

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



**YOL GÖRÜNTÜSÜ VE HAREKET DUYARGA VERİLERİNİ
KULLANARAK OTOMATİK YOL ANORMALLİĞİ TESPİTİ**

Erkan DEVECİ

Yüksek Lisans Tezi

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Donanım

HAZİRAN 2022

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

YOL GÖRÜNTÜSÜ VE HAREKET DUYARGA VERİLERİNİ KULLANARAK OTOMATİK YOL ANORMALLİĞİ TESPİTİ

Tez Yazarı
Erkan DEVECİ

Danışman
Prof. Dr. Burhan ERGEN

HAZİRAN 2022
ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Başlığı: Yol Görüntüsü ve Hareket Duyarga Verilerini Kullanarak Otomatik Yol
Anormalliği Tespiti
Yazarı: Erkan Deveci
İlk Teslim Tarihi: 13/05/2022
Savunma Tarihi: 21/06/2022

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

Danışman:	Prof. Dr. Burhan ERGEN Fırat Üniversitesi, Mühendislik Fakültesi	<i>İmza</i>	Onayladım
Başkan:	Dr. Öğr. Üyesi Vedat TÜMEN Bitlis Eren Üniversitesi, Mimarlık Fakültesi		Onayladım
Üye:	Doç. Dr. Erhan AKBAL Fırat Üniversitesi, Teknoloji Fakültesi		Onayladım

Bu tez, Enstitü Yönetim Kurulunun/...../2022 tarihli toplantısında tescillenmiştir.

İmza

Prof. Dr. Kürşat Esat ALYAMAÇ
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım “Yol Görüntüsü ve Hareket Duyarga Verilerini Kullanarak Otomatik Yol Anormallığı Tespiti” başlıklı Yüksek Lisans tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

21/06/2022

Erkan DEVECİ



ÖNSÖZ

Günümüzde Derin Öğrenmenin görüntü işleme ve bilgisayarlı görü alanlarında uygulanması ile başarılar elde edilmiştir. Kentleşmenin şehirlerin hızlı gelişmesine neden olması gelişen şehirlerin alt yapılarının takibi ve güncellenme gerekliliğinin analizini zorunlu kılmıştır. Özellikle ulaşım karmaşasının önüne geçmek için çeşitli trafik planlamaları yapmanın yanı sıra ulaşımın sağlandığı kara yolunun niteliğinin de yeterli seviyede olması gereklidir. Bu çalışmada Evrişimsel Sinir Ağları, Makine Öğrenmesi algoritmaları ve Gabor filtresi kullanılarak yol yüzeyindeki anormalliklerin sınıflandırılması, tespiti yapılmıştır. Sensör bilgileri ile çatlağın şiddeti gösterilmiştir.

Bu çalışmada değerli vaktini bana ayırarak ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım danışman hocam Sayın Prof. Dr. Burhan ERGEN'e teşekkür ederim.

Bu tez çalışması, Fırat Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimi (FÜBAP) tarafından MF.21.32 nolu proje ile desteklenmiştir.

Erkan DEVECİ

ELAZIĞ, 2022

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	iv
ÖZET	vii
ABSTRACT	viii
ŞEKİLLER LİSTESİ	ix
TABLolar LİSTESİ	xi
SİMGELER VE KISALTMALAR	xii
1. GİRİŞ.....	1
1.1. Tezin Amacı ve Kapsamı	2
1.2. Tezin Katkıları.....	3
1.3. Tezin Yapısı	3
2. İLGİLİ ÇALIŞMALAR	5
3. YAPAY ZEKÂ ÖĞRENME MODELLERİ VE YÖNTEMLERİ	8
3.1. Evrişimsel Sinir Ağı.....	8
3.2. Makine Öğrenmesi	11
3.1.1. Denetimli Makine Öğrenmesi.....	13
3.1.2. Denetimsiz Makine Öğrenmesi	16
3.3. Derin Öğrenme	18
3.4. Derin Öğrenme Modelleri	19
3.4.1. Yolo Mimarisi	19
3.4.2. SSD	20
3.4.3. Retina-Net.....	21
3.4.4. R-CNN.....	21
3.4.5. Fast R-CNN.....	21
3.4.6. Faster R-CNN	21
3.4.7. SPPNet.....	22
3.4.8. LeNet-5.....	22
4. MATERYAL VE METOT	23
4.1. Çatlak Tespiti	23
4.2. Veri Setleri	24
4.2.1. Beton Çatlak Görüntüleri.....	24
4.2.2. Basit Veri Kümesi	24
4.2.3. Gerçekçi Veri Kümesi	25
4.3. Araçlar.....	26
4.3.1. Tensorflow 2.0 Kütüphanesi	27
4.3.2. Keras Kütüphanesi	27
4.3.3. OpenCV Kütüphanesi.....	27
4.3.4. 6 Eksenli ivme sensörü	27
4.4. VGG-16 Mimarisi	28
4.5. Yönlü Gabor Filtresi.....	29

5. BULGULAR VE TARTIŞMA	32
5.1. Çatlak Sınıflandırmada VGG-16 mimarisinin Kullanımı.....	32
5.1.1. Bir Bloklı VGG-16 Modeli	32
5.1.2. İki Bloklı VGG-16 Modeli	33
5.1.3. Üç Bloklı VGG-16 Modeli	34
5.2. ESA ve Gabor Filtre Kullanarak Çatlak Tespiti	36
5.2.1. ESA Modeli İle Çatlak Tespiti	36
5.2.2. Gabor Filtreli ESA Modeli İle Çatlak Tespiti.....	38
5.3. Gabor Filtresi'nde Değişen Sigma ve Teta Değerlerinin Çatlak Tespitine Etkisi	40
5.3.1. Gabor Filtre Yönelim Açısı (Teta)	41
5.3.2. Gabor Filtre Standart Sapma (Sigma).....	41
5.4. 6 Eksenli İvme Sensörü	46
6. SONUÇLAR	49
KAYNAKLAR.....	51
ÖZGEÇMİŞ	

ÖZET

Yol Görüntüsü ve Hareket Duyarga Verilerini Kullanarak Otomatik Yol Anormalliği Tespiti

Erkan DEVECİ

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı
Haziran 2022, Sayfalar: xii +55

Köprüler, barajlar, betonarme yapılar ve karayolu gibi farklı yapılarda oldukça önemli olan çatlak tespiti görüntü işleme ve sınıflandırma ile tespiti yapılabilir. Karayollarında oluşan çatlakların tespit edilmesi trafik yoğunluğunun azaltılmasını, kazaların önlenmesini, yol denetçilerinin iş yükünün hafiflemesini ve yol bakımının daha iyi yapılmasını sağlar. Asfaltlarda bulunan çatlakların ve kusurların manuel olarak incelenmesi, yorgunluk, sorumsuz denetim, zayıf göz görme gibi bir dizi nedenden dolayı insan hata olasılığı yüksektir. Çatlakların tespiti ve analizi için son yıllarda görüntü işleme ve makine öğrenmesi teknikleri kullanılmıştır. Bu teknikler belirleme ve sınıflandırma ile otomatik olarak gerçekleştirilmesine yoğunlaşmıştır. Karayolu gerçek koşullarının sürekli haritalanması, tutarlı bir kaynak tahsisi ile birlikte yeterli altyapının bakım ve yönetim işlemlerine de izin verir. Bu nedenle, yol kullanıcılarını altyapı kalitesi konusunda bilgilendirmek ve aynı kullanıcılardan bilgi almak, güvenli bir sürüş için, navigasyon için mobil cihaz uygulamalarının (güvenlik yolu için özel bir sosyal ağ gibi) yeni sınırı haline gelmiştir.

