  $z=85$  cm (veya masa yüksekliği) olarak seçilir. Armatür noktasının  $z$  değeri armatürün kullanım şekline göre belirlenir. Örneğin, armatür "sıva altı" tip ise  $z=\text{oda yüksekliği}$  olarak alınır. Eğer armatür "sıva üstü" ise,  $z = (\text{oda yüksekliği}) - (\text{armatür yüksekliği})$  şeklinde

olur. Aydınlatma hesap noktaları belirlenirken tüm noktaların z değeri çalışma yüksekliğine (genelde 85 cm) ayarlanır.

Aydınlatma hesabını yapılacak noktaların x,y koordinatları seçilirken Algoritma 6.3 kullanılır.

### Algoritma 6.3: Aydınlatma hesabı yapılacak noktaların belirlenmesi

**Girdi:**  $X_{min}, X_{max}$ : Mahal min-x, max-x koordinatı,  
 $Y_{min}, Y_{max}$ : Mahal min-y, max-y koordinatı,  
 $np$ : Aydınlatma hesabı yapılacak nokta sayısı.

**Çıktı:**  $C_{i,j}$ : Aydınlatma hesabı yapılacak noktalar kümesi

```
 $L_x = x_{max} - x_{min}$  // odanın x uzunluğu  
 $L_y = y_{max} - y_{min}$  // odanın y uzunluğu  
 $A = L_x * L_y$  // oda alanı  
 $d = (A / np)^{1/2}$  // np aydınlatma hesabı yapılacak nokta sayısı (parametre)  
 $n_x = (L_x / d)$  // x ekseninde nokta sayısı  
 $n_y = (L_y / d)$  // y ekseninde nokta sayısı  
 $d_x = L_x / n_x$  // x ekseninde noktalar arası mesafe  
 $d_y = L_y / n_y$  // y ekseninde noktalar arası mesafe  
 $x = x_{min} + (d_x / 2)$   
 $y = y_{min} + (d_y / 2)$   
 $i=0, j=0$   
while(  $x < x_{max}$  )  
    while(  $y < y_{max}$  )  
         $C_{i,j} = point(x,y)$   
         $y = y + d_y$   
         $j = j + 1$   
    end-while  
     $x = x + d_x$   
     $i = i + 1$   
end-while
```

### 6.2.6. Armatür Adedi Öneri Listesi Oluşturma

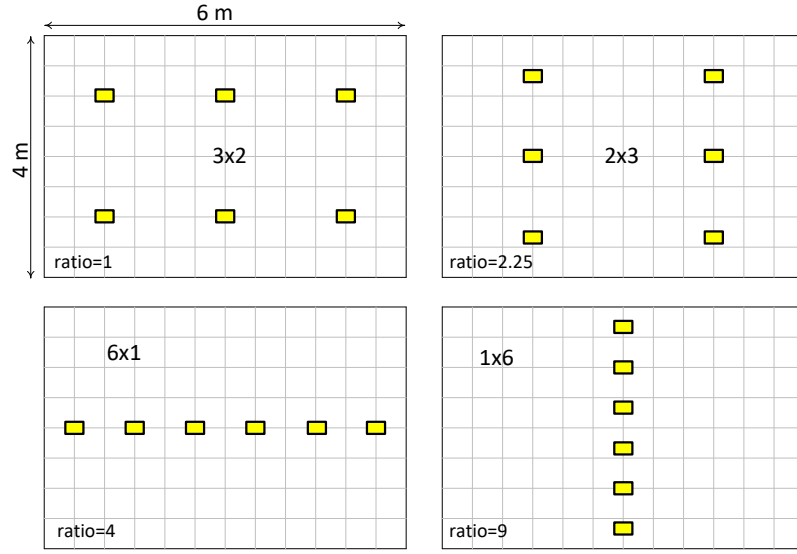
İstenen minimum aydınlatma seviyesi için kaç adet armatürün seçilmesi gerektiği belirlenen armatür tipi, oda boyutları ve duvarların rengi (yansımalar nedeniyle) ile doğrudan ilişkilidir. Bu bölümde kaç adet armatür kullanılabileceğine dair öneriler için

bir liste oluşturulur. Bu listeye “Adet-öneri-listesi” denir. “Adet-öneri-listesi” oluşturmak için öncelikle mahal ve armatürün boyutlarına göre yatay ve dikey olarak kaç adet armatürün sığabileceği hesaplanır. “Adet-öneri-listesi” oluştururken Algoritma 6.4 kullanılır.

<b>Algoritma 6.4: Adet öneri listesi oluşturma</b>	
<b>Girdi:</b> $L_x$ : Mahal x boyu, $L_y$ : Mahal y boyu, $W_{lum}$ : Armatür x boyu, $H_{lum}$ : Armatür y boyu	
<b>Çıktı:</b> Adet Öneri Listesi	
$Adet-Öneril-Listesi = \{\}$	// boş liste
$MaxNumX = L_x/W_{lum}$	// $L_x$ : Mahal x boyu ; $W_{lum}$ armatür x boyu
$MaxNumY = L_y/H_{lum}$	// $L_y$ : Mahal y boyu; $H_{lum}$ armatür y boyu
<b>for</b> $i=1$ <b>to</b> $MaxNumX$	
$r1 = L_x/i$	// x ekseninde armatürler arası mesafe
<b>for</b> $j=1$ <b>to</b> $MaxNumY$	
$r2 = L_y/j$	// y ekseninde armatürler arası mesafe
$ratio = \max(r1,r2)/\min(r1,r2)$	// x ve y eksenindeki armatürler arası mesafe oranı
<b>if</b> $ratio \leq MaxRatio$ <b>then</b>	// kullanıcının belirlediği max mesafe oranı (2.5)
Adet-Öneri-Listesi.Ekle (Öneri(i, j, ratio))	// Listeye ekle
<b>end-if</b>	
<b>end-for</b>	
<b>end-for</b>	

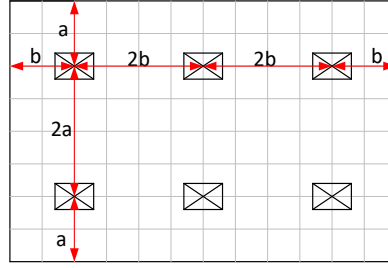
Armatürlerin yatay ve düşey uzaklıkları arasındaki oran *ratio* olarak tanımlanmıştır. Ratio uygun olmayan armatür yerleşimlerini önlemek için kullanılan bir değişkendir. *Ratio* armatürler arasındaki yatay mesafe ile dikey mesafeden büyük olanının küçük olanına oranıdır. Deneyimlerimize göre öneri listesine eklenebilecek en büyük ratio değeri “MaxRatio” 2.5 olarak seçilmiştir. Yani, yatayda ve dikeyde armatürler arası mesafe birbirinin 2.5 katından büyük olamayacaktır.

Örneğin “Adet-öneri-listesi” oluşturmak için Şekil 6.14’te gösterilen 6m x 4m ebatlarında bir mahalde 6 armatür için 1x6, 2x3, 3x2 ve 6x1 olmak üzere 6 armatürün tüm kombinasyonları denenebilir. Bu mahal için 1x6 ve 6x1 armatür düzeni durumlarında  $ratio > 2.5$  olacağı için öneri listesine giremeyecektir. Ancak 2x3 ve 3x2 durumlarının ikisi de listeye girecektir.



Şekil 6.14. Örnek armatür yerleşimleri.

“Adet-öneri-listesi” oluşturulduktan sonra bazı önerilerde örneğin 2x3 ve 3x2 gibi eşit sayıda armatür olacaktır. Bu tür durumlar için listenin sadeleştirilmesi gerekir. Listeyi sadeleştirmek için öncelikle liste armatür sayısına göre küçükten büyüğe sıralanır. Eşit durumlarda, armatürler arasındaki uzaklık oranı (ratio) en az olan listede kalır, diğerleri listeden silinir. Örneğin Şekil 6.14’te gösterilen durumda 3x2 (ratio = 1) olan kalır, 2x3 (ratio = 2.25) listeden atılır. Dolayısıyla böyle bir mahal için 6 armatür gerekiyorsa 3x2 şeklinde yerleştirilmesi önerilecektir. Bu işlemden sonra armatür sayısına göre artan düzende sıralanmış bir *Adet-öneri-listesi* oluşur. *Adet-Öneri-listesi*’nin ortasından ikili aramaya göre bir öneri seçilir. Seçilen öneriye göre aydınlatma, yansıma olmaksızın belirli bir tolerans içinde 100 noktada hesaplanır. Hesaplanan aydınlatma şiddeti istenilenden fazla ise ilk yarının ortasında yeni bir öneri seçilir, aksi takdirde listenin ikinci yarısının ortasında yeni bir öneri seçilir. Bu işlem yinelemeli olarak tekrarlanarak belirlenen toleranslar dahilinde bir "öneri" seçilir. Tolerans, istenen aydınlatma seviyesinin tipik olarak %20 ila %25 altındadır. Aydınlatmanın %20 ila %25’inin yansımalarla yapıldığı görülmüştür. Seçilen öneri mahal çokgeninin içine düzenli bir şekilde dağıtılarak armatürler olması gereken yere otomatik olarak yerleştirilmiş olur. Örneğin (3x2) öneri seçilirse armatürler Şekil 6.15’da gösterildiği gibi düzgün bir şekilde dağıtılır.

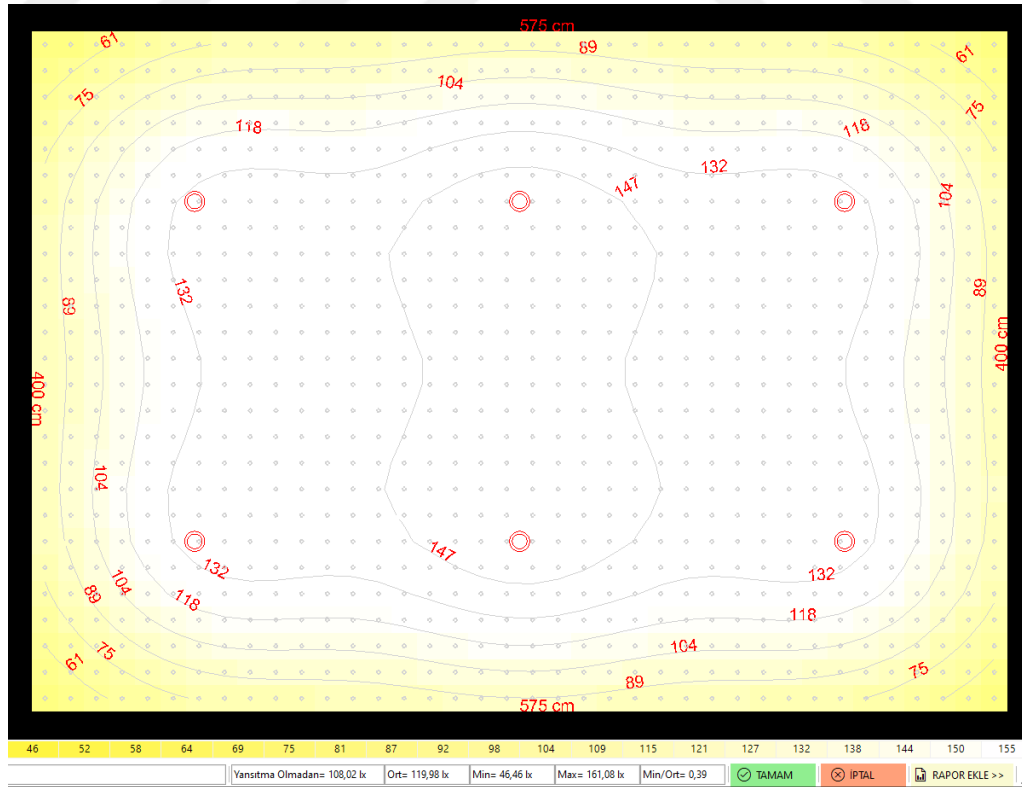


Şekil 6.15. Armatürlerin düzenli bir şekilde dağıtımı.

Parametreleri Çizelge 6.5’te verilmiş örnek bir mahalde 1000 noktada yapılan aydınlatma hesabı sonucu oluşan eş aydınlık eğrileri Şekil 6.16’de gösterilmiştir.

Çizelge 6.5. Örnek mahal aydınlatma parametreleri.

Mahal Ebatları	575cm×400cm, H=280cm, çalışma yüksekliği = 85cm
Önerilen Armatür Sayısı	3x2 (6 Adet) ratio =1.04
Önerilen Armatür	10 W 865 lümen spot armatür
Aydınlatma hesabı	Yansıtma olmadan: 108.02 lx (100 nokta) Yansıtma dahil: 119.98 lx (1000 nokta)



Şekil 6.16. Örnek bir mahalde aydınlatma

### **6.3. BAĞLANTI HATLARININ OLUŞTURULMASI**

Bu kısımda, elektrik malzeme sembolleri konulmuş bir kat planında elektrik plan çizimini tamamlamak için linye ve sortilerin otomatik olarak oluşturulması anlatılmıştır.

Kat dağıtım panolarından elektrik malzemelerinin bağlandığı buata kadar olan elektrik hattına linye hattı, buat ile elektrik malzemesi arasındaki bağlantıya ise sorti hattı denir. Aydınlatma armatürlerinin bağlandığı linyelere aydınlatma veya ışık linyesi, prizlerin bağlandığı linyelere ise priz linyesi denir. Buat ile armatürler arasındaki bağlantı aydınlatma veya ışık sorti, buat ile priz arasındaki bağlantı ise priz sortidir.