Bu çalışmada yol yüzey kalitesinin izlenmesi ve tam yol sahne envanter görüntüleri için kullanışlı bir otomatik algılama sistemini gerçekleştirilmiştir. Temel amaç karayolu ve beton yapılarda bulunan çatlakları tespit etmenin optimal yolunu bulmaktır. Yapılan uygulamalar VGG-16 mimarisi, Evrişimsel Sinir Ağı ve Gabor Filtresi gibi yapılar kullanılarak görüntülerden çatlak tespitine odaklanmaktadır.

Anahtar Kelimeler: Derin Öğrenme, Yol Yüzey Anormalliği, Hareket Duyargası, Gabor Filtresi, VGG-16, Evrişimsel Sinir Ağı

ABSTRACT

Automatic Road Anomaly Detection Using Road Image and Motion Sensor Data

Erkan DEVECİ

Master's Thesis

FIRAT UNIVERSITY

Institute of Science Graduate School of Natural and Applied Sciences

Department of Computer Engineering

June 2022, Pages: xii +55

Crack detection, which is very important in different structures such as bridges, dams, reinforced concrete structures and highways, can be detected by image processing and classification. Detection of cracks on highways reduces traffic density, prevents accidents, eases the workload of road inspectors and provides better road maintenance. The possibility of human error is high due to a number of reasons such as manual inspection of cracks and defects found in asphalt, fatigue, irresponsible inspection, poor eye vision. Image processing and Machine Learning techniques have been used in recent years for the detection and analysis of cracks. These techniques are focused on identification and classification and automatic realization. Continuous mapping of road real conditions allows for the maintenance and management of adequate infrastructure along with a consistent resource allocation. Therefore, informing road users about the quality of infrastructure and getting information from the same users has become the new frontier of mobile device apps for navigation (like a private social network for safety road) for safe driving.

In this study, a useful automatic detection system has been implemented for monitoring road surface quality and full road scene inventory images. The main goal is to find the optimal way to detect cracks found in roads and concrete structures. The applications focus on crack detection from images using structures such as VGG16 architecture, Convolutional Neural Network and Gabor Filter.

Keywords: Deep Learning, Highway Surface Anomaly, Motion Sensor, Gabor Filter, VGG-16, Convolutional Neural Network

ŞEKİLLER LİSTESİ

Sayfa

Şekil 1.1. Gerçek otoyol çatlak görüntüleri	1
Şekil 2.1. Kusur yakalama, sınıflandırma ve değerlendirme için bilgisayarla görme yöntemleri sınıflandırması[6].....	5
Şekil 3.1. Bir ESA'nın genel mimarisi	9
Şekil 3.2. a) ESA'nın orijinal yapısı b) DropOut katmanından sonraki hali.....	10
Şekil 3.3. ÇD'de veri bölümlleme	12
Şekil 3.4. Modelin eğitim ve doğrulama hatası ölçümünde varyans ve standart sapmanın etkisi	13
Şekil 3.5. Denetimli MÖ tasarımı.....	14
Şekil 3.6. Sigmoid fonksiyonunun grafiği. $R = 0,5$	15
Şekil 3.7. Denetimsiz MÖ tasarımı	17
Şekil 3.8. DÖ ve MÖ farkının gösterimi	19
Şekil 3.9. YOLO Mimarisinin Genel Yapısı [57].....	20
Şekil 3.10. LeNet-5 Mimarisinin Genel Yapısı [82]	22
Şekil 4.1. Çatlak Tespiti'nde ESA Mimarisi	23
Şekil 4.2. Basit test görüntülerinin alt kümesi, çatlaklar	24
Şekil 4.3. Gerçekçi veri kümeleri	25
Şekil 4.4. İvme Sensörünün yapısı	28
Şekil 4.5. VGG-16 Mimari Haritası [73].....	29
Şekil 4.6. a) Gabor çekirdeğinin Theta 0° 'daki durumu b) Gabor çekirdeğinin Theta 45° 'deki durumu c) Gabor çekirdeğinin Theta 90° 'daki durumu	31
Şekil 5.1. Çatlaklar ve Normaller Veri Seti'nde Bir Bloklü VGG-16 Modeli için Kayıp ve Doğruluk Öğrenme Modellerinin Çizgi Grafikleri.....	33
Şekil 5.2. Çatlaklar ve Normaller Veri Seti'nde İki Bloklü VGG-16 Modeli için Kayıp ve Doğruluk Öğrenme Modellerinin Çizgi Grafikleri	34
Şekil 5.3. Çatlaklar ve Normaller Veri Seti'nde Üç Bloklü VGG-16 Modeli için Kayıp ve Doğruluk Öğrenme Modellerinin Çizgi Grafikleri	35
Şekil 5.4. Sistemin temel adımları.....	36
Şekil 5.5. a), b) Yüzey çatlak görüntüleri c), d) Yüzey normal görüntüleri.....	37
Şekil 5.6. Çatlaklar ve normaller veri setinde ESA Modeli için doğruluk grafiği.....	37
Şekil 5.7. Çatlaklar ve normaller veri setinde ESA Modeli için hata grafiği	38
Şekil 5.8. a), b) Yüzey çatlak görüntülerinin Gabor Filtre çıktısı c), d) Yüzey normal görüntülerinin Gabor Filtre çıktısı.....	38
Şekil 5.9. Gabor Filtre uygulanmış veri setinin doğruluk grafiği	39

Şekil 5.10. Gabor Filtre uygulanmış veri setinin hata grafiği.....	39
Şekil 5.11. a) Normal çatlak görüntüsü, b) Teta parametresi 0° iken oluşturulan veri seti c) Teta parametresi 30° iken oluşturulan veri seti d) Teta parametresi 60° iken oluşturulan veri seti e) Teta parametresi 90° iken oluşturulan veri seti.....	41
Şekil 5.12. a) Çatlak görüntüsünün histogram grafiği, b) Teta parametresi 0 iken oluşan çatlak resminin histogram grafiği, c) Teta parametresi 30 iken oluşan çatlak resminin histogram grafiği, d) Teta parametresi 60 iken oluşan çatlak resminin histogram grafiği, e) Teta parametresi 90 iken oluşan çatlak resminin histogram grafiği.....	42
Şekil 5.13. En iyi başarı gösteren modelin doğruluk grafiği	45
Şekil 5.14. En düşük hata payına sahip modelin grafiği.....	46
Şekil 5.15. İvme Sensörü'nden gelen değerlerin anlık grafiği.....	47
Şekil 5.16. Yol çatlak görüntüsü	47



TABLÖLAR LİSTESİ

	Sayfa
Tablo 1.1. Çatlak tespiti etmek için kullanılan metotlar	7
Tablo 3.1. Aktivasyon fonksiyonlarının matematiksel gösterimi	16
Tablo 4.1. Veri Setlerinin Özellikleri.....	26
Tablo 5.1. 5 kat veri kümesinde farklı eğitilmiş modeller kullanılarak ortalama sınıflandırma başarıları	35
Tablo 5.2. Yalın ESA ve Gabor Filtreli Veri Setlerinin Başarı Ölçütleri Tablosu.....	40
Tablo 5.3. Karmaşıklık matrisi.....	43
Tablo 5.4. Başarı ölçütlerinin Teta°'na göre değişimi	43
Tablo 5.5. Başarı ölçütlerinin Sigma'ya göre değişimi.....	44
Tablo 5.6. İlk 20 adım için İvme Sensör değerleri.....	48

SİMGELER VE KISALTMALAR

Kısaltmalar

CNN	: Convolutional Neural Networks
ÇEK	: Çapraz Entropi Kaybı
DN	: Doğru Negatif
ÇD	: Çapraz Doğrulama
DÖ	: Derin Öğrenme
DP	: Doğru Pozitif
TBA	: Temel Bileşen Analizi
BDA	: Bağımsız Bileşen Analizi
DVM	: Destek Vektör Makinesi
ESA	: Evrişimsel Sinir Ağı
GYA	: Geri Yayılım Algoritması
MÖ	: Makine Öğrenmesi
ReLU	: Düzeltilmiş Doğrusal Birim
SSD	: Single-Shot-Multi-box Detector
YOLO	: You Only Look Once
TF	: Tensorflow (Python kütüphanesi)
YN	: Yanlış Negatif
YP	: Yanlış Pozitif
YSA	: Yapay Sinir Ağı

1. GİRİŞ

Bu tez çalışmasında her ne kadar çevrimdışı bir veri seti üzerinde uygulama geliştirilse de geliştirilen yöntemleri gerçek zamanlı olarak uygulamak mümkündür. Ancak, dinamik tabanlı sistemler ve bu sistemlerin daha karmaşık bir yapısı, gelecek çalışma konusu olduğu için bu tez çalışmasında değinilmemiştir.