#### **6.3.1. Ön İşlemler**

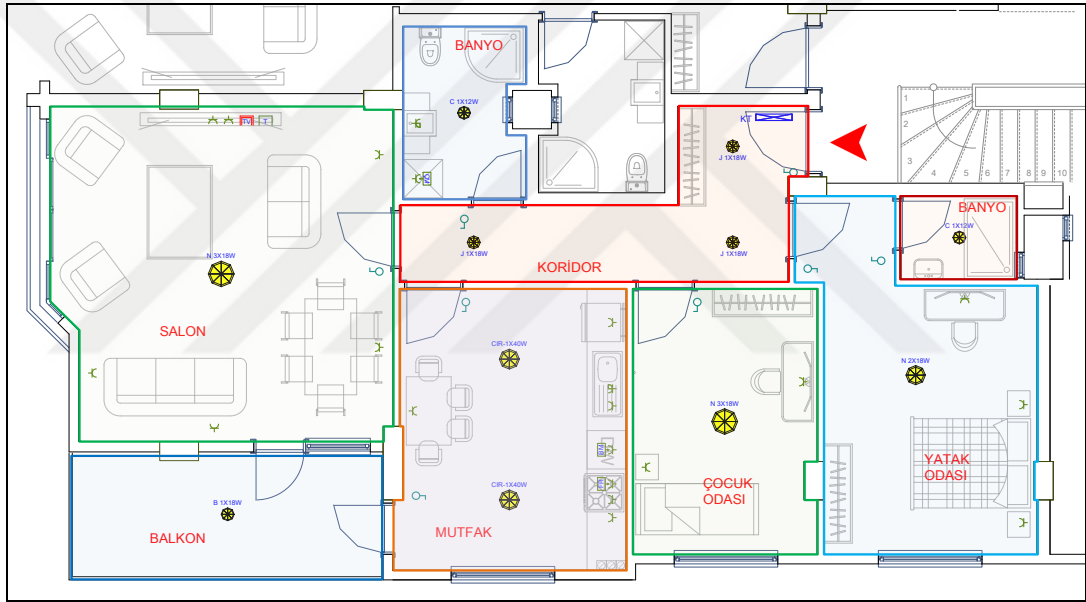
Otomatik linye ve sortilerin çizilme aşamasına başlamadan önce yapılması gereken işlemler Algoritma 6.1’de (otomatik elektrik malzeme sembolü yerleştirme) açıklanmıştır.

Tefriş sembolleri düzenli olan bir kat planında, elektrik malzeme sembollerinin tefriş bağımlı olarak kat planları üzerine otomatik yerleştirilmesi sağlanmaktadır. Ancak geliştirilen sistem tarafından oluşturulan otomatik yerleştirme işleminden sonra bazen priz veya armatürlerin yeri manuel olarak değiştirilebilir. Bir mahalde, eksik tefriş sembolleri yüzünden veya müşteri isteğiyle manuel olarak fazladan priz veya armatürler konulabilir. Bazen tefriş sembollerinin olup olmadığına bakmadan da plan üzerinde elektrik malzeme sembolleri eklenebilir. Örneğin, koridorlarda elektrikli süpürge ve modem için uygun yerlere birer priz ve gerekiyorsa balkona kombi için bir priz konulmalıdır.

Koridor, otomatik elektrik tesisat çizimine başlamak için kritik bir mahaldir. Kat planı çizimlerinde çoğunlukla koridorlarda herhangi bir tefriş sembolü bulunmamaktadır. Bu yüzden tefriş tabanlı oda tanıma, koridorlar için pek mümkün olmamaktadır. Bununla birlikte tefriş tabanlı puanlama ile tipi tespit edilememiş bir mahalin, salon ve mutfak gibi diğer mahaller ile ortak kapıları varsa koridor olarak değerlendirilebilir. Fakat bu şekilde koridor tespiti için önce diğer mahallerin tespit edilmiş olması

gerekecektir. Bir başka alternatif ise; *daire dağıtım panoları* manuel şekilde koridora eklenerek koridorlar işaretlemiş olur. Böylece her daire dağıtım panosu sembolünün olduğu mahal alanı koridor olarak kabul edilecektir.

Şekil 6.17’de gösterildiği gibi sınırları belirlenmiş mahallerde bulunan tefriş sembollerine göre otomatik olarak yerleştirilmiş elektrik malzeme sembolleri ve bulunan bu tefriş sembollerine göre puanlanarak tipleri belirlenmiş odalar bulunmaktadır. Otomatik yerleştirilmiş malzeme sembollerine ek olarak odalarda manuel olarak fazladan prizler/armatürler eklenebilir veya otomatik konumlandırılmış prizlerin veya armatürlerin yerleri değiştirilebilir.



Şekil 6.17. Sınırları ve tipleri belirlenmiş mahal çokgenleri (odalar) ve elektrik malzeme sembolleri yerleştirilmiş kat planı.

Bu işlemler tamamlandıktan sonra, elektrik malzeme sembolleri konulmuş bir kat planında elektrik plan çizimini tamamlayacak linye ve sorti hatlarının otomatik olarak oluşturulması tamamlanacaktır.

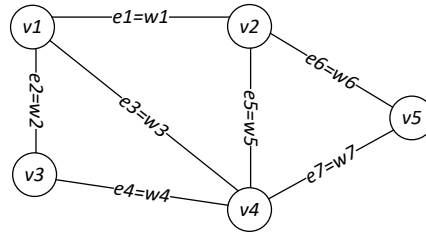
Otomatik elektrik plan çizimine başlamadan önce, kat planları ve elektrik tesisatının bilgisayar ortamında modellenmesi için gerekli veri yapıları anlatılacaktır.

### 6.3.2. Graf ve Ağaç Veri Yapıları

Gerçek hayattaki birçok problem gibi elektrik tesisatı ve kat planları, graf veri yapısı ve graf veri yapılarının özel bir türü olan ağaç veri yapıları ile modellenenir.

Graf veri yapısı, düğümler ve bu düğümleri birbirine bağlayan kenarlardan oluşur. Graflar  $G=(V,E)$  şeklinde ifade edilir. Burada  $V$  düğümleri ( $V = \{v1, v2, \dots\}$ ),  $E$  ise kenarları ( $E \subseteq \{\{v1, v2\} \mid v1, v2 \in V\}$ ) ifade eder [141,142].

Graflar farklı şekillerde isimlendirilmiştir: İki düğüm arasında sadece bir kenar varsa *basit graf*, iki düğüm arasında birden fazla kenar varsa *çoklu graf*, bir düğümden kendine bir kenar (çevrim) varsa *pseudo graf*, kenarların ağırlıkları varsa *ağırlıklı graf*, kenarların yönleri varsa *yönlü graf*, düzlemde birbirini kesmeden kenarları çizilebiliyorsa *düzlemsel graf*, kenarlar düzlemde birbirini kesmeden çizilemiyorsa *üç boyutlu graf*, tüm düğümler arasında doğrudan kenar varsa *tamamlanmış graf* şeklinde isimlendirilir. Şekil 6.18’de basit, yönsüz, ağırlıklı ve düzlemsel bir graf örneği verilmiştir.



Şekil 6.18. Örnek bir graf yapısı.

Graflar üzerinde bir düğümden başka bir düğüme gitmek için geçilen tüm düğümler bir “yol” (path) olarak isimlendirilir. Bu yoldan geçilen tüm kenarların toplam ağırlığı “yolun ağırlığı” olarak isimlendirilir. Bir yoldan geçerken bir düğümden bir kez geçiliyorsa buna *basit yol* denir. Bir düğümden başka bir düğüme gitmek için birçok yol olabilir. Bunların en kısası (en az toplam maliyetlisi) “en kısa yol” olarak isimlendirilir. Eğer herhangi bir düğümden başka herhangi bir düğüme gidilebiliyorsa buna *bağlı graf*, eğer gidilemiyorsa buna *ayrık graf* denir. Bir düğüme bağlanan kenar sayısı *düğümün derecesi* olarak adlandırılır. Derecesi en yüksek düğüm aynı zamanda *grafın derecesi* olur.

Graf yapılarında en kısa yol (*shortest path*), iki düğüm arasındaki toplam maliyeti en düşük olan yoldur. Bir şehirden başka bir şehre gitmek için geçilmesi gereken şehirler ve yollar günlük hayattaki bu problemin örneğidir. Klasik en kısa yol problemi Dijkstra algoritması [143] ile çözülebilmektedir.

Eğer bir graf yapısındaki tüm düğümler bağlı ise ve hiç çevrim yoksa bu aynı zamanda bir *ağaç* yapısıdır. Ağaç veri yapısı, bir kök, sonlu sayıda düğüm ve onları birbirine bağlayan dallardan oluşan bir yapıdır.

Bir grafın tüm düğümlerini herhangi bir döngü olmadan mümkün olan en az kenar ağırlığı ile birbirine bağlayan ağaç yapısına *en az yayılan ağaç* (*MST – minimum spanning tree*) denir. Klasik MST problemi Kruskal [144] veya Prim [145] algoritmaları ile çözülebilir.

En az yayılan ağaç yapısındaki toplam uzunluğu daha da azaltmak için aralara fazladan düğümler ve kenarlar eklenerek oluşturulan ağaç yapısına Steiner ağacı (*steiner tree*), fazladan eklenen düğümlere “steiner noktaları” denir. Steiner ağacının en az toplam ağırlıklı olanına SMT (*steiner minimal tree*) denir [146].

Steiner ağacında iki nokta arasındaki maliyet için noktaların birbirine uzaklıkları hesaplanırken öklit uzaklıkları (Eşitlik 6.15) yerine köşeli uzaklıkları (Eşitlik 6.16) kullanılırsa buna *rectilinear steiner ağacı* denir.

$$d_1 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (6.15)$$

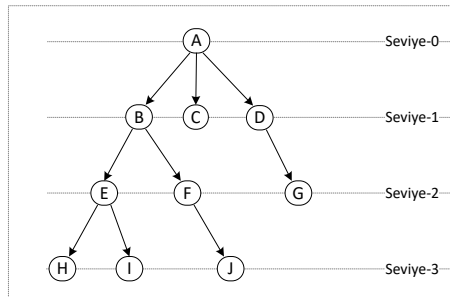
$$d_2 = |x_1 - x_2| + |y_1 - y_2| \quad (6.16)$$

Sadece yatay veya dikey yönlerde hareketin yapılabildiği ortamlarda rectilinear steiner ağacı kullanılır. Elektronik dizayn programları, fiziksel elektronik dizayn otomasyonları, VLSI devreleri ve benzeri alanlar, problemin bu versiyonu için sık kullanılan alanlardandır [147,148].

Gerçek hayatta elektrik tesisatları, su şebekeleri, telefon hatları, bilgisayar ağları, tren ve kara yolları gibi birçok yapı graf yapısındadır. Bu yüzden graf yapıları ile ilgili en kısa yol, en az kapsayan ağaç, steiner ağacı gibi birçok problem modellenmiş ve bunların çözüm yöntemleri geliştirilmiştir.

Ağaç veri yapısı; aile soyağacı gibi hiyerarşik verilerin modellenmesinde kullanılan ve kök, dal, yaprak gibi kavramlara uygun olarak modellenen özel bir graf yapısıdır. Ağacın her bir elemanına düğüm (node), en üstteki düğüme kök (root), iki düğüm arasındaki bağlantıya dal (branch) adı verilir. Bir düğümün bir alt düğüme çocuğu (child), bir üst düğüme babası (parent) denir. Bir düğümün tüm alt düğümleri o düğümün varisleri (descendant), tüm üst düğümleri de o düğümün ataları (ancestor) olur. Kök düğümü, kendisi hariç diğer tüm düğümlerin atasıdır. Çocuğu olmayan düğümlere yaprak (leaf), aynı babaya sahip düğümlere kardeş düğüm (sibling) denir. Bir düğümün çocuk sayısı o düğümün derecesidir. Derecesi en büyük düğüm aynı zamanda ağacın derecesi olur. Bir düğümün köke olan uzaklığı o düğümün derinliği olur. En derindeki düğüm aynı zamanda ağacın derinliğidir. Kökün derinliği ve yaprakların derecesi sıfırdır. Ağaçtaki yaprak sayısı ağacın genişliğidir. Yüksekliği ayarlanmış ağaçlara dengeli ağaç (balanced tree) denir. Dengeli ağaçlarda tüm yapraklar en fazla bir fark ile aynı derinlikte olur [142].

Şekil 6.19'de genel bir ağaç veri yapısı gösterilmiştir. Genel ağaç veri yapılarında çocuk sayısı ve bağlandığı düğümler ile ilgili herhangi bir kısıtlama yoktur. Ancak ağaç yapılarında belirli kısıtlamalar ve kurallar ile farklı ağaç yapıları oluşmuştur. İkili arama ağaçları (binary search tree), kodlama ağaçları (coding tree), sözlük ağaçları (dictionary/trie tree), kümeleme ağaçları (heap tree) bunlardan bazılarıdır.

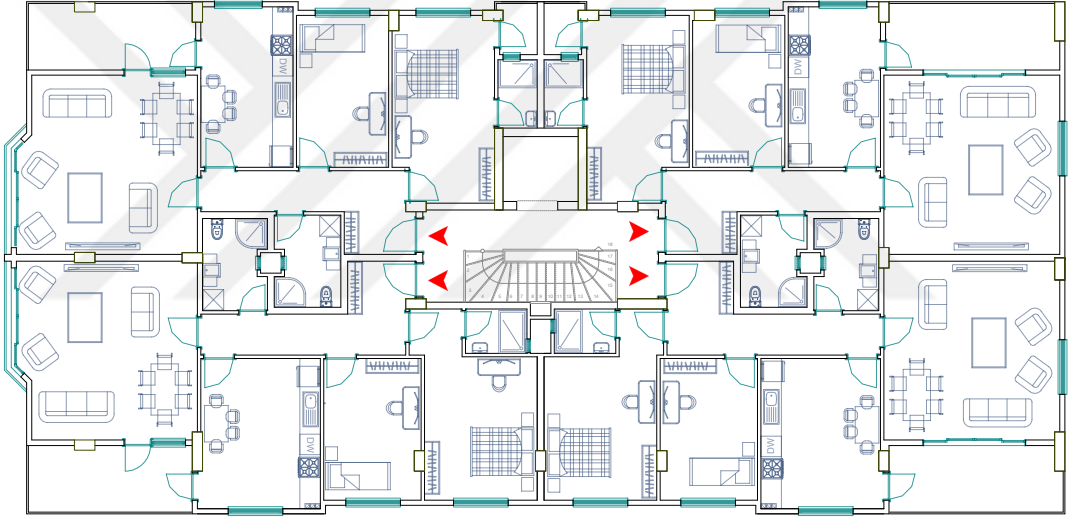


Şekil 6.19. Genel bir ağaç yapısı.