Çatlaklar, asfalt veya beton kaplama meydana gelen anormali türlerinden biridir. Kaplamanın hizmet ömrünü kısaltır ve güvenli sürüşü tehlikeye atar. Yol bakım maliyetini optimize etmek için yol sıkıntısının erken tespit edilmesi gerekir. Yol sıkıntısının otomatik olarak incelenmesi, kaplama dokularının önemli varyasyonlarından kaynaklanan küçük çatlakların tespiti için büyük bir zorluktur.

Kusurların boyutları, asfalt veya beton deseninin homojenliğini bozan görüntülenen yüzeye göre genellikle küçük yapıya sahiptir. İnsan denetçiler oldukça esnektir ve dokulu yüzeylerdeki yeni kusurları kolayca tanıyabilir. Bu öngörülemeyen kusurların çoğu, açık önlemlerle tanımlanamaz, bu da otomatik kusur tespitini zorlaştırır. Yapay algılamaya dayalı geleneksel tespit yöntemleri, yoğun emek gerektirir, zaman alıcıdır ve beton kaplama çatlaklarını doğru bir şekilde tespit edemez. Görüntü analizi teknikleri, endüstriyel denetimi otomatikleştirmek için giderek daha fazla kullanılmaktadır. Manüel teftiş faaliyeti sübjektif olabilir ve büyük ölçüde insan müfettişlerin deneyimine bağlı olabilir. Bu nedenle çatlakların erken bulunması ve onarılması gerekmektedir. Manuel olarak fotoğraflanan yol yüzeyinde meydana gelen birkaç kusur çeşidi Şekil 1.1’de görülmektedir.



Şekil 1.1. Gerçek otoyol çatlak görüntüleri

Dijital görüntü işleme teknolojisinin gelişmesiyle birlikte, birçok araştırmacı, asfalt veya beton kaplama çatlak tespiti için çeşitli tespit yöntemleri önermiştir. Asfalt yapıların yüzeylerinin çatlak genişliğini nicel olarak değerlendirmek için stereo görüş tabanlı bir çatlak genişliği algılama yöntemi önerildi [1]. Gürültüyü gidermek için bir genişlik haritası kullanan kenar tabanlı bir çatlak algılama yöntemi önerdi ve artık gürültü, çatlak bölgelerinin yeniden sınıflandırılmasıyla giderildi [2]. Otomatik yüzey incelemesinde, homojen bir dokuya gömülü yerel anomaliler olarak görünen küçük yüzey kusurlarını tespit etme problemini çözmek gerekir.

Dokular genellikle yapısal ve istatistiksel olmak üzere iki ana tipte sınıflandırılır [3]. Yapısal dokular, deterministik bir yer değiştirme kuralı ile bazı temel doku ilkellerinin tekrarlarından oluşan dokulardır. Bu tip kusur incelemesi tekstil kumaşlarında ve işlenmiş yüzeylerde ortaya çıkar. İstatistiksel dokular, ilkel ve yer değiştirme kuralları ile tanımlanamaz. Bu tür dokulu görüntüdeki gri seviyelerin uzamsal dağılımı oldukça stokastiktir. Zımpara kağıdı ve dökümlerin dokulu yüzeyleri bu kategoriye girer. Bunlar gibi yapılan bir çok çalışmanın detaylı anlatımı ilgili çalışmalar kısmında açıklanmaktadır.

Bu çalışma, Derin Öğrenme (DÖ) tekniğini benimsemektedir. Çatlak olan bölgeyi tanıma ve hasarsız olan yüzeyden ayırt etme problemini çözmek için Evrimsel Sinir Ağı (ESA)'nın etkili mimarilerini ve ikili sınıflandırma işlevlerini kullanmaktadır. Bütün bu araştırmalar sonucunda bu tez çalışması aracın karayolundan geçişi esnasında meydana gelen sarsıntıyı araçtaki sensörler yardımıyla tespit etmeye ve aynı zamanda bu kusurun grafiğini çizdirmeyi ve hasarın boyutunu belirlemeyi amaçlar. Ayrıca var olan veri setleri üzerinden oluşturulan modeller ile sınıflandırma yapmaktadır. Veri setlerini ön işlem adımlarından geçirdikten sonra yönlü Gabor Filtresi'ni kullanarak ve değişen parametrelerle optimizasyonunu sağlayarak, var olan sistemlerle karşılaştırılmasını ve en iyi başarı ölçütlerinin elde edilmesini amaçlamıştır.

1.1. Tezin Amacı ve Kapsamı

Sunulan tezin amacı, karayolu çatlaklarının tespitine yönelik olarak daha başarılı yöntemler geliştirmek ve literatüre katkıda bulunmaktır.

Yol çatlak tespiti, karayolu tehlike değerlendirmesi alanında önemli bir rol oynar. Geleneksel çatlak tespiti yöntemleri zaman alıcı, zahmetli ve insan muhakemesine dayanır. Bu nedenle, yoldaki çatlakları hızlı ve doğru bir şekilde tespit etmek ve tanımlamak için otomatik çatlak tespiti çok önemlidir. Yol üzerinde bulunan çatlaklardan, araçların geçişi esnasında her türlü titreşim ve sarsıntı hassas ivmeölçer tarafından kaydedilmektedir. Kaydedilen veriler veri tabanında depolanmaktadır. Bu verilerle ilgili grafikler ve çatlak analizi yapılmaktadır. Çatlak veri setleri ile yapılan sınıflandırmalar ile çatlak tespiti ve ivme sensöründen gelen veriler ile de sarsıntı tespiti ve şiddete göre grafikleme işlemi yapılmaktadır. Sensör verileri ve veri setlerindeki

görüntülerden yapılan analizler ve sınıflandırmalar ile yol yüzeyindeki çatlakların büyük ölçüde tespiti amaçlanmıştır.

1.2. Tezin Katkıları

Bu tez çalışmasının, literatüre katkıları şu şekilde özetlenebilir.

- İkili sınıflandırma kullanarak karayolu çatlak tespiti yapma.
- Yol envanter görüntülerinden yola çıkarak çatlak resim ile çatlak olmayan resimler arasındaki farkların neler olduğunu araştırmak ve şimdiye kadar yapılan çeşitli uygulamaları inceleme.
- Yönlü Gabor Filtresi kullanarak çatlak tespiti için yeni bir yöntem geliştirme.
- VGG-16 mimarisi kullanarak yol ve yüzey görüntülerini sınıflandırma
- Gabor Filtresi'nin değişen parametrelerini kullanarak oluşturulan veri setlerinden optimum başarı elde etme.
- Yüzey görüntülerinde hasarlı bölgeyi ayırt etmek ve sınıflandırmak için birçok inceleme ve araştırma yapma.

Bu tez çalışması kapsamında bir adet konferans bildirisi sunulmuş, bir adet Bilimsel Araştırma Projeleri (BAP) gerçekleştirilmiş ve bir adet TR-DİZİN indeksli makale yayınlanmıştır. İlgili çalışmalar hakkındaki detaylar özgeçmiş bölümünde yer almaktadır.