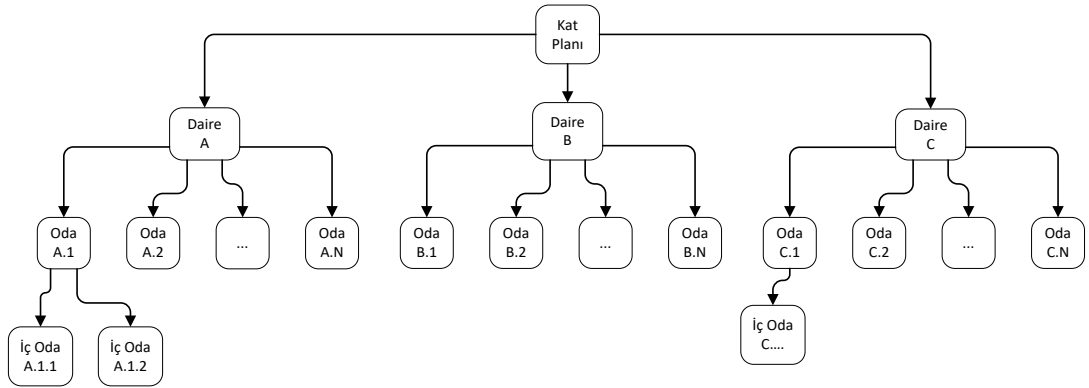
Ağaç veri yapıları gerçek hayatta döngü içermeyen birçok problemin modellenmesinde kullanılabildiği için yazılım dünyasında en sık kullanılan veri yapılarından biridir. Hiyerarşik ve/veya ilişkisel veri bulunduran birçok alanda ağaç veri yapıları kullanılmaktadır.

### 6.3.3. Kat Planı Modelleme

Bina mimari projeleri, kat planları ve diğer bazı kesit/görünüş detaylarından oluşmaktadır. Elektrik tesisat planı için ilgilenilen kısım kat planlarıdır. Kat planları dairelerden ve daireler de odalardan oluşur. Bir mimari projede birden fazla kat planı ve her kat planında Şekil 6.20’de gösterildiği gibi birden fazla daire bulunabilir.



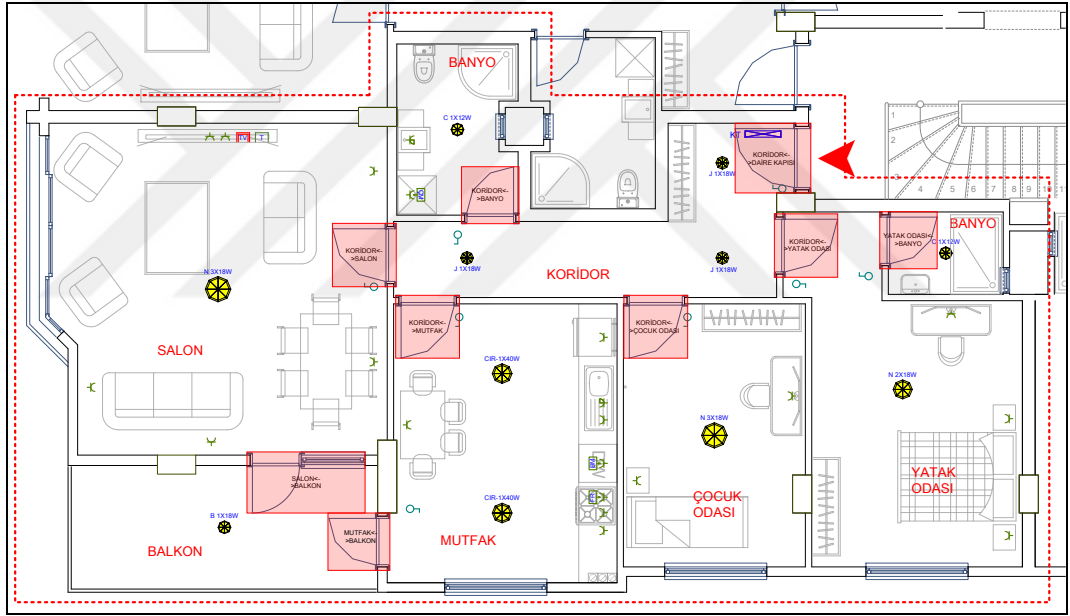
Şekil 6.20. Dört daire içeren örnek bir kat planı.



Şekil 6.21. Kat planının ağaç veri yapısı şeklinde modellenmesi.

Bir kat planındaki elektrik tesisatını otomatik çizebilmek için mekanlar arasındaki ilişkiyi ifade eden kat planı hiyerarşisinin bilinmesi gerekmektedir. Kat planındaki daireler ve odalar arasındaki hiyerarşi Şekil 6.21’de gösterilen bir ağaç veri yapısı şeklinde temsil edilebilir. Kat planlarında, her bir oda bir düğüm olarak modellenmiştir. Kat planındaki her bir dairenin kökü ise o dairenin koridorudur.

Bir dairede elektrik plan çizimi daire dağıtım panolarından başlar. Daha sonra daire dağıtım panoları bina ana dağıtım ve sayaç panolarına bağlanır. Konutlarda, daire elektrik dağıtım panoları koridorlara konulmaktadır. Bu yüzden daireler için kök düğüm koridorlar olmaktadır. Koridorlardan her bir odaya geçiş kapılarıyla sağlandığı için koridor ile odalar arasındaki ağaç dalı (mekânsal ilişki), kapılar olmaktadır.



Şekil 6.22. Kat planı hiyerarşisi için kapıların tespiti.

Şekil 6.22’te kat planından temsili olarak sadece bir daire gösterilmiştir. Bu dairede line ve sortilerin çizilmesi için dairedeki tüm odaların, daha sonra odalar arasındaki mekânsal ilişkinin oluşturulması için geçiş kapılarının tespiti gerekmektedir. Kat planındaki odaların tipleri (isimleri) içlerindeki tefriş sembollerinin sınıflandırılması ve puanlaması ile belirlenmektedir.

Kat planı hiyerarşisinin doğru bir şekilde oluşturulması, kapı tefrişlerinin konumları ile birlikte doğru tespit edilmesine bağlıdır. Tefriş sembollerinin bulunması ve

sınıflandırılması için tasarlanan sınıflandırma modeli ile kapılar zaten bulunmaktadır. Kapı olarak bulunup sınıflandırılan tefriş sembollerinin geometrik verilerinden kapıların konumu ve çerçeveleri vektörel veri dolayısıyla kolaylıkla elde edilebilmektedir. Şekil 6.22’te gösterilen örnek daire planında, kapı olarak sınıflandırılmış tefriş sembollerinin çerçeveleri çizilmiş ve hangi iki mahal arasındaki geçiş olduğu gösterilmiştir. Hiyerarşinin oluşturulabilmesi için koridor olan mahalde daire elektrik dağıtım kutusunun bulunması gerekmektedir. Mimari kat planı hiyerarşisinin oluşturulması için Algoritma 6.5’teki adımlar sırayla yapılır.

#### **Algoritma 6.5: Mimari kat planı hiyerarşisinin oluşturulması.**

**Girdi:** Mahal çokgenleri oluşturulmuş ve elektrik malzeme sembolleri konulmuş kat planı.

**Çıktı:** Kat planı mimari hiyerarşisi (ağaç veri modeli)

#### **Başla**

Tüm kapı tefrişlerini listele (**kapı-listesi**)

*Not: kapıları sınıflandırma sonucu biliyoruz*

Tüm mahalleri (mahal çokgenlerini) listele (**oda-listesi**)

*Not: Mahal çokgenleri kullanıcı tarafından bir kere tanımlanmıştır*

Çokgeni içinde daire elektrik dağıtım kutusu olan mahalleri (odaları) **daire-listesi** içine ata ve **oda-listesi** içinden çıkar

*Not-1: Sadece koridorlarda elektrik dağıtım kutusu olduğunu biliyoruz, koridor daireler için kök düğüm olacaktır.*

*Not-2: Bir katta birden fazla daire olabileceği için birden fazla kök oluşabilir. (Her daire için bir kök)*

**Her bir daire-listesi** içindeki **koridor** için

**Her bir oda-listesi** içindeki **oda** için // koridorlar hariç diğer tüm odalar

**Her bir kapı-listesi** içindeki **kapı** için

**Eğer** kapı dikdörtgeni, oda çokgeni ve koridor çokgenini beraber kesiyorsa

*oda-kapısı = kapı*

*kapı-listesinden kapıyı çıkar*

*odayı koridorun alt düğümü yap (ağaca ekle) ve kalan oda-listesinden çıkar (çıkarılan oda ağaca eklendiği için bir daha bakılmasın)*

**Bitir**

**Bitir**

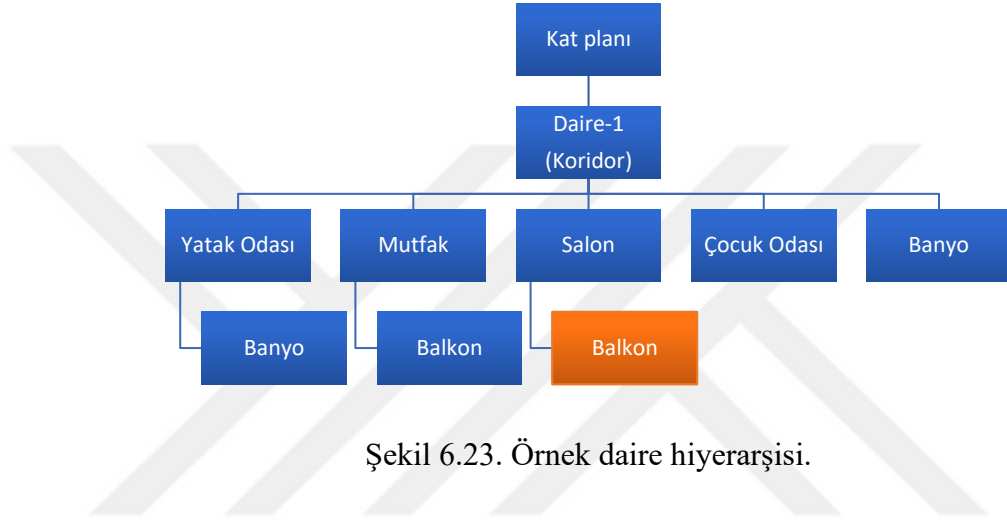
**Bitir**

**Bitir**

**Bitir**

Algoritma 6.5 ile koridorların birinci seviye odaları ağaca eklenmiş olur. Eğer her odayı koridor gibi düşünüp her oda için bu algoritma öz yinelemeli (recursive) olarak tekrarlanırsa her odanın alt odası da (ebeveyn banyosu, balkon, vb.) ağaca eklenmiş olacaktır.

Şekil 6.22’te gösterilen mahal bölgeleri seçilmiş örnek daire, Algoritma 6.5’e girdi olarak verilince oluşacak ağaç veri yapısı Şekil 6.23’de gösterilen gibi olacaktır.



Şekil 6.23. Örnek daire hiyerarşisi.

Şekil 6.22’te gösterilen daire planına dikkat edilirse sadece bir tane balkon olduğu görülecektir. Ancak hem mutfak hem de salon ile ortak kapısı (geçiş) vardır. Ağaç veri yapılarında döngü olmadığı için kat hiyerarşisine iki defa eklenmiştir. Fakat elektrik planı çizilirken bir odaya iki defa elektrik tesisatı çizilemez. Bu yüzden iki odadan sadece birinden planlama yapılmalıdır. Birden fazla geçiş olan odalarda üst oda seçiminde bağlantı için en az kablo kullanımı belirleyici olacaktır.

#### 6.3.4. Elektrik Dağıtım Planının Modellenmesi

Elektrik dağıtımı, panolardan elektrik malzemelerine doğru yapılır. Elektrik dağıtımında bir elektrik malzemesine birden fazla bağlantı yapılamaz. Dolayısıyla elektrik dağıtımında döngüler yoktur. Bir hat üzerinde birden fazla elektrik malzemesi bulunabilir.

Bu kısıtlar dikkate alınarak elektrik tesisatı ağaç veri yapısı şekilde modellenilebilir. Bu model şu şekilde özetlenebilir:

- Kök: Elektrik dağıtım panosu,
- Dallar: Bağlantı hatları (linye/sorti),
- Dügümler: Bağlantı noktaları (buatlar),
- Yapraklar: Elektrik malzemeleri (prizler, aydınlatma armatürleri, anahtarlar)

Elektrik tesisatı kurulumunda kullanılacak kablo miktarının az olması istenir. Ancak gerilim düşümünün de belirlenen sınırlarda kalması gerekir. Kablo uzunluğu ve bağlı malzemelerin gücü, gerilim düşümü hesabı ile doğrudan ilgilidir. Bir fazlı devrelerde sadeleştirilmiş gerilim düşümü hesaplama formülü Eşitlik 6.17’de verilmiştir.

$$\%e = 0.074 \frac{L \cdot P}{S} \quad (6.17)$$

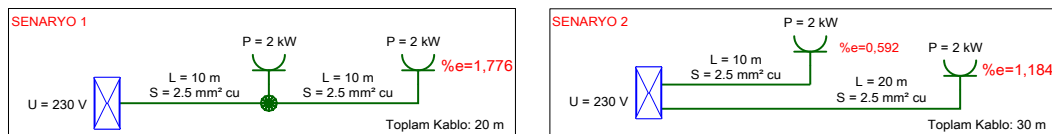
$L$ : Kablo uzunluğu (m),

$P$ : Devreye bağlı toplam güç (kW),

$S$ : Kablo kesiti ( $\text{mm}^2$ ), (tüm linyeler ve priz sortilerde 2.5, ışık sortilerde 1.5),

$\%e$ : Yüzde gerilim düşümü.

Sistemdeki tüm malzemeler mümkünse tek bir kabloya bağlanıp en az miktarda kablo kullanılabilir. Ancak tüm yük bir kabloya bağlanırsa gerilim düşümünün belirlenen limitleri aşmaması için bu kez bağlı kablo kesitinin büyütülmesi gerekecektir. Elektrik iç tesisat planlarında standart olarak priz ve ışık linye kesitleri  $2.5 \text{ mm}^2$ , priz sortileri  $2.5 \text{ mm}^2$  ve ışık sortileri ise  $1.5 \text{ mm}^2$  olarak kullanılmaktadır. Dolayısıyla linye ve sortilerin kesitleri sabittir ve değiştirilmez. Konulmuş güç malzemelerinin güçleri de belli olduğuna göre, gerilim düşümünü belirli sınırlarda tutmak için mecburen bazı yükler dağıtım panolarına farklı kablolar ile bağlanmalıdır. Şekil 6.24’te aynı yükün farklı senaryolar ile bağlanması gösterilmiştir.



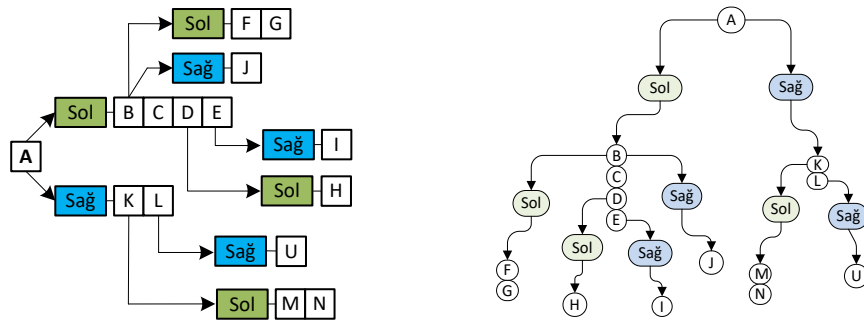
Şekil 6.24. Aynı yükün farklı senaryolar ile bağlanması.

Priz tesisatında, yapı bağlantı kutusundan son prize kadar toplam gerilim düşümü %1.5'ten büyük olamaz. Şekil 6.24'te gösterilen senaryo 1'de iki yük aynı kablo ile dağıtım panosuna bağlanarak toplam 20 m kablo kullanılmış ancak son prizdeki toplam gerilim düşümü Eşitlik 6.17'ye göre hesaplandığında %1.776 çıktığı için bu durum kabul edilemez. Senaryo 2'de ise iki yük farklı kablolar ile dağıtım panosuna bağlanmış, toplam 30 m kablo kullanılmış ancak en uzaktaki prizde toplam gerilim düşümü %1.184 olmuştur.

Eğer elektrik dağıtım planı sadece en az kablo kullanımı dikkate alınarak modellenseydi, en az yayılan ağaç (MST) şeklinde modellenebilirdi. Kablo miktarı tek başına belirleyici olsaydı, MST yapısı gereği tüm malzemeler, tek bir kablo hattına bağlanabilirdi. Bu durum gerilim düşümünde problem oluşturacaktı.

Elektrik tesisat planlarında linyeler duvarlara paralel olarak çizilir. Dolayısıyla elektrik dağıtım ağacını otomatik oluşturabilmek için kat planının da modellenmesi gerekecektir.

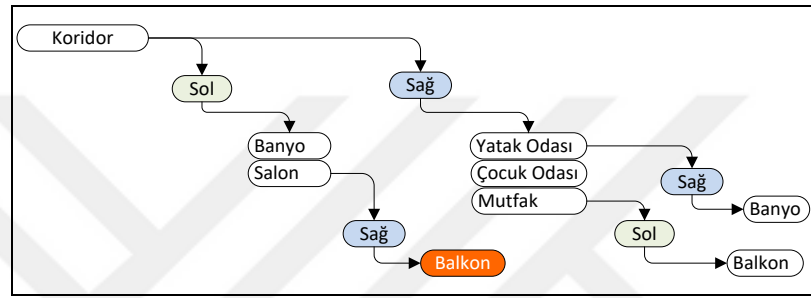
Binalarda elektrik dağıtım planı koridorlardan yapılır. Koridorların çoğunlukla iki yanında (sağında ve solunda) odalar vardır. Dolayısıyla bir odaya elektrik dağıtımını sağdan veya soldan yapılmalıdır. Elektrik dağıtım planını modellemek için özel bir ikili ağaç ve liste karışımı hibrit bir veri yapısı tasarlanmıştır. Bu yapıda ikili ağaç gibi bir düğümün sağ ve sol olmak üzere iki alt düğümü vardır. Düğümler liste şeklinde tasarlanmıştır. Listelerdeki her elemanın da sol ve sağ olmak üzere iki düğümü ve bu düğümlerin de listeleri vardır. Şekil 6.25'da örnek bir ağaç-liste veri yapısı aynı verilerle iki farklı şekilde gösterilmiştir.



Şekil 6.25. Ağaç-liste hibrit veri yapısı.

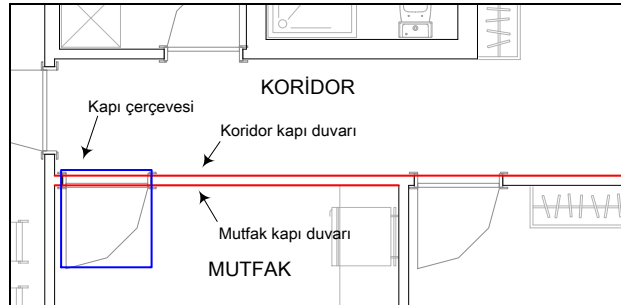
Şekil 6.23'te verilen örnek daire hiyerarşisinin dönüştürülmüş ağaç-liste yapısı Şekil 6.26'de verilmiştir. Bu hibrit veri yapısında koridor, kök olarak ayarlanmıştır. Sağda kalan ve solda kalan odalar da birer listeye eklenmiştir. Bu odaların alt odaları da sağ liste ve sol liste olarak ayrılmıştır. Listelerdeki odalar bağlı oldukları üst odaya olan uzaklıklarına göre küçükten büyüğe doğru sıralanmıştır.

Bir dairede tüm odalar sağda veya solda olabilir. Böyle bir dairenin ağaç-liste yapısında tüm odalar sağda ise sol liste, solda ise bu kez sağ liste boş kalacaktır.



Şekil 6.26. Kat planının ağaç-liste hibrit yapısı.

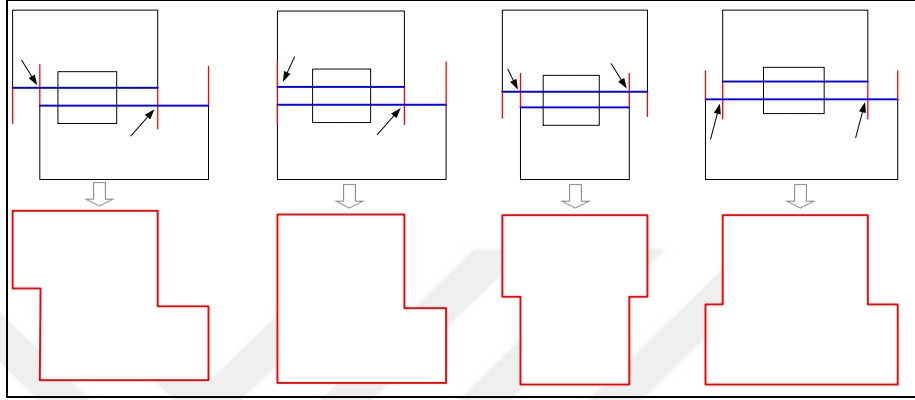
Ağaç-liste yapısı, kat planı hiyerarşisi kullanılarak elde edilir. Ağaç-liste yapısını elde etmek için önce kat planı hiyerarşisindeki her bir odanın çokgeni ile alt odalardaki çokgenler; kendi aralarında ortak olan duvar üzerinden birleştirilir. Birleştirildikten sonra hangi taraftan daha kısa uzunluk ile ulaşırsa o tarafın listesine eklenir. Ortak duvar, iki oda arasındaki kapının bulunduğu duvardır. Şekil 6.27'de iki mahalın ortak duvarlar çizgileri gösterilmiştir.



Şekil 6.27. İki mahalın ortak duvar çizgileri.

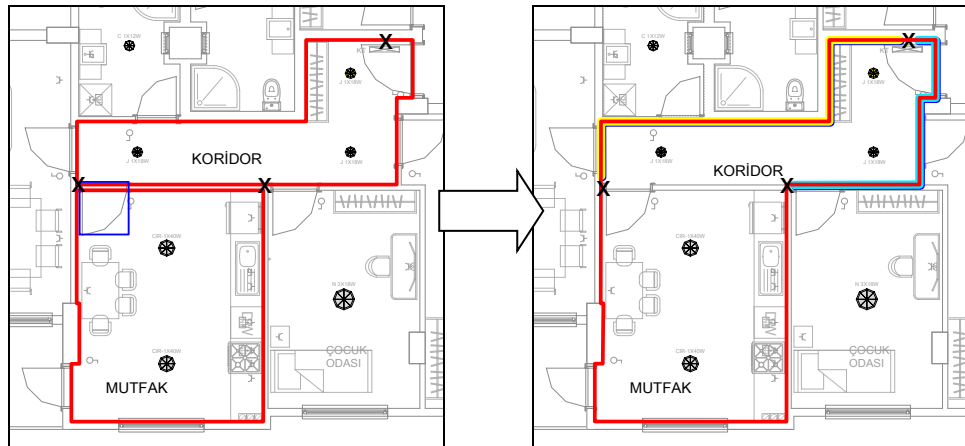
Mahal çokgenlerini birleştirmek için her bir odadaki kapı çerçevelerini (kapı dörtgeni) kesen çokgendeki çizgi parçaları bulunur. Bu çizgi parçaları her bir oda için bir tanedir.

İki çokgeni birleştirmek için iki odanın ortak kapısının olduğu duvar çizgi parçaları ele alınır. Bu iki çizgi parçasının her bir ucundan bu çizgi parçalarına dik olarak yardımcı çizgiler çizilir. Bu çizgi parçaları iki noktada karşı duvar ile kesişir. Bu noktalar, iki çokgeni birleştirme noktaları olacaktır. Şekil 6.28’de mahal çokgeni birleştirme örnekleri verilmiştir.



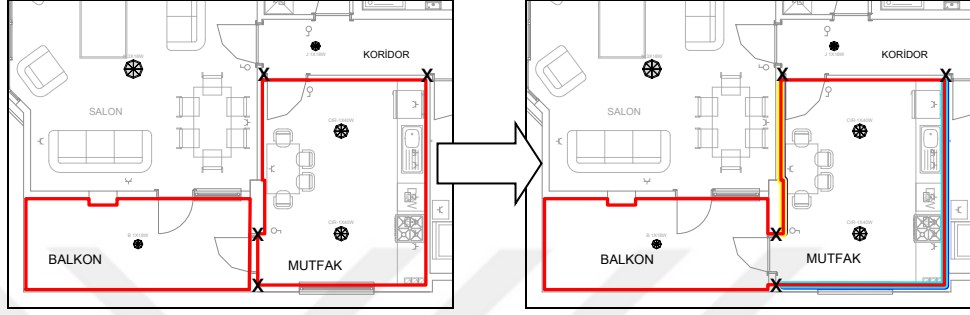
Şekil 6.28. Mahal çokgeni birleştirme örnekleri.

Çokgen birleştirme işlemlerinden sonra, *birleşme noktaları* mahalın ağaç-liste yapısında sol listede mi yoksa sağ listede mi kalacağını hesaplamak için kullanılır. Bunu yapmak için dağıtım panosunun çokgen üzerindeki en yakın noktası kök olarak hesaplanır. Kök noktadan, saat yönünde (sağdan) ve tersi (soldan) yönünde çokgen üzerinde hareket edilir. Bu şekilde bu noktalardan hangisi sol nokta, hangisi sağ nokta olduğu tespit edilir. Sol veya sağ noktalardan hangisi kök noktaya daha yakın ise mahal buna göre sol veya sağ listeye eklenir.



Şekil 6.29 Oda çokgenleri birleştirme.

Şekil 6.29’da koridor ile mutfak mahal çokgenleri birleştirilmiş, kök nokta ve birleşme noktaları işaretlenmiştir. Bu senaryoya göre mutfak, koridorun saat yönünde (sağ) listesine eklenecektir. Benzer şekilde oda ile alt odaların birleştirilmesine örnek olarak Şekil 6.30’de mutfak ile balkon birleştirilmiş ve balkon mutfağın saat yönünün tersi (sol) listesine eklenmiştir.



Şekil 6.30. İç oda çokgenleri birleştirme.

Tüm kat planında ağaç-liste veri modeli oluşturulduktan sonra her bir düğümün listesindeki düğümler, köklerine olan uzaklıklarına göre küçükten büyüğe sıralanır. Bu ağaç-liste yapısı kullanılarak elektrik dağıtım ağacı oluşturulacaktır.

### 6.3.5. Elektrik Dağıtım Ağacının Oluşturulması

Linyeleri çizmek için oluşturulan elektrik dağıtım modelindeki odalar, sol liste ve sağ liste olmak üzere iki ayrı listeye dönüştürüldü. Buradaki amaç, koridorun solunda kalan odalara sol taraftan, sağında kalan odalara sağ taraftan linyelerin çizilmesidir. Oluşturulan bu listeler kök düğüme uzaklıklarına göre sıralanır. Sıralamaktaki amaç önce en yakın odaya linyeleri çizip çizim aşamasında kesişmelerini önlemektir.

Elektrik tesisatı döşenirken her ülkenin bazı yönetmelik kuralları vardır. Elektrik tesisat planları çizilirken bu yönetmelik kurallarına dikkat edilmektedir. Ülkemizde aydınlatma (ışık) kabloları ve priz kabloları ayrı çekilmek zorundadır. Bir priz linyesine en fazla yedi adet priz sortisi bağlanabilirken bulaşık makinesi, çamaşır makinesi, fırın gibi nispeten yüksek akım çeken elektrik malzemelerinin linyeleri müstakil olarak çekilmektedir. Bir aydınlatma linyesine tavsiye olarak en fazla dokuz aydınlatma armatürü bağlanabilirken günümüzde aydınlatma armatür güçleri çok

düşük (çoğunlukla LED armatür 5-15 W) olduğu için bu sayı toplam güç 2000 VA'ı geçmemek şartı ile arttırılabilmektedir. Yapılan çalışmada yönetmelik kurallarına bağlı kalınarak tüm dairelerde aydınlatma tesisatı planı için sağda ve solda olmak üzere iki hat, priz tesisatı çizilirken her oda için en az bir hat olacak şekilde planlama yapılmıştır.