1.3. Tezin Yapısı

Karayolu ya da beton yüzeyde bulunan anormaliyi tespit etmek için yol görüntüleri üzerinde çatlak bulunan resimler çatlak ve çatlak bulunmayan resimler ise normal olarak etiketlendi. Yeni bir çatlak algılama yöntemi geliştirmek için çevrimdışı veriler daha tutarlı sonuçlar vermektedir. Bu doğrultuda, görüntü üzerinden hasarlı olan bölgenin çıkarılması ile ilgili olarak geçmiş çalışmalardan hareketle VGG-16 mimarisi kullanılmıştır. Sinir ağı yardımıyla bu özelliklerden çatlağı tespit eden ve tahmin eden bir uygulama elde edilmiştir. Çatlak sınıflandırılmasını daha iyi biçimde gerçekleştirmek adına yönlü Gabor çekirdeğinden yararlanılmıştır. Bu filtrenin değişen parametrelerine göre veri setleri oluşturulmuş ve sınıflandırmadaki başarısı gözlemlenmiştir. Bu tez çalışmasının geri kalanı aşağıdaki gibi düzenlenmiştir.

Bölüm 2'de Çatlak tespiti kısaca açıklanmış ve literatürde yapılan bazı çalışmalara yer verilmiştir.

Bölüm 3’de, Yapay Sinir Ağı (YSA), Makine Öğrenmesi (MÖ), DÖ ve ESA açıklanmış ve çatlak tespitinde bu yapı ve yöntemlerin nasıl uygulandığı anlatılmış. Ayrıca kullanılan diğer DÖ modellerinden bahsedilmiştir.

Bölüm 4’de, yol anormalisi algılanırken kullanılan yöntemler, araçlar tanıtılmış. Veri setine uygulanan filtrelerden bahsedilmiş ve mimariler açıklanmıştır.

Bölüm 5’te, tasarlanan sistem yapısını ve değerlendirilmesini, görüntü sınıflandırması için hangi tür ağların uygun olduğunu açıklanmıştır. Kullanılan mimarilerin doğruluk ve hata grafikleri, karmaşıklık matrisleri ve elde edilen sonuçların başarı ölçütleri tablolarına yer verilmiştir.

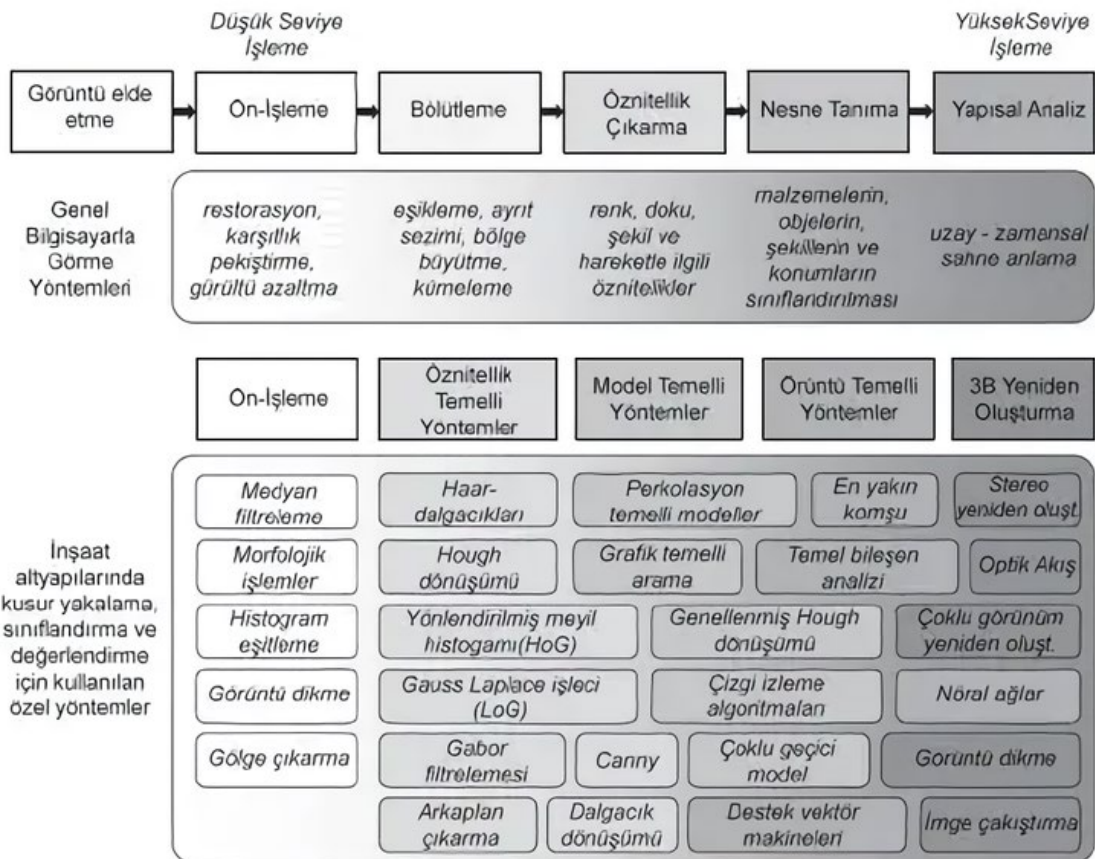
Bölüm 6 sonuç kısmıdır. Tez hakkında yapılan çalışmalar değerlendirilerek gelecek çalışmalar için öneriler verilmiştir.



2. İLGİLİ ÇALIŞMALAR

Çatlak tespiti için görüntü işlemeden köken alan teknikler, esas olarak görüntü ön işlemeyi, bölütlemeyi, görüntüden özniteliklerin çıkarılmasını ve çatlak tanımayı içerir. Görüntü işleme teknikleri, kenar algılama (piksellerin piksel yoğunluğundaki ani değişikliği tarama) ve segmentasyon (nesneyi tanımlama) kullanır. Çatlak tespit problemini çözmek için yakalanan görüntüler aracılığıyla çatlağı görsel verilerden tespit etmek, sınıflandırmak ve segmentlere ayırmak için kullanılan en eski tekniklerden biridir.

Literatürde birçok çalışma derlemesi bulunmakla beraber, bu derlemeler genellikle ya söz konusu çalışmayı besleyen literatürün derlemesi ya da iş akışına bağlı olarak yöntemlerin karmaşıklığına göre sınıflandırılması ya da tarihsel gelişimi ile sınırlandırılmıştır. Bu tip derlemelere örnek olarak [4, 5] gösterilebilir. Yayınlanan çalışmada genel bilgisayarla görme yöntemleri ve asfalt ve beton altyapılarda kusur yakalama, sınıflandırma ve değerlendirme için kullanılan özel yöntemleri, yöntemlerin karmaşıklığına göre sınıflandırmıştır [6].



Şekil 2.1. Kusur yakalama, sınıflandırma ve değerlendirme için bilgisayarla görme yöntemleri sınıflandırması[6]

Kusur yakalama, sınıflandırma ve değerlendirme için özel yöntemler ön-işleme, öznel temelli yöntemler, model temelli yöntemler, örüntü temelli yöntemler ve 3 boyutlu yeniden oluşturma olarak Şekil 2.1’de görüldüğü gibi 5 kategoride sınıflandırmıştır. Bu kategoriler iş akışını doğrudan tanımlamıyor olsa da tamamen ayrı oldukları da söylenemez. Bunun yerine iki veya daha fazla kategori birleştirerek ve birbirini izleyen şekillerde uygulanmaktadır [6]. Aynı şekilde kategori içerisinde de birden fazla yöntem kullanmak mümkündür. Örnek vermek gerekirse hem medyan filtresi hem de histogram eşitleme uygulanarak ön-işlemeden geçirilen bir görüntü öznel temelli yöntemlerde kullanılabilir. Dolayısıyla bilgisayarla görme ile çatlak tespiti, sınıflandırılması ve değerlendirilmesi işi için birçok kombinasyonla farklı algoritmalar yaratmak mümkündür.

İnsanların görüntü sınıflandırmasını nasıl tanımladıkları konusunda net bir sınır yoktur. Ancak, görüntü sınıflandırma ve nesne tanıma terimlerini birbirinin yerine kullanıyoruz ve sorunu, bir görüntüyü ait olduğu bir sınıf veya nesne kategorisine göre etiketleme sorunu olarak tanımlıyoruz. Birçok uygulama ve görev, başarılı bir görüntü sınıflandırması gerektirdiğinden, görev büyük önem taşımaktadır. Örneğin, amacın görüntü uzayında belirli bir yerde hangi nesnenin bulunduğunu bulmak olduğu bir nesne algılama görevinde, bir görüntü sınıflandırması “hangi nesne” kısmında rol oynayabilir.

Görüntülerden, kenar bilgileri, morfolojik işlemler [7], istatistiksel yöntemler, dijital görüntü korelasyonu [8] ve model eşleştirme [9] kullanılarak çatlaklar tespit edilebilir. İşlem, görüntünün yakalanmasından, işlenmesinden, çatlak özelliklerinin çıkarılmasından ve çatlağın tanımlanmasından geçer. Çatlak piksellerinin yoğunluğunun diğer yakındaki piksellere kıyasla daha koyu olduğu varsayılır. Bir görüntünün komşu pikselleri arasındaki kontrast, çatlak ve çatlak olmayan bölgeleri sınıflandırmak için karşılaştırılır [10]. Piksel yoğunluğuna bağlı olarak, istatistiksel yöntemler kullanılarak çatlak ve çatlak olmayan pikseller arasında sınıflandırma yapmak için bir eşik ayarlanır [11]. Pikselin çatlak veya çatlak olmayan piksel olarak sınıflandırılma olasılığını belirlemek için farklı yoğunluklar kullanılır [12]. 2D görüntünün bir çatlak niceleme yöntemi geliştirilmiştir. Siyah beyaz görüntü ayrılır, filtrelenir ve beş morfolojik işlemle geliştirmek için incelenir. Bağlı bileşenler çatlakları etiketler ve genişlik, uzunluk, alan, en boy oranı ve yönelimi belirler [13]. Üstyapı görüntülerinden istatistiksel özelliklerin çıkarılması da çatlak görüntülerini bölütlere ayırmak için kullanılır [14]. Yolların bakımı ve onarımı için, önce görüntüdeki gürültüyü azaltan ve ardından çatlakları vurgulayan çatlak algılama ve sınıflandırma yöntemi sunulmaktadır [15]. Bir karar ağacı kullanarak özellikleri öğrendikten sonra, yol görüntülerindeki çatlakları tespit etmek için istatistiksel sınıflandırıcı uygulanır. İki aşamalı çatlak tespit yöntemi, istatistiksel filtrelemeye dayalı olarak tasarlanmıştır. İlk olarak, çatlak özelliklerinin çıkarılması ve ardından çatlağın bölümlere ayrılması [16]. Grafik tabanlı yöntem, çatlak özelliklerini belirlemek ve daha sonra çatlağı tespit etmek için kullanılır,

ancak bu yöntem, çatlağın başlangıç ve bitiş noktasının manuel olarak eklenmesini gerektirir [17]. Bu problem [18]'de önerilen çatlak tespiti ve izleme algoritması çatlak tespiti için çatlak özelliklerini çıkardığında çözülmüştür. Sınıflandırma, çatlak ve çatlak olmayan görüntüyü sınıflandırmak için bir Destek Vektör Makinesi (DVM) kullanılarak histogram yöntemiyle de gerçekleştirilir [19]. Sınıflandırma yöntemleri, gri tonlamalı değerleri karşılaştırarak görüntüyü çatlak ve çatlak olmayan bölgelere ayıran ikili ağaç ve geri yayılıma dayanmaktadır.

Otsu'nun yöntemi, görüntülerin bölütlenmesinde iyi bir yardım sağlar [20]. Filtreler ayrıca çatlak görüntüsünün ikili sürümlerini birleştirerek çatlakları tespit etmek için kullanılır. Orijinal görüntü, farklı yönlerdeki filtrelerle kıvrılır [21]. Bir eşik değerine bağlı olarak, arka plan pikselleri ön plan piksellerinden ayrılır ve Sobel'in filtresi paraziti ortadan kaldırır. Son olarak, büyük çatlakları tespit etmek için Otsu'nun yöntemi uygulanır [22]. Çatlak tespiti, MATLAB tabanlı araç kutusu CrackIT [23], en az 2 mm genişliğindeki ince çatlakları tespit etmek ve karakterize etmek için önerilmiştir. Beton yapıdaki çatlaklar, önce görüntünün gri tonlamaya dönüştürülmesi ve ardından çatlağın tespiti için Sobel filtresi uygulanarak tespit edilir.

Gabor Filtreleri veya Gabor dalgacıkları görüş alanında yaygın olarak kullanılmaktadır. Bilgisayar bilimindeki eğilim, bunların yerini özgürce öğrenilen öznitelik çıkarıcılar alıyor gibi görünse de, yön bilgisini kodlama yeteneklerinden dolayı [24], doku tabanlı özellik çıkarımı için kullanılabilir ve sınıflandırmada çok önemlidir [25]. Segmentasyon [26] ve kenar algılama [27] gibi bilgisayarla görme görevlerinin çoğunda. Daha genel olarak, Gabor Filtrelerinin yanıtları, insan görsel sistemindeki V1 hücrelerinin yanıtlarına benzer olduğundan, görüntü temsil çalışması için kullanılabilir [28, 29]. Görüntü temsilini incelemek önemlidir çünkü bir görüntünün hem yüksek hem de düşük seviyeli tanımını ortaya çıkarabilir. Basit bir görüntü temsili örneği olarak, evrişimli katmanlardan çıkarılan öznitelikler, bir görüntünün bir tür düşük seviyeli açıklaması olarak düşünülebilir. Daha pratik ve uygulama tarafında, Gabor Filtreleri, birkaç isim vermek gerekirse, yüz tanıma [30, 31], parmak izi tanıma veya parmak izi tarama gürültü giderme [32] için bilgisayarlı görme algoritmalarında kullanılabilir. İmgelerden çatlak tespiti için kullanılan diğer yöntemler ve yöntemlerin kullanıldığı çalışmalar Tablo 1.1'de gösterilmiştir.

Tablo 1.1. Çatlak tespiti etmek için kullanılan metotlar

Yöntem	Yapılan Çalışma
Çapraz Entropi Kaybı (ÇEK)	[33, 34, 35, 36]
Weighted Kaybı	[37]
Softmax Kaybı	[38]
Hinge Kaybı	[39]
Log Kaybı	[40]
Regresyon	[41, 42]
Ortalama Kare Hata Kaybı	[41]

3. YAPAY ZEKÂ ÖĞRENME MODELLERİ VE YÖNTEMLERİ

Yapay zekâ ile ilgilenen bilim insanları her zaman daha zeki bir sistem tasarlamak için çaba göstermişlerdir. İnsan düşünce yapısını, karar verebilme yeteneğini ve bu gibi diğer insana ait özellikleri taklit etmek temel hedefleridir.

Tezin bu bölümünde ESA, DÖ, DÖ modelleri, YSA ve MÖ gibi mimariler ve yapılar hakkında kısaca bilgi verilmektedir.

3.1. Evrişimsel Sinir Ağı

Bir evrişimli sinir ağı, bir özellik çıkarma işlemi için kullanılan işlemde evrişim kelimesinin geldiği bir tür YSA'dır. Bir YSA, en basit haliyle, bu düğümlerden gelen çıktıyı belirleyen bir aktivasyon fonksiyonuna sahip bir dizi düğümdür. Burada aktivasyon fonksiyonu, ağırlık matrisi (W^T) ve bir girdi (x) artı bir önyargı terimi (b) çarpımı üzerine uygulanan doğrusal olmayan bir fonksiyon (g) olarak Denklem 3.1'deki gibi hesaplanır.