Elektrik tesisat planında linyeler duvarlara paralel olarak çizilir. Sortiler ise iki nokta arasında bir doğru parçası olarak çizilir. Kat planı çizimlerinde odalar çokgen formda ve oda duvarları çokgenin kenarlarıdır. Vektörel işlemler bölümünde açıklanan paralel çıkarma algoritmaları kullanılarak çokgenlerin paraleli çıkarılarak linye hatları için taslaklar oluşturulmaktadır.

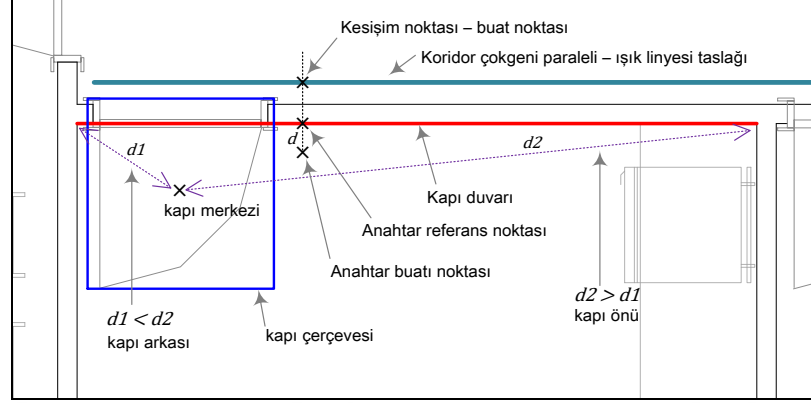
#### **6.3.5.1. Aydınlatma Tesisatı Çizimi**

Bir dairede aydınlatma tesisatı, o dairedeki tüm aydınlatma armatürlerinin ve anahtarlarının aydınlatma linyelerine sorti ile bağlanmasıdır. Elektrik iç tesisleri yönetmeliği gereği bir dairenin büyüklüğüne bakılmaksızın en az iki adet aydınlatma linyesi olmalıdır. Bir aydınlatma linyesine bağlanacak sorti sayısı linyenin kurulu gücü ve gerilim düşümüne göre belirlenmektedir.

Aydınlatma linyeleri koridorlarda çizilir ve tüm oda aydınlatmaları bu linyelere bağlanır. Linye çizimleri duvarlara paralel olarak çizilir. Bunun için koridor çokgeninin belirlenmiş  $d$  (linye aralığı) mesafesinde paraleli çıkarılır. Çıkarılan bu paralel aydınlatma linyesi için taslak olacaktır.

İkinci işlem anahtar bağlantı konumunun belirlenmesidir. Anahtarlar, kapıların açılma yönünde ve kapı kenarından 20 cm mesafede olmalıdır. Anahtar konumunu tespit etmek için kapının açılma yönünün tespit edilmesi gerekmektedir. Kapılar genellikle odaların köşelerine konulur. Kapı, duvarı iki parçaya böler ve her zaman kısa parça kapının arkasında kalır. Bunu tespit etmek için kapı merkez noktasından kapı duvarının iki noktasına olan uzaklıklar hesaplanır. Kısa uç, kapı arkası; uzun uç, kapı önünde kalan kenar noktalarıdır. Dolayısıyla anahtar her zaman uzun kenara doğru kapıdan 20 cm mesafeye konulur. Banyo/tuvalet ve balkon gibi odalarda anahtar

odanın dışında (koridor veya geçiş yapılan oda tarafında) kalır. Şekil 6.31 'te kapı önü ve kapı arkasının tespiti ve anahtar noktasının bulunması gösterilmiştir.



Şekil 6.31. Anahtar koyma noktası tespit etme.

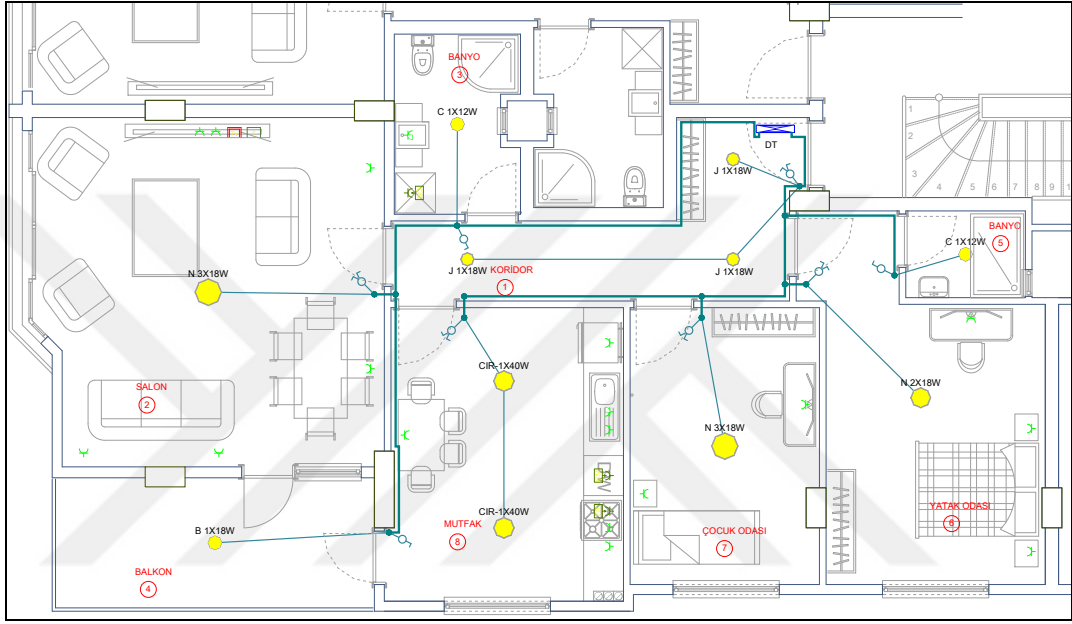
Anahtar için referans noktası tespit edildikten sonra bu noktadan aydınlatma linyesi üzerine bir dik çizilir ve kesişim noktası hesaplanır. Kesişim noktası, bağlantı için buat noktası olacaktır. Kesişim noktası, ışık linyesi taslağına bir nokta olarak eklenir.

Eğer banyo veya tuvalet gibi bir oda değilse referans noktasından oda içine doğru bir  $d$  mesafesi ile anahtarın konulacağı nokta belirlenir. Anahtarın konulacağı noktaya bir buat eklenir. Buna da *anahtar buatı* denir. Anahtar buatı ile buat bağlantı noktası arasına bir hat çekilerek oda içi linye bağlantısı tamamlanmış olur. Eğer banyo veya tuvalet gibi anahtarın dışarıda kalması gereken bir oda ise kesişim noktası aynı zamanda anahtar buat noktası da olacaktır.

Ebeveyn banyosu veya balkon gibi iç odalarda aydınlatma anahtarı odanın dış tarafında olur. İç odalar için ışık linyesi taslağı, geçiş olan odanın ilgili duvarıdır. İlgili duvar, daire için oluşturulan elektrik dağıtım modelindeki ağaç-liste yapısındaki konumuna (sol veya sağ düğüm) göre belirlenir. İç oda ağaç-liste yapısında sol düğümde ise sol duvar paraleli, sağ duvarda ise sağ duvar paraleli koridor paraleli ile kesiştirilerek iç oda bağlantı linyesi oluşturulur.

Aydınlatma linyesi çiziminin tamamlanması için, ışık linyesi taslağı üzerindeki her oda için hesaplanan bağlantı noktaları ışık linyesi taslağı takip edilerek birbirine bağlanır. Böylece aydınlatma linye bağlantısı tamamlanmış olur.

Aydınlatma tesisatı aydınlatma line sortilerinden oluşur. Sortileri oluşturmak için anahtar buatı ve armatürler MST (minimum spanning tree) ile birbirine bağlanarak aydınlatma sorti çizimi tamamlanmış olur. Eğer bir mahalde birden fazla armatür veya bir armatürde örneğin avize gibi birden fazla lamba varsa anahtar otomatik olarak komütatör (ikili) olarak seçilmektedir. Şekil 6.32’te sistem tarafından otomatik oluşturulan aydınlatma tesisatı gösterilmiştir.



Şekil 6.32. Sistem tarafından otomatik oluşturulan aydınlatma tesisatı.

### 6.3.5.2. Priz Tesisatı Çizimi

Elektrik tesisleri yönetmeliğine göre müstakil linyeler hariç, bir priz linyesine yedi adet priz sortisi bağlanabilir.

Normal durumlarda priz sayısı yedi ve altında kaldığı sürece birden fazla odanın priz sortileri ortak linyelere bağlanabilir. Ancak kullanım konforu ve tesisatın güvenliği açısından bu önerilmez. Bir odadaki elektrik arızası veya kaçağı diğer odaları etkilememelidir. Örneğin, konfor açısından mutfaktaki arızalı bir priz yüzünden salondaki televizyon prizinin devre dışı kalmaması gerekir. Yine herhangi bir tadilat anında güvenlik açısından hangi prizlerin hangi linyeye bağlı olduğunun bilinmesi önemlidir. Bu açılardan her odanın priz tesisatı ayrı olacak şekilde tasarlanmıştır.

Bu açıdan her oda için müstakil linyeler hariç, en az bir adet priz linyesi planlanmıştır. Bir odadaki priz linye sayısını hesaplamak için önce oda çokgeni içinde kalan tüm prizler konumları ile bir listeye alınır. Çamaşır makinesi, bulaşık makinesi, fırın gibi prizlerin her biri için müstakil bir linye, diğer prizlerin her yedi tanesi için en fazla bir linye belirlenir. Bir odada prizler için linye sayısı Eşitlik 6.18'e göre hesaplanır. İç odaların prizleri bağlı odalar için sayılır.

$$s = m + \left\lceil \frac{t - m}{x} \right\rceil \quad (6.18)$$

$s$  : Hesaplanan priz linye sayısı

$m$  : Müstakil priz sayısı

$t$  : Toplam priz sayısı

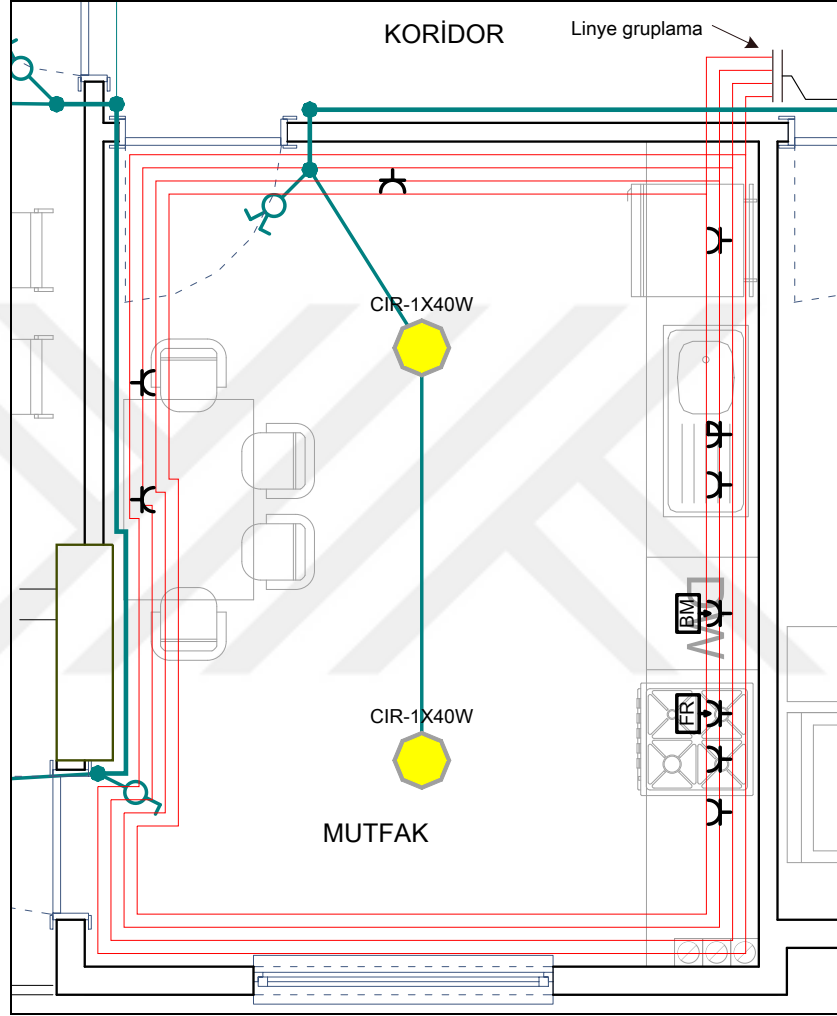
$x$  : Bir linyede olabilecek en fazla priz sortisi sayısı (priz linyeleri için: 7)

Gerilim düşümü hesabı, kablo uzunluğu ve kabloya bağlı toplam yük miktarı ile doğrudan ilişkilidir (Eşitlik 6.17). Dolayısıyla dağıtım panosuna yakın bir odada bir linyeye yedi adet priz sortisi bağlanabilirken dağıtım panosuna uzak bir odada bir linyeye yedi priz sortisi gerilim düşümü hesabını kurtarmayabilir. Dolayısıyla priz linyesine bağlanabilecek en fazla priz sortisi sayısı gerilim düşümü hesabı açısından her oda için adaptif olmalıdır.

Hesaplanan linye sayısına göre her linye için oda çokgenine linye aralığı mesafesi ile taslak paraleller çıkarılır. Eğer bir oda içinde iç odalar için geçen bir aydınlatma linyesi varsa oluşturulan priz linyesi taslak çizimleri buna göre ötelenir. Şekil 6.33'te gösterilen mahalde iki müstakil ve sekiz tane de diğer priz konulmuştur. Eşitlik 6.18'e göre hesaplandığında bu oda için dört tane priz linyesi çekilecektir.

Bir odada prizler odanın her tarafında bulunabilir. Ancak bir odaya bir yerden (soldan veya sağdan) linye bağlantısı yapılır. Bir odanın priz linye bağlantısı elektrik dağıtım modelinde oluşturulan ağaç-liste yapısındaki konumuna göre yapılır. Eğer oda sol düğümde ise sol duvardan, sağ düğümde ise sağ duvardan giriş bağlantısı yapılır. Şekil 6.33'te gösterilen mahal, ağaç-liste yapısında sağ düğümde olduğu için mahale bağlantı girişleri sağdan yapılmıştır.