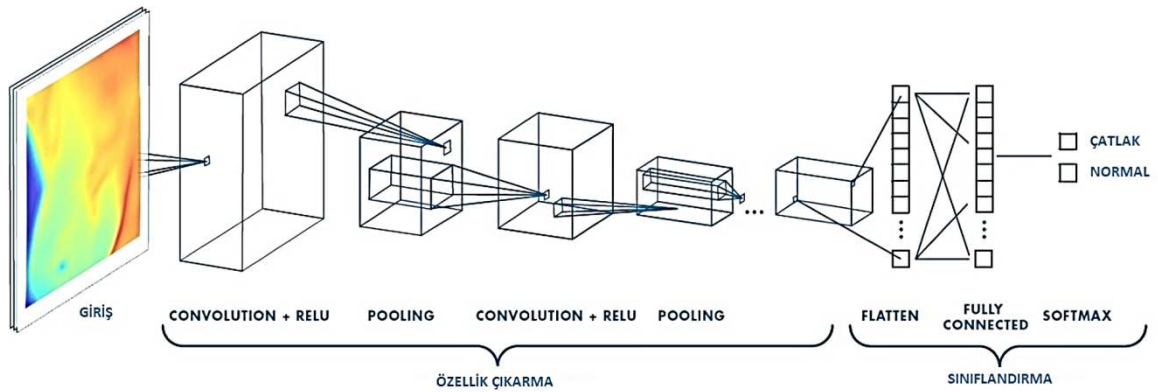
$$f(x) = g(W^T * x + b) \quad 3.1$$

Bu düğüm kümesi hiyerarşik bir yapı oluşturmaya başladığında ve derinleştğinde (yani çoklu gizli katmanlar), genellikle Derin Sinir Ağı olarak adlandırılırlar. Bu düğümlerin diğer düğümlerle bağlantıları olabilir. Tipik bir ileri beslemeli ESA'nın iki farklı bölümü vardır: konvolüsyonel katmanlar ve tam bağlantılı katmanlar. Evrişimsel katmanlar öznitelik çıkarıcılarıdır. Seyrek olarak bağlanırlar ve yerel bağlantılara sahiptirler. Hesaplama açısından bu, parametre ağırlıklarının bir katman boyunca paylaşıldığı anlamına gelir, bu da daha hızlı hesaplama ve daha az bellek gereksinimine yol açar. Tamamen bağlı bir katman yoğun bir şekilde bağlanır ve çıkarılan özellikleri bir çıktı formuna sıkıştırır ve dönüştürür.

Doğrusal olmayı yakalama gücü, temel alınan veri kümesinin doğrusal olmadığı varsayıldığında, bir sınıflandırma sisteminin önemli bir yönüdür. Bunun nedeni, tüm bilgilerin doğrusal bir şekilde modellenemeyeceğidir. Örneğin, bir regresyon yöntemi, görünüşte açık korelasyona sahip değişkenler arasındaki bazı doğrusal ilişkileri yakalamak için yeterli olabilir, ancak görüntü veri setindeki karmaşıklığı gerçekten yakalayamaz. Tipik YSA, sistemi bu doğrusal olmayan durumla başa çıkma gücüyle güçlendirmek için ReLU (Düzeltilmiş Doğrusal Birim) [43] gibi yöntemler kullanır. Nörofizyolojik bağlamda, ReLU ayrıca sistemi biyolojik bir nöronla uyumlu hale getirir çünkü ReLU'dan gelen çıktı her zaman pozitif olacaktır.

Görüntüler her zaman hizalı veya aynı biçimde değildir. Bir bilgisayar monitörünü görüşümüzün neresinde olursa olsun algılayabiliriz. Öte yandan, ESA'lar için her zaman bu kadar basit değildir. ESA'lar, evrişimli katmanlarla birlikte, özellik çıkarma işlemine yardımcı olmak için havuz katmanlarını kullanır. Bir havuzlama katmanının iki sorumluluğu vardır. İlk olarak, bir

havuz katmanı güçlü aktivasyonları geçirir ve zayıf aktivasyonları filtreler. Bu, ağıın genelleme kabiliyetine yardımcı olur. Aktivasyonları bir araya toplayarak sezgisel olarak düşünerek, bir nesnenin uzamsal konumunun veya özelliklerinin yalnızca genel eğilimi bir sonraki katmana aktarılır. Bu, ağıın, bir nesnenin tek bir örneğinin belirli özelliklerine değil, bir nesnenin genel özelliklerine göre karar vermesini sağlar. İkincisi, özellikleri altörnekler. Özellik uzayı sonraki evrişim katmanlarına doğru genişledikçe, hesaplamada bir patlamayı önlemek için bireysel özellikleri küçültmek önemlidir.



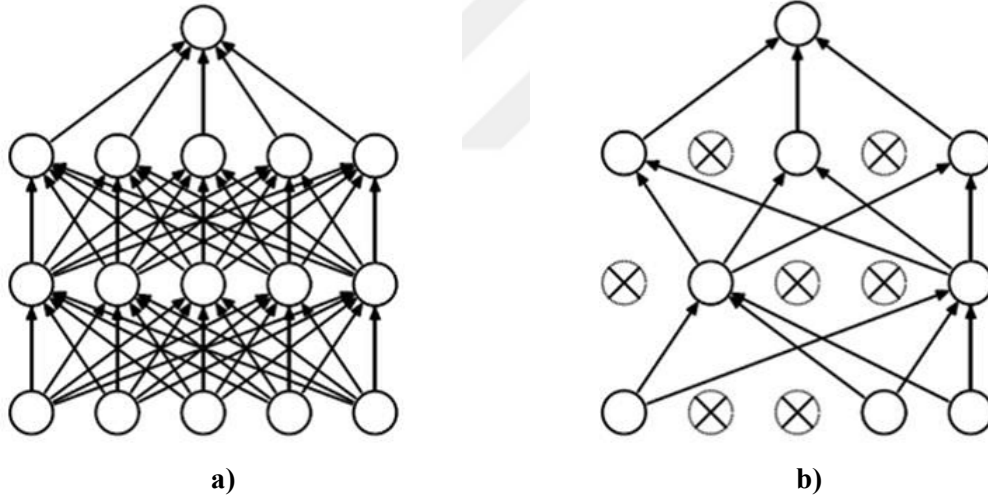
Şekil 3.1. Bir ESA'nın genel mimarisi

Veri setinde bulunan bir imgenin ESA katmanlarındaki genel görünümü Şekil 3.1'de gösterilmektedir. Bir görüntü sınıflandırma alanında, ESA'ları kullanmak için sabit bir kural yoktur. MNIST [44] gibi daha az karmaşık bir veri kümesi üzerinde sınıflandırma, evrişim olmadan kolayca iyi bir performans elde edebilir. Bununla birlikte, görüntü veri kümesinin karmaşıklığı ve boyutu büyüdükçe, öznetelikleri çıkarabildikleri ve bu özellikleri verimlilikle daha az karmaşık bir boyuta soyutlayabildikleri için ESA'lar giderek daha kullanışlı hale gelebilir. Ayrıca, havuzlama katmanlarının kullanımı, bilgi içinden geçtiği her an, bilginin genişliğini ve yüksekliğini faktörlere göre keserek bu süreci hızlandırır. Ayrıca, havuzlama katmanları, ESA'ları nesne özelliklerinin uzamsal konumundaki değişikliklere karşı daha değişmez hale getirerek daha sağlam hale getirir.

Daha önce açıklandığı gibi, ESA'lar iki ana bölüme ayrılabilir. İkinci kısımda kullanılabilecek bilgileri çıkaran bir özellik çıkarıcı, bir sınıflandırıcı. Bir sınıflandırıcı, bir özellik çıkarıcıdan çıkarılan bu bilgileri alır ve amaca göre kararlar vermek için bunları işler. Sezgisel olarak düşünüldüğünde, ideal bir sistemde karar verme sürecini olumsuz etkileyebilecek her türlü bilginin karar sürecinden önce ele alınması gerektiği söylenebilir. ESA bağlamında, karar sürecini olumsuz bir şekilde etkileyebilecek bilginin gürültü kısmını ortadan kaldıran veya manipüle eden bir özellik çıkarıcıya indirgenecektir.