Her oda için birden fazla priz linyesi çekilebileceğinden çizimde karışıklık olmaması için linyeler gruplandırılarak çizilebilmektedir. Eğer oda ağaç-liste yapısının sol düğümünde ise gruplama odanın sol noktasında, sağ düğümde ise sağ noktasında yapılır.

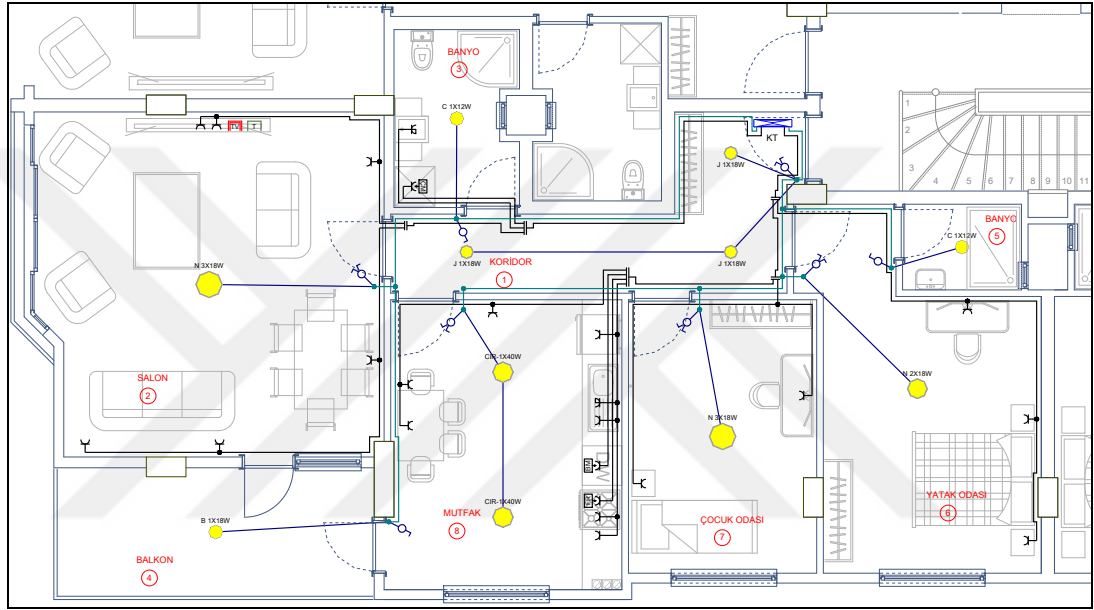


Şekil 6.33. Priz linyeleri için taslak hazırlama.

Oda içindeki prizler bağlantı noktasına soldan veya sağdan uzaklıklarına göre sol liste ve sağ liste olmak üzere ikiye ayrılır. Çizim aşamasında birbirini kesmeyecek şekilde linye çizimi yapılabilmesi için bu listelerdeki prizler bağlantı noktasına uzaklıklarına göre sıralanır. Her bir listedeki prizler uzaktan yakına doğru; müstakil prizler tek başına bir grup, diğerleri bir linyede olabilecek en fazla priz sayısı kadar bir grup olacak şekilde gruplara bölünür. En uzaktaki grup en alt taslak paraleline, en yakındaki grup en üst taslak paraleline bağlanacaktır. Gruplardaki her bir prizden bağlanacağı

taslak polyline nesnesi üzerine birer dikme inerek kesiştirilir. Kesişim noktası buat noktası olacaktır. Eğer priz gruptaki en uzak priz (son priz) ise buat eklenmez. Kesişim noktası ile priz arasındaki çizgi ile priz sorti çizilmiş olacaktır. Buat noktaları ile lineye gruplama arası polyline üzerinden çizilerek priz linyeleri çizilmiş olacaktır.

Sistem tarafından otomatik oluşturulmuş priz ve aydınlatma tesisat çizimleri Şekil 6.34'te gösterilmiştir.



Şekil 6.34. Sistem tarafından otomatik çizilmiş elektrik tesisatı planı.

## BÖLÜM 7

### SONUÇ VE ÖNERİLER

#### 7.1. SONUÇLAR

Bu çalışma ile yapay zekâ alanında popüler bir konu olan derin öğrenme teknikleri kullanılarak konut elektrik iç tesisat planlarının otomatik olarak çizilmesi sağlanmıştır.

Çalışma, gerçek mimari kat planları üzerinde:

- Tefriş sembollerinin sınıflandırılması,
- Sınıflandırmaya bağlı olarak elektrik malzemesi önerme ve uygun yerlere yerleştirme,
- Sınıflandırmaya bağlı mahal tipi tespiti ve otomatik mahal aydınlatma,
- Konulmuş elektrik malzemelerinin otomatik olarak pano bağlantılarını oluşturma

işlemlerini yapmaktadır.

Çalışmanın en önemli kısmını tefriş sembollerinin sınıflandırılması oluşturmaktadır. Çünkü diğer tüm işlemler sınıflandırma üzerinden devam etmektedir.

Tefriş sembollerinin sınıflandırılması için FDS (*Furnishing Dataset*) isminde bir veri seti hazırlanmıştır. Veri setinin hazırlanması için yüzlerce farklı mimari kat plandan 23 farklı sınıfta 1020 adet tefriş sembolü ayıklanmıştır. Bu semboller veri artırma yöntemleri ile 24480 adet sembole dönüştürülmüş, %75 eğitim ve %25 test olacak şekilde homojen olarak ikiye bölünmüştür.

Tefriş sembollerinin sınıflandırılması için Transfer öğrenme ile Inception-v3 modeli kullanılmıştır. Test veri seti üzerinde %98.64, gerçek kat planları üzerinde %97.05 doğruluk ile başarı elde edilmiştir.

Tefriş sembollerinin sınıflandırılmasından sonra, elektrik malzeme sembollerinin mimari kat planları üzerinde uygun yerlere yerleştirilmesi ve bunları daire elektrik dağıtım panolarına otomatik olarak bağlanması için algoritmalar geliştirilmiş ve başarılı bir şekilde uygulanmıştır.

Elektrik dağıtım planının oluşturulması için ağaç ve liste veri yapılarının karışımı hibrit bir veri yapısı olan ağaç-liste veri yapısı modellenmiş ve elektrik tesisatı planı çiziminde başarılı bir şekilde uygulanmıştır.

Kat planındaki her bir mahal için oda çokgeni oluşturulduktan sonra aşağıdaki işlemler otomatik olarak yapılmaktadır:

- Mahal çokgenleri içinde kalan tüm blok nesnelere (tefrişler) listelenir.
- Blok nesnelere belirlenen çözünürlük ve özelliklerde görüntüye dönüştürülür.
- Her bir bloğun hangi tefrişe ait sembol olduğunu belirlemek için oluşturulan görüntü modellenen derin sinir ağına verilerek tanımlanır.
- Tanımlanan tefrişe uygun elektrik malzeme sembolleri uygun duvarlara uygun açılar ile yerleştirilir.
- Tanımlanan tefrişlerden puanlama yapılarak mahal tipi belirlenir.
- Seçilmiş mahal çokgeninin koordinatları bilindiği için mahalın ebatları hesaplanır.
- Mahal tipine ve büyüklüğüne göre uygun armatür tipi ve sayısı seçilerek aydınlatma tasarımı yapılarak armatürler olması gereken yerlere yerleştirilir.
- Kat planı hiyerarşisi oluşturularak mekânsal ilişkiler belirlenir.
- Mekânsal ilişkiler kullanılarak elektrik dağıtım ağacı oluşturulur.
- Aydınlatma ve priz tesisatı çizdirilir.

Bildiğimiz kadarıyla, elektrik malzeme sembollerini otomatik olarak yerleştirip buna uygun elektrik planı çiziminde ilk kez derin ESA ve transfer öğrenmesi kullanmıştır.

Tefriş sembolü veri seti ve test kat planları dxf, pdf ve jpg formatlarında <https://www.kaggle.com/bayramakgl/furnishing-dataset> adresinde herkese açık olarak paylaşılmıştır.

Tefriş veri setine bir sınıflandırıcı model oluşturmak için .Net ortamında Microsoft ML.NET Image Classification API'si kullanılmıştır. Önerilen modelin başarısına göre tatmin edici sonuçlar elde edilmiştir. Bu çalışma, elektrik plan çizimleri alanında yapay zekâ kullanımını teşvik edecektir.

Elektrik malzeme sembollerinin başarılı bir şekilde otomatik olarak yerleştirilmesi, tefriş sembollerinin doğru tanınmasına bağlıdır. Dolayısıyla önerilen sistemin başarısının sınıflandırma başarısı ile aynı olduğu söylenebilir. Önerilen sistem için veri setini kendimiz oluşturduğumuz için daha önceki sembol tanıma çalışmaları ile karşılaştırmak doğru olmayacaktır. Yine de %97,05'lik başarı oranı oldukça tatmin edici bir sonuçtur.

Bu çalışma ile tefriş sembollerinin tanınmasına dayalı olarak elektrik planının otomatik çizilmesi için yeni bir vizyon sunulmuştur.

Bu çalışmanın öncesi olan çalışmamızda, tüm hesaplamaların ve dökümlerin tesisat çizimine entegre edilmesi sağlanmıştı [3]. Dolayısıyla bu çalışma ile elektrik tesisat planlarının kat planı üzerinde otomatik yapılması da sağlanarak elektrik iç tesisat planlarının bazı kısıtlar ile otomatik olarak yapılması sağlanmıştır.

Günümüzde elektrik tesisat projeleri çoğunlukla Autocad veya başka standart CAD programları ile çizilmektedir. Klasik yöntemler ile büyükçe bir elektrik projesinin çizimi ve hesaplamaları günler alabilirken geliştirilen bu sistem sayesinde elektrik tesisat çizimi (tüm hesaplamalar ve dökümler dâhil) en az hata payı ile bir saatten daha kısa bir sürede halledilmektedir.

Bu tez ile bir *gerçek dünya problemine* çözüm getirilmiştir. Manuel olarak yapılan gerçek elektrik tesisat plan çizimlerinde harcanan uzun süreler dikkate alındığında çalışmamızda elde edilen çizimlerdeki %2-3 oranındaki hataların proje üzerinde manuel olarak düzeltilmesi çok kısa sürede yapılabilmektedir. Dolayısıyla bu çalışma sonucunda oluşturulan model hem zaman açısından hem de maliyet açısından oldukça verimlidir.

## 7.2. KISITLAR

Binaların elektrik tesisat projeleri, kuvvetli akım tesisatı ve zayıf akım tesisatı olmak üzere temelde ikiye ayrılır. Çalışmamız sadece kuvvetli akım aydınlatma ve priz tesisatlarını kapsamaktadır.

Önerilen sistem, vektör tabanlı gerçek dünya konut mimari kat planları üzerinde çalışır. Tefriş sembolleri blok şeklinde olmalıdır. Blok şeklinde çizilmemiş tefriş sembolleri tespit edilmeyecektir.

Bu çalışmada kullanılan tefriş sembolleri, sadece konutlarda yaygın olarak kullanılan tefriş sembolleri arasından seçilmiştir.

Şofben, kombi, klima, hidrofor gibi bazı elektrik malzeme sembollerinin tefriş bağımlı olmadıkları için manuel olarak konulması gerekecektir.

Eğer bir mimari projede birden fazla kat planı varsa her bir kat planı üzerinde ayrı ayrı çalışılması gerekmektedir. Mimari yapı birden fazla kat planından oluşuyorsa katlar arası elektrik bağlantısı manuel olarak yapılmalıdır.

Koridorlar elektrik dağıtım panolarının konulması gereken mahallerdir. Ancak elektrik dağıtım panosu tefriş tabanlı konulamaz. Her dairedeki daire elektrik dağıtım panolarının koridorlarda uygun yerlere manuel olarak konulması gerekmektedir.

Kat planında kapı tefrişleri yoksa kat planı hiyerarşisi otomatik oluşturulamayacağı için elektrik plan çizimi yapılamayacaktır.

## 7.3. ÖNERİLER VE SONRAKİ ÇALIŞMALAR

Bu tez çalışması kapsamında veri seti için sadece konut binalarında en sık kullanılan tefriş sembolleri toplanmıştır. Diğer yapılar (fabrikalar, atölyeler, iş yerleri, hastaneler, okullar, oteller, vb.) için kullanılan tefriş sembolleri toplanarak daha kapsamlı bir veri seti oluşturulabilir. Bu şekilde diğer yapılar için de tefriş tabanlı olarak otomatik elektrik malzeme sembollerini olması gereken yerlere yerleştirilebilir.

Tefriş sembolü sınıflandırması, temelde bir görüntü sınıflandırma problemidir. Tefriş sınıflandırma için Transfer öğrenme ile Inception-v3 modeli kullanılmıştır. Geçmişten günümüze görüntüleri sınıflandırmak için pek çok çalışma yapılmıştır. Sınıflandırma başarısını artırmak için farklı ince ayarlar ile farklı (yeni veya önceden eğitilmiş) ESA modelleri (ResNet [110], MobileNet [116] vb.) test edilebilir.

Her ne kadar yapı sektöründe piksel tabanlı kat planları kullanılsa bile vektör tabanlı kat planları çalışma anında piksel görüntülere dönüştürülüp nesne tespiti konusunda başarılarını kanıtlamış Faster R-CNN [126] veya YOLO [128] gibi derin ağlar kullanılarak tefriş sembolleri (blok yapısında olmasa bile) tespit edilebilir ve sınıflandırılabilir.

Tam otomatik elektrik plan çizimine yaklaşmak için mahallerin kat planları üzerinde manuel seçilmesi yerine otomatik olarak ayrıştırılması üzerine çalışılabilir. Bu bizi tam otomatik elektrik planı çizimlerine bir adım daha yaklaştıracaktır.

CAD uygulamalarında çizim sürecinin otomatikleştirilmesi, çizimden mühendislik hesaplamalarının elde edilmesini içerir. Sonuç olarak bu çalışmanın bir sonraki adımı, birden fazla kat planından oluşabilen elektrik tesisat projelerinde tüm daire içi elektrik dağıtım panolarının sayaç panolarına otomatik olarak bağlanması ve tüm hesaplamaların ve hesap dökümlerinin aynı şekilde otomatik yapılmasını sağlamak olacaktır.

## KAYNAKLAR

1. Orak, İ. M. and Akgul, B., "Fast and simple computer aided internal electrical wiring project design & integration of calculations", *Procedia Technology*, 1: 280–285 (2012).
2. İnan, N. and Yıldırım, T., "Interdisciplinary Relations and Synchronized-Digital Environment Design Abilities in Architectural Design Process", *Journal Of The Faculty Of Engineering And Architecture Of Gazi University*, 24 (4): 583–595 (2009).
3. Akgül, B., "Yapı Elektrik Tesisat Çizim ve Optimizasyon Uygulaması", *Karabük Üniversitesi Fen Bilimleri Enstitüsü*, Karabük, (2012).
4. Cote, M., Rezvanifar, A., and Albu, A. B., "Automatic Generation of Electrical Plan Documents from Architectural Data", *Proceedings Of The ACM Symposium On Document Engineering 2020*, Virtual Event CA USA, 1–4 (2020).
5. Al-Ma'aitah, M., Alwadain, A., and Saad, A., "Application dependable interaction module for computer vision-based human-computer interactions", *Computers & Electrical Engineering*, 97: 107553 (2022).
6. Ripley, B. D., "Pattern Recognition and Neural Networks", 1. Ed., *Cambridge University Press*, (1996).
7. Rusiñol, M. and Lladós, J., "Symbol Spotting in Digital Libraries", *Springer London*, London, (2010).
8. Rezvanifar, A., Cote, M., and Branzan Albu, A., "Symbol spotting for architectural drawings: state-of-the-art and new industry-driven developments", *IPSJ Transactions On Computer Vision And Applications*, 11 (1): 2 (2019).
9. Gavali, P. and Banu, J. S., "Deep Convolutional Neural Network for Image Classification on CUDA Platform", *Deep Learning and Parallel Computing Environment for Bioengineering Systems*, *Elsevier*, 99–122 (2019).
10. Khan, A. I. and Al-Habsi, S., "Machine Learning in Computer Vision", *Procedia Computer Science*, 167: 1444–1451 (2020).
11. Domínguez, B., García, Á. L., and Feito, F. R., "Semiautomatic detection of floor topology from CAD architectural drawings", *Computer-Aided Design*, 44 (5): 367–378 (2012).

12. Gerstweiler, G., Furlan, L., Timofeev, M., and Kaufmann, H., "Extraction of Structural and Semantic Data from 2D Floor Plans for Interactive and Immersive VR Real Estate Exploration", *Technologies*, 6 (4): 101 (2018).
13. Zhu, J., Zhang, H., and Wen, Y., "A New Reconstruction Method for 3D Buildings from 2D Vector Floor Plan", *Computer-Aided Design And Applications*, 11 (6): 704–714 (2014).
14. Yin, X., Wonka, P., and Razdan, A., "Generating 3D Building Models from Architectural Drawings: A Survey", *IEEE Computer Graphics And Applications*, 29 (1): 20–30 (2009).
15. Ahmed, S., Liwicki, M., Weber, M., and Dengel, A., "Improved Automatic Analysis of Architectural Floor Plans", *International Conference on Document Analysis and Recognition (ICDAR)*, Beijing, China, 864–869 (2011).
16. Ah-Soon, C. and Tombre, K., "Architectural symbol recognition using a network of constraints", *Pattern Recognition Letters*, 22 (2): 231–248 (2001).
17. Dosch, P., Tombre, K., Ah-Soon, C., and Masini, G., "A complete system for the analysis of architectural drawings", *International Journal On Document Analysis And Recognition*, 3 (2): 102–116 (2000).
18. Guo, T., Zhang, H., and Wen, Y., "An improved example-driven symbol recognition approach in engineering drawings", *Computers & Graphics*, 36 (7): 835–845 (2012).
19. de las Heras, L.-P., Terrades, O. R., Robles, S., and Sánchez, G., "CVC-FP and SGT: a new database for structural floor plan analysis and its groundtruthing tool", *International Journal On Document Analysis And Recognition (IJDAR)*, 18 (1): 15–30 (2015).
20. Tabbone, S., Wendling, L., and Tombre, K., "Matching of graphical symbols in line-drawing images using angular signature information", *International Journal On Document Analysis And Recognition*, 6 (2): 115–125 (2003).
21. Su Yang, "Symbol recognition via statistical integration of pixel-level constraint histograms: a new descriptor", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 27 (2): 278–281 (2005).
22. Escalera, S., Fornés, A., Pujol, O., Radeva, P., Sánchez, G., and Lladós, J., "Blurred Shape Model for binary and grey-level symbol recognition", *Pattern Recognition Letters*, 30 (15): 1424–1433 (2009).
23. Escalera, S., Fornés, A., Pujol, O., Lladós, J., and Radeva, P., "Circular Blurred Shape Model for Multiclass Symbol Recognition", *IEEE Transactions On Systems, Man, And Cybernetics, Part B (Cybernetics)*, 41 (2): 497–506 (2011).
24. Nguyen, T.-O., Tabbone, S., and Terrades, O. R., "Symbol Descriptor Based on Shape Context and Vector Model of Information Retrieval", *The Eighth IAPR*

- International Workshop on Document Analysis Systems (DAS)*, Nara, Japan, 191–197 (2008).
25. Nguyen, T.-O., Tabbone, S., and Boucher, A., "A Symbol Spotting Approach Based on the Vector Model and a Visual Vocabulary", *10th International Conference on Document Analysis and Recognition*, Barcelona, Spain, 708–712 (2009).
  26. Rusiñol, M., Borràs, A., and Lladós, J., "Relational indexing of vectorial primitives for symbol spotting in line-drawing images", *Pattern Recognition Letters*, 31 (3): 188–201 (2010).
  27. Santosh, K. C., Lamiroy, B., and Wendling, L., "Integrating vocabulary clustering with spatial relations for symbol recognition", *International Journal On Document Analysis And Recognition (IJ DAR)*, 17 (1): 61–78 (2014).
  28. Santosh, K. C., Wendling, L., and Lamiroy, B., "BoR: Bag-of-Relations for Symbol Retrieval", *International Journal Of Pattern Recognition And Artificial Intelligence*, 28 (06): 1450017 (2014).
  29. Ramel, J. Y., Vincent, N., and Emptoz, H., "A structural representation for understanding line-drawing images", *International Journal On Document Analysis And Recognition*, 3 (2): 58–66 (2000).
  30. Broelemann, K., Dutta, A., Jiang, X., and Lladós, J., "Hierarchical Graph Representation for Symbol Spotting in Graphical Document Images", Structural, Syntactic, and Statistical Pattern Recognition, *Springer Berlin Heidelberg*, Berlin, Heidelberg, 529–538 (2012).
  31. Broelemann, K., Dutta, A., Jiang, X., and Lladós, J., "Hierarchical Plausibility-Graphs for Symbol Spotting in Graphical Documents", Graphics Recognition. Current Trends and Challenges, *Springer Berlin Heidelberg*, Berlin, Heidelberg, 25–37 (2014).
  32. Dutta, A., Lladós, J., Bunke, H., and Pal, U., "Near Convex Region Adjacency Graph and Approximate Neighborhood String Matching for Symbol Spotting in Graphical Documents", *12th International Conference on Document Analysis and Recognition (ICDAR)*, Washington, DC, USA, 1078–1082 (2013).
  33. Rusiñol, M. and Lladós, J., "A Region-Based Hashing Approach for Symbol Spotting in Technical Documents", Graphics Recognition. Recent Advances and New Opportunities, *Springer Berlin Heidelberg*, Berlin, Heidelberg, 104–113 (2008).
  34. Bodic, P. L., Locteau, H., Adam, S., Héroux, P., Lecourtier, Y., and Knippel, A., "Symbol Detection Using Region Adjacency Graphs and Integer Linear Programming", *10th International Conference on Document Analysis and Recognition*, Barcelona, Spain, 1320–1324 (2009).
  35. Le Bodic, P., Héroux, P., Adam, S., and Lecourtier, Y., "An integer linear program for substitution-tolerant subgraph isomorphism and its use for symbol

- spotting in technical drawings", *Pattern Recognition*, 45 (12): 4214–4224 (2012).
36. Barducci, A. and Marinai, S., "Object recognition in floor plans by graphs of white connected components", *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 298–301 (2012).
  37. Santosh, K. C., Lamiroy, B., and Wendling, L., "Symbol recognition using spatial relations", *Pattern Recognition Letters*, 33 (3): 331–341 (2012).
  38. Dutta, A., Lladós, J., and Pal, U., "A symbol spotting approach in graphical documents by hashing serialized graphs", *Pattern Recognition*, 46 (3): 752–768 (2013).
  39. Luqman, M. M., Ramel, J.-Y., Lladós, J., and Brouard, T., "Subgraph Spotting through Explicit Graph Embedding: An Application to Content Spotting in Graphic Document Images", *International Conference on Document Analysis and Recognition (ICDAR)*, Beijing, China, 870–874 (2011).
  40. Dey, S., Dutta, A., Lladós, J., Fornes, A., and Pal, U., "Shallow Neural Network Model for Hand-Drawn Symbol Recognition in Multi-Writer Scenario", *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, 31–32 (2017).
  41. Riba, P., Dutta, A., Lladós, J., and Fornes, A., "Graph-Based Deep Learning for Graphics Classification", *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, 29–30 (2017).
  42. Ghosh, S., Shaw, P., Das, N., and Santosh, K. C., "GSD-Net: Compact Network for Pixel-Level Graphical Symbol Detection", *International Conference on Document Analysis and Recognition Workshops (ICDARW)*, Sydney, Australia, 68–73 (2019).
  43. Ziran, Z. and Marinai, S., "Object Detection in Floor Plan Images", *Artificial Neural Networks in Pattern Recognition*, Springer International Publishing, Cham, 383–394 (2018).
  44. Rezvaniyar, A., Cote, M., and Albu, A. B., "Symbol Spotting on Digital Architectural Floor Plans Using a Deep Learning-based Framework", *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA, 2419–2428 (2020).
  45. Delalandre, M., Pridmore, T., Valveny, E., Locteau, H., and Trupin, E., "Building Synthetic Graphical Documents for Performance Evaluation", *Graphics Recognition. Recent Advances and New Opportunities*, Springer Berlin Heidelberg, Berlin, Heidelberg, 288–298 (2008).
  46. Internet: Delalandre, M., "SESYD, Systems Evaluation SYnthetic Documents", <http://mathieu.delalandre.free.fr/projects/sesyd/> (2022).

47. Fan, Z., Zhu, L., Li, H., Chen, X., Zhu, S., and Tan, P., "FloorPlanCAD: A Large-Scale CAD Drawing Dataset for Panoptic Symbol Spotting", *IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 10108–10117 (2021).
48. Valveny, E. and Dosch, P., "Symbol Recognition Contest: A Synthesis", *Graphics Recognition. Recent Advances and Perspectives*, **Springer Berlin Heidelberg**, Berlin, Heidelberg, 368–385 (2004).
49. Valveny, E., Delalandre, M., Raveaux, R., and Lamiroy, B., "Report on the Symbol Recognition and Spotting Contest", *Graphics Recognition. New Trends and Challenges*, **Springer Berlin Heidelberg**, Berlin, Heidelberg, 198–207 (2013).
50. Bi, Z. and Wang, X., "Computer Aided Design and Manufacturing", 1. Ed., **Wiley**, (2020).
51. Akgül, B. and Kutucu, H., "Convex Hull Problemine Genel Bakış Algoritmalar ve Karmaşıklıkları", *Akademik Bilişim Konferansı*, Eskişehir, Türkiye, (2015).
52. Liu, X.-Z., Yong, J.-H., Zheng, G.-Q., and Sun, J.-G., "An offset algorithm for polyline curves", *Computers In Industry*, 58 (3): 240–254 (2007).
53. Hescock, J., Newman, C., and Agioutantis, Z., "Development of a new algorithm for implementing the edge effect offset for subsidence calculations", *International Journal Of Mining Science And Technology*, 28 (1): 61–66 (2018).
54. Wein, R., "Exact and approximate construction of offset polygons", *Computer-Aided Design*, 39 (6): 518–527 (2007).
55. Kaşıkçı, İ., "Elektrik Tesisleri Güvenlik, Koruma ve Uygulama Esasları", **Birsen Yayınevi**, İstanbul, 355 (2018).
56. İlisu, İ. Ö., "Elektrik Tesislerinde Dolaylı Dokunmaya Karşı Koruma ve Topraklama", *Elektrik Mühendisleri Odası Yayını*, 109 (2010).
57. TMMOB Elektrik Mühendisleri Odası, "Elektrik İç Tesisleri Proje Hazırlama Yönetmeliği", *Elektrik Mühendisleri Odası Yayını*, (2010).
58. Fürnkranz, J., "Separate-and-Conquer Rule Learning", *Artificial Intelligence Review*, 13 (1): 3–54 (1999).
59. McCulloch, W. S. and Pitts, W., "A logical calculus of the ideas immanent in nervous activity", *The Bulletin Of Mathematical Biophysics*, 5 (4): 115–133 (1943).
60. Blum, A. L. and Langley, P., "Selection of relevant features and examples in machine learning", *Artificial Intelligence*, 97 (1–2): 245–271 (1997).