Bir ESA'nın eğitilme işleminde öncelikli olarak model oluşturulur. Oluşturulan modelde havuzlama katman sayısı, tam bağlantılı katman sayısı ve konvolüsyon katmanı ve sınıflandırma katmanı gibi yapılar belirlenir. Daha sonra başlangıç değişkenleri tanımlanır. Bunlar filtre boyutları, filtre sayısı ve adım kayma miktarıdır. Her bir filtre için rastgele 0-1 aralığında değer atanır. Bu modele giriş verisi olarak bir imge verilir. Giriş katmanından verilen imge ağdaki tüm katmanlardan geçerek sonuç elde edilir. Bu olay ileri besleme olarak adlandırılır. İleri beslemede görüntünün piksel değerleri ile her bir filtrenin ağırlığı çarpılıp toplanarak bir sonraki katmana aktarılır. Ağdan beklenen sonuç ile ağın üretmiş olduğu sonuçların toplam farkları alınarak toplam hata hesaplanır. Bu hata değerinin ağda bulunan tüm ağırlıklara dağıtılması gerekmektedir. Ağırlıkların güncellenmesiyle toplam hatanın düşürülmesi sağlanır. Ağırlıkların güncellenmesi olay Geri Yayılım Algoritması (GYA) olarak adlandırılır.

Mevcut eğitim verilerinin tamamını öğrenme durumu ezberleme (Over Fitting) olarak adlandırılır. Bu ezberlemenin önüne geçebilmek için bazen ağın bazı düğümlerinin kaldırılması gerekmektedir. Şekil 3.2 a)'da Standart bir ESA ağı gösterilirken, b) 'de DropOut katmanından sonra kullanılmayan bazı düğümler gösterilmiştir.



Şekil 3.2. a) ESA'nın orijinal yapısı b) DropOut katmanından sonraki hali

3.2. Makine Öğrenmesi

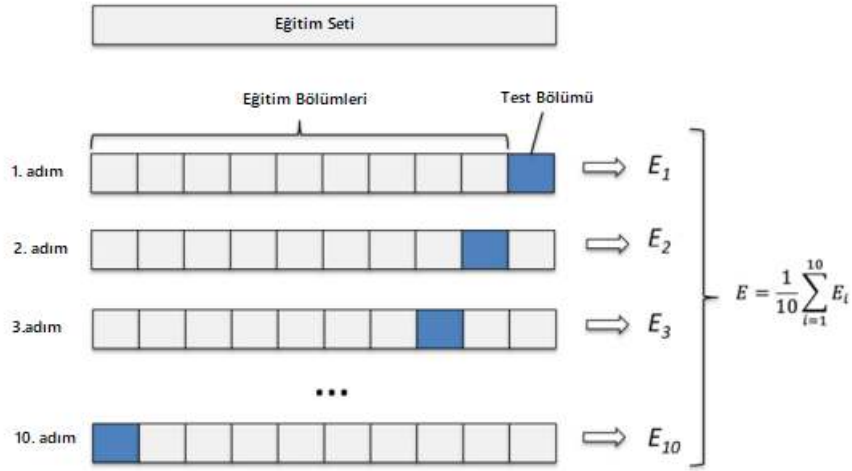
MÖ, yapay zekânın temel olarak birbiriyle ilişkili iki konuya odaklanan alt alanını tanımlamak için kullanılan bir terimdir. Bunlardan birincisi; deneyim yoluyla otomatik olarak gelişen bilgisayar sistemleri nasıl inşa etmek. İkincisi ise bilgisayarlar, insanlar ve kuruluşlar da dâhil olmak üzere tüm öğrenme sistemlerini yöneten temel istatistiksel-hesaplama-bilgi-teorik yasalarını açıklamak [45].

MÖ algoritmaları ve modelleri tarafından çözülen sorunlar, denetimli veya denetimsiz olarak kategorize edilir. Denetlenen problemlerde modeller, daha sonra diğer verileri tahmin etmek için kullanılabilecek verilerdeki kalıpları tespit etmek üzere eğitilir. Denetimsiz öğrenme algoritmaları yapısal özellikleri bulmaya ve bunlara dayalı olarak küme verilerini bulmaya çalışır [45]. Ayrıca, denetlenen problemler iki alt probleme, sınıflandırma ve regresyon görevlerine ayrılmıştır. Her iki problem de etiketlenmiş verileri kullanır, ancak sınıflandırma problemlerinde model verileri doğru şekilde kategorize etmeyi öğrenir ve çıktısı kesiklidir. Regresyon problemlerinde, model bunun yerine girdi verilerine dayalı olarak sürekli bir çıktı hesaplar [46].

Denetimli öğrenmede modeller, genellikle eğitim verileri [47] olarak adlandırılan etiketli bir veri kümesi kullanılarak eğitilir. Model, çıktıyı tahmin ederek ve çıktıyı temel gerçekle daha iyi eşleştirmek için parametrelerini düzelterek yinelemeli olarak eğitilir. Modelin tahmin performansı genellikle, modelin eğitim sırasında görmediği verilerden oluşan test seti olarak da adlandırılan ayrı bir veri setinde değerlendirilir. Tahmin performansının eğitim setinde değerlendirilmemesinin nedeni, modelin performansını gerçek dünya kullanım senaryosunda görünmeyen veriler üzerinde benzetim yapmaktır. Bu durum genelleme sorunu olarak ifade edilir [46,48].

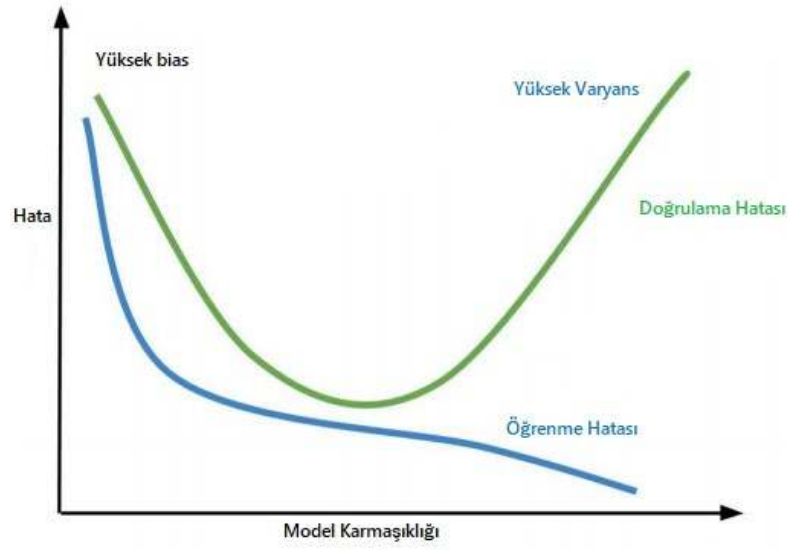
Bir modelin genellemesini tahmin etmek için eğitim bittikten sonra sınıflandırılan bir test seti kullanılabilir. Ancak, yalnızca bir modeli test ederek gerçek performansı tahmin ederken çok fazla belirsizlik söz konusudur. Bu nedenle, Çapraz Doğrulama (ÇD) genellikle k sınıflandırıcılarını eğitmek için kullanılır. Bazı durumlarda, her bir katlamanın benzer bir sınıf dağılımına sahip olmasını sağlamak için katmanlı bir ÇD kullanılabilir. Burada her bir eğitim E ile temsil edilir. ÇD, verileri k bölümlere ayırır ve her sınıflandırıcı, bu bölümlerin $k-1$ 'i üzerinde eğitilir ve ardından Şekil 3.3.'de olduğu gibi k 'ıncı bölümde değerlendirilir [48]. Dolayısıyla tüm bu modeller, verilerin farklı bölümleri üzerinde eğitilir ve değerlendirilir ve performansları, tek bir model kullanmaktan daha doğru bir şekilde, sınıflandırıcının gerçek performansını Denklem 3.2' de gösterildiği gibi, yani genelleme becerisini tahmin etmek için kullanılır.