61. Kutucu, H. and Almryad, A., "An application of artificial neural networks to assessment of the wind energy potential in Libya", *7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, Hammamet, Tunisia, 405–409 (2016).
62. Arica, N. and Yarman-Vural, F. T., "An overview of character recognition focused on off-line handwriting", *IEEE Transactions On Systems, Man And Cybernetics, Part C (Applications And Reviews)*, 31 (2): 216–233 (2001).
63. Jayabala, P., Srinivasan, E., and Himavathi, S., "An investigation on the performance of hybrid features for feed forward neural network based English handwritten character recognition system", *WSEAS Transactions On Signal Processing*, 10: 21–29 (2014).
64. Das, T. K., Tripathy, A. K., and Mishra, A. K., "Optical character recognition using artificial neural network", *International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, 1–4 (2017).
65. Karouni, A., Daya, B., and Bahlak, S., "Offline signature recognition using neural networks approach", *Procedia Computer Science*, 3: 155–161 (2011).
66. Lakshmi, K. V. and Nayak, S., "Off-line signature verification using Neural Networks", *3rd IEEE International Advanced Computing Conference (IACC 2013)*, Ghaziabad, 1065–1069 (2013).
67. Yegnanarayana, B., "Artificial neural networks for pattern recognition", *Sadhana*, 19 (2): 189–238 (1994).
68. Yalçın, N., "Konuşma Tanıma Teorisi ve Teknikleri", *Kastamonu Eğitim Dergisi*, 16 (1): 249–266 (2008).
69. Shahamiri, S. R. and Binti Salim, S. S., "Real-time frequency-based noise-robust Automatic Speech Recognition using Multi-Nets Artificial Neural Networks: A multi-views multi-learners approach", *Neurocomputing*, 129: 199–207 (2014).
70. Baldi, P. and Chauvin, Y., "Neural Networks for Fingerprint Recognition", *Neural Computation*, 5 (3): 402–418 (1993).
71. Jain, A. and Pankanti, S., "Fingerprint Recognition", *The Essential Guide to Image Processing*, Elsevier, 649–676 (2009).
72. Erdinc Kocer, H. and Kursat Cevik, K., "Artificial neural networks based vehicle license plate recognition", *Procedia Computer Science*, 3: 1033–1037 (2011).
73. Meng Joo Er, Shiqian Wu, Juwei Lu, and Hock Lye Toh, "Face recognition with radial basis function (RBF) neural networks", *IEEE Transactions On Neural Networks*, 13 (3): 697–710 (2002).
74. De Vries, J. and others, "Object recognition: a shape-based approach using artificial neural networks", *Department Of Computer Science, University Of Utrecht*, 34–51 (2006).

75. Barthakur, M., Thakuria, T., and Sarma, K. K., "Artificial Neural Network (ANN) Based Object Recognition Using Multiple Feature Sets", *Soft Computing Techniques in Vision Science*, **Springer Berlin Heidelberg**, Berlin, Heidelberg, 127–135 (2012).
76. LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning", *Nature*, 521 (7553): 436–444 (2015).
77. Batmaz, Z., Yurekli, A., Bilge, A., and Kaleli, C., "A review on deep learning for recommender systems: challenges and remedies", *Artificial Intelligence Review*, 52 (1): 1–37 (2019).
78. Song, H. A. and Lee, S.-Y., "Hierarchical Representation Using NMF", *Neural Information Processing*, **Springer Berlin Heidelberg**, Berlin, Heidelberg, 466–473 (2013).
79. Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition", *Proceedings Of The IEEE*, 86 (11): 2278–2324 (1998).
80. Krizhevsky, A., Sutskever, I., and Hinton, G. E., "ImageNet classification with deep convolutional neural networks", *Communications Of The ACM*, 60 (6): 84–90 (2017).
81. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., and Chen, T., "Recent advances in convolutional neural networks", *Pattern Recognition*, 77: 354–377 (2018).
82. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P., .
83. Diamantis, D. E., Iakovidis, D. K., and Koulaouzidis, A., "Look-behind fully convolutional neural network for computer-aided endoscopy", *Biomedical Signal Processing And Control*, 49: 192–201 (2019).
84. Seo, Y. and Shin, K., "Hierarchical convolutional neural networks for fashion image classification", *Expert Systems With Applications*, 116: 328–339 (2019).
85. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D., "Mastering the game of Go with deep neural networks and tree search", *Nature*, 529 (7587): 484–489 (2016).
86. Fukushima, K., "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position", *Biological Cybernetics*, 36 (4): 193–202 (1980).

87. İnik, Ö. and Ülker, E., "Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri", *Gaziosmanpaşa Journal Of Scientific Research*, 85–104 (2017).
88. Maas, A. L., Hannun, A. Y., and Ng, A. Y., "Rectifier nonlinearities improve neural network acoustic models", *ICML Workshop On Deep Learning For Audio, Speech And Language Processing*, (2013).
89. Xie, S. and Tu, Z., "Holistically-Nested Edge Detection", *International Journal Of Computer Vision*, 125 (1–3): 3–18 (2017).
90. Yang, H. and Patras, I., "Mirror, mirror on the wall, tell me, is the error small?", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 4685–4693 (2015).
91. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., "Dropout: A simple way to prevent neural networks from overfitting", *Journal Of Machine Learning Research*, 15: 1929–1958 (2014).
92. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R., "Improving neural networks by preventing co-adaptation of feature detectors", *Arxiv:1207.0580*, (2012).
93. Zhang, H., Zhang, L., and Jiang, Y., "Overfitting and Underfitting Analysis for Deep Learning Based End-to-end Communication Systems", *11th International Conference on Wireless Communications and Signal Processing (WCSP)*, Xi'an, China, 1–6 (2019).
94. Seyyarer, E., Ayata, F., Uçkan, T., and Karci, A., "Derin Öğrenmede Kullanılan Optimizasyon Algoritmalarının Uygulanması ve Kıyaslanması", *Anatolian Journal Of Computer Sciences*, 5 (2): 90–98 (2020).
95. Kingma, D. P. and Ba, J., "Adam: A Method for Stochastic Optimization", *ArXiv:1412.6980*, (2014).
96. Bock, S., Goppold, J., and Weiß, M., "An improvement of the convergence proof of the ADAM-Optimizer", *ArXiv:1804.10587*, (2018).
97. Singarimbun, R. N., Nababan, E. B., and Sitompul, O. S., "Adaptive Moment Estimation To Minimize Square Error In Backpropagation Algorithm", *International Conference of Computer Science and Information Technology (ICoSNiKOM)*, Medan, Indonesia, 1–7 (2019).
98. Arı, A. and Berberler, M. E., "Yapay Sinir Ağları ile Tahmin ve Sınıflandırma Problemlerinin Çözümü İçin Arayüz Tasarımı", *Acta Infologica*, 1 (2): 55–73 (2017).
99. Internet: Amidi, A. and Amidi, S., "CS 230 - Deep Learning Tips and Tricks Cheatsheet", <https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks> (2022).

100. Stone, M., "Cross-Validatory Choice and Assessment of Statistical Predictions", *Journal Of The Royal Statistical Society: Series B (Methodological)*, 36 (2): 111–133 (1974).
101. Liu, Y., Zhou, Y., Wen, S., and Tang, C., "A Strategy on Selecting Performance Metrics for Classifier Evaluation:", *International Journal Of Mobile Computing And Multimedia Communications*, 6 (4): 20–35 (2014).
102. Stehman, S. V., "Selecting and interpreting measures of thematic classification accuracy", *Remote Sensing Of Environment*, 62 (1): 77–89 (1997).
103. Smith, L. N., "A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay", *Arxiv:1803.09820*, (2018).
104. Kaur, T. and Gandhi, T. K., "Deep convolutional neural networks with transfer learning for automated brain image classification", *Machine Vision And Applications*, 31 (3): 20 (2020).
105. Pan, S. J. and Yang, Q., "A Survey on Transfer Learning", *IEEE Transactions On Knowledge And Data Engineering*, 22 (10): 1345–1359 (2010).
106. Bird, J. J., Kobylarz, J., Faria, D. R., Ekart, A., and Ribeiro, E. P., "Cross-Domain MLP and CNN Transfer Learning for Biological Signal Processing: EEG and EMG", *IEEE Access*, 8: 54789–54801 (2020).
107. Internet: Stanford Vision Lab, "ILSVRC, ImageNet Large Scale Visual Recognition Challenge", <https://image-net.org/challenges/LSVRC/index.php> (2022).
108. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z., "Rethinking the Inception Architecture for Computer Vision", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2818–2826 (2016).
109. He, K., Zhang, X., Ren, S., and Sun, J., "Identity Mappings in Deep Residual Networks", *Computer Vision – ECCV 2016*, Springer International Publishing, Cham, 630–645 (2016).
110. He, K., Zhang, X., Ren, S., and Sun, J., "Deep Residual Learning for Image Recognition", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 770–778 (2016).
111. Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A., "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning", *Arxiv:1602.07261*, (2016).
112. Xie, S., Girshick, R., Dollar, P., Tu, Z., and He, K., "Aggregated Residual Transformations for Deep Neural Networks", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 5987–5995 (2017).

113. Chollet, F., "Xception: Deep Learning with Depthwise Separable Convolutions", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 1800–1807 (2017).
114. Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q., "Densely Connected Convolutional Networks", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2261–2269 (2017).
115. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", *ArXiv:1704.04861*, (2017).
116. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C., "MobileNetV2: Inverted Residuals and Linear Bottlenecks", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, 4510–4520 (2018).
117. Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., and Adam, H., "Searching for MobileNetV3", *ArXiv:1905.02244*, (2019).
118. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q. V., "MnasNet: Platform-Aware Neural Architecture Search for Mobile", *Arxiv:1807.11626*, (2018).
119. Simonyan, K. and Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition", *Arxiv:1409.1556*, (2014).
120. Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., "Going deeper with convolutions", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 1–9 (2015).
121. Internet: Google Cloud TPU, "Advanced Guide to Inception v3 on Cloud TPU | Google Cloud", <https://cloud.google.com/tpu/docs/inception-v3-advanced> (2022).
122. Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P., "Microsoft COCO: Common Objects in Context", *Arxiv:1405.0312*, (2014).
123. Uijlings, J. R. R., van de Sande, K. E. A., Gevers, T., and Smeulders, A. W. M., "Selective Search for Object Recognition", *International Journal Of Computer Vision*, 104 (2): 154–171 (2013).
124. Girshick, R., Donahue, J., Darrell, T., and Malik, J., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, 580–587 (2014).

125. Girshick, R., "Fast R-CNN", *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 1440–1448 (2015).
126. Ren, S., He, K., Girshick, R., and Sun, J., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 39 (6): 1137–1149 (2017).
127. He, K., Gkioxari, G., Dollár, P., and Girshick, R., "Mask R-CNN", *Arxiv:1703.06870*, (2017).
128. Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You Only Look Once: Unified, Real-Time Object Detection", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 779–788 (2016).
129. Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M., "YOLOv4: Optimal Speed and Accuracy of Object Detection", *Arxiv:2004.10934*, (2020).
130. Redmon, J. and Farhadi, A., "YOLO9000: Better, Faster, Stronger", *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*, 7263–7271 (2017).
131. Redmon, J. and Farhadi, A., "YOLOv3: An Incremental Improvement", *Arxiv:1804.02767*, (2018).
132. Internet: Autodesk Inc, "DXF Format", [http://www.autodesk.com/techpubs/autocad/acad2000/dxf/dxf\\_format.htm](http://www.autodesk.com/techpubs/autocad/acad2000/dxf/dxf_format.htm) (2022).
133. Zhao, Z.-Q., Zheng, P., Xu, S.-T., and Wu, X., "Object Detection With Deep Learning: A Review", *IEEE Transactions On Neural Networks And Learning Systems*, 30 (11): 3212–3232 (2019).
134. Internet: Dial, "DIALux", <https://www.dialux.com> (2022).
135. Internet: RELUX Informatik AG, "ReluxNet", <https://reluxnet.relux.com> (2022).
136. Internet: Dial, "Description of the EULUMDAT Format", <https://evo.support-en.dial.de/support/solutions/articles/9000074164-description-of-the-eulumdat-format> (2022).
137. Marx, P., "New goniophotometers for lighting engineering laboratories", *Light & Engineering*, 5 (4): 32–36 (1997).
138. Şahin, M., Oğuz, Y., and Büyüktümtürk, F., "İç Mekân Aydınlatmasında Renk Seçiminin Aydınlatma Ekonomisi ve Görseelliğe Etkisi", *Makine Teknolojileri Elektronik Dergisi*, 2013 (10): 15–26 (2013).
139. Dincer, K., Köse, H., Dede, O., Serdar, P., and Tosun, M., "Duvar Renginin Aydınlatmada Güç Performansına Etkisinin İncelenmesi ve Örnek Bir Uygulama", *Journal Of Selcuk-Technic*, 12 (2): 25–38 (2013).

140. Internet: TMMOB Elektrik Mühendisleri Odası, "Aydınlatma-En Az Aydınlık Düzeyleri Tablosu", [http://bbm.emo.org.tr/genel/katalog\\_detay.php?katalog=11&kayit=13](http://bbm.emo.org.tr/genel/katalog_detay.php?katalog=11&kayit=13) (2022).
141. Beineke, L. W. and Wilson, R. J., "Introduction to Graph Theory.", *The American Mathematical Monthly*, 81 (6): 679 (1974).
142. Weiss, M. A., "Data Structures and Algorithm Analysis in C++", Fourth. Ed., *Addison-Wesley*, New Jersey, 635 (2014).
143. Dijkstra, E. W., "A note on two problems in connexion with graphs", *Numerische Mathematik*, 1 (1): 269–271 (1959).
144. Kruskal, J. B., "On the shortest spanning subtree of a graph and the traveling salesman problem", *Proceedings Of The American Mathematical Society*, 7 (1): 48–50 (1956).
145. Prim, R. C., "Shortest Connection Networks And Some Generalizations", *Bell System Technical Journal*, 36 (6): 1389–1401 (1957).
146. Gilbert, E. N. and Pollak, H. O., "Steiner Minimal Trees", *SIAM Journal On Applied Mathematics*, 16 (1): 1–29 (1968).
147. Hwang, F. K., "On Steiner Minimal Trees with Rectilinear Distance", *SIAM Journal On Applied Mathematics*, 30 (1): 104–114 (1976).
148. Chen, H., Qiao, C., Zhou, F., and Cheng, C.-K., "Refined single trunk tree: a rectilinear steiner tree generator for interconnect prediction", *Proceedings Of The 2002 International Workshop On System-Level Interconnect Prediction - SLIP '02*, San Diego, California, USA, 85 (2002).

## ÖZGEÇMİŞ

Bayram AKGÜL ilk ve orta öğrenimini Şanlıurfa’da tamamladı. 2005 yılında Anadolu Üniversitesi Mühendislik Mimarlık Fakültesi (şimdiki Eskişehir Teknik Üniversitesi Mühendislik Fakültesi) Bilgisayar Mühendisliği Bölümünden mezun oldu. 2009 yılına kadar Bartın Dinç Mühendislik A.Ş.’de yazılım mühendisi olarak çalıştı. 2012 yılında Karabük Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda yüksek lisans programını tamamladı. 2009 yılından beri Bartın Üniversitesi Bartın Meslek Yüksekokulu Bilgisayar Teknolojileri Bölümünde Öğretim Görevlisi olarak çalışmaktadır.