$$\text{Doğru per fomans} = \frac{1}{k} \sum_{i=1}^k \text{Per fomans}_i \quad (3.2)$$



Şekil 3.3. ÇD'de veri bölümlenme

Eğitimin amacı, yeni verileri iyi genelleleyen bir model üretmektir. Bu nedenle, eğitim verileri üzerinde iyi performans gösteren ancak görünmeyen veriler üzerinde kötü performans gösteren bir modele sahip olmak istenmeyen bir durumdur ve modelin eğitim verilerine gereğinden fazla uyduğu söylenir [46]. Aşırı uyum, birden çok faktöre bağlı olabilir; bunlardan biri, kullanılan modelin eldeki veriler veya sorun için çok karmaşık olmasıdır. Aşırı uyumun anlamı, ESA modelinin eğitim verilerini hızlı öğrenmesi karşısında test verilerinin bu eğitim hızıyla aynı performansı göstermemesidir [49]. Bu, herhangi bir nedenle eğitim verilerinin gerçek veri dağılımını temsil etmemesi durumunda modelin kötü performans göstereceği anlamına gelir. Büyük varyansı olan bir modeli engellemek veya azaltmak için, düzenleme teknikleri kullanılır. Bunlar hangi modelin kullanıldığına bağlıdır. Bir düzenleme tekniği, eğitim sırasında modelin tüm parametrelerinin aşırı uyumu önleyen aynı ölçekte olmasını zorlayan bir L2 / sırt regresyonu uygulamaktır [50]. Spektrumun diğer tarafında, çok fazla düzenlenmiş bir model, eğitim verilerinden yeterince öğrenemeyecek bir modele yol açabilir. Buna yetersiz uyum denir ve model karmaşıklığını artırarak veya daha az düzenleyici kullanarak çözülebilir. Diğer düzenleme tekniği ise sönümlemedir. Bazı sınırları belirli bir oranda elimine ederek başarıya ulaşmayı sağlar. Ancak olumsuz yönü maliyet fonksiyonunu iyi tanımlamamasıdır. Çok büyük veya küçük varyansı olmayan ve iyi genelleştiren bir model seçme problemi, Şekil 3.4.'de gösterildiği gibi "sapma-varyans ödünleşimi" olarak adlandırılır [46, 51].

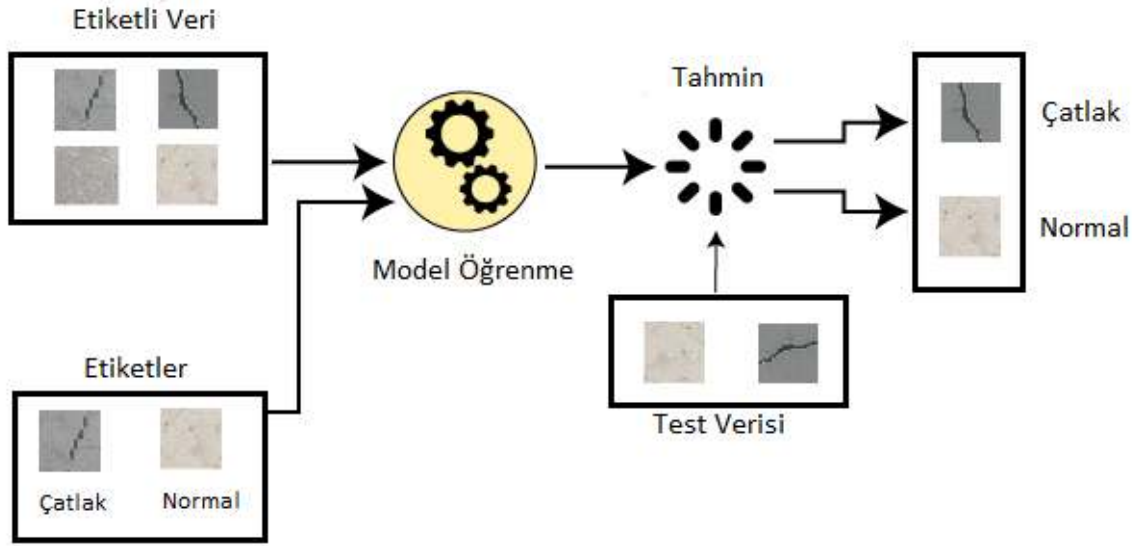


Şekil 3.4. Modelin eğitim ve doğrulama hatası ölçümünde varyans ve standart sapmanın etkisi

3.1.1. Denetimli Makine Öğrenmesi

Denetimli öğrenme, makinelerin iyi "etiketlenmiş" eğitim verileri kullanılarak eğitildiği ve bu verilere dayanarak makinelerin çıktıyı tahmin ettiği makine öğrenimi türleridir. Etiketli veriler, bazı girdi verilerinin zaten doğru çıktıyla etiketlendiği anlamına gelir. Denetimli öğrenmede, makinelere sağlanan eğitim verileri, makinelere çıktıyı doğru tahmin etmeyi öğreten gözetmen olarak çalışır. Öğrencinin öğretmen gözetiminde öğrendiği kavramın aynısını uygular. Denetimli öğrenme, makine öğrenimi modeline girdi verilerinin yanı sıra doğru çıktı verilerini sağlama sürecidir. Denetimli öğrenme algoritmasının amacı, girdi değişkenini çıktı değişkeni ile eşlemek için bir eşleme işlevi bulmaktır.

Denetimli öğrenmede modeller, modelin her bir veri türü hakkında öğrendiği etiketli veri kümesi kullanılarak eğitilir. Eğitim süreci tamamlandıktan sonra model, test verilerine (eğitim setinin bir alt kümesi) dayanarak test edilir ve ardından çıktıyı tahmin eder. Denetimli öğrenmenin çalışması Şekil 3.5'te gösterilmiştir.



Şekil 3.5. Denetimli MÖ tasarımı

‘Çatlak’ ve ‘Normal’ etiketlerine sahip iki türden veri kümemiz olduğu için modelimizi her iki şekil içinde eğitmemiz gerekmektedir. Piksel tabanlı yapılan model öğrenme aşamasından sonra eğitilen model etiketsiz olan test veri seti ile test edilmektedir. Makine zaten daha önceden resimler üzerinden eğitimini tamamladığı için benzer olan test verisi üzerinden çıktıyı tahmin eder. Denetimli öğrenme ayrıca kendi içerisinde 2 alt başlığa ayrılmaktadır.

Regresyon

Girdi değişkeni ile çıktı değişkeni arasında bir ilişki varsa, regresyon algoritmaları kullanılır. Hava tahmini, Piyasa trendleri vb. gibi sürekli değişkenlerin tahmini için kullanılır. Denetimli öğrenme kapsamında olan bazı popüler Regresyon algoritmaları şunlardır:

- Doğrusal Regresyon
- Regresyon Ağaçları
- Doğrusal Olmayan Regresyon
- Bayes Doğrusal Regresyon
- Polinomal Regresyon

Sınıflandırma

Sınıflandırma algoritmaları genel olarak bilinen çıktılar kategorik olduğu zaman kullanılır. Eğer çıktılarımız ‘Evet-Hayır’, ‘Erkek-Kadın’, ‘Doğru-Yanlış’ gibi ikili formatla ifade edilebiliyorsa sınıflandırma algoritmaları kullanılır. Bu algoritmalar:

- Random Forest (Rastgele Orman)
- Karar Ağaçları

- Lojistik Regresyon
- DVM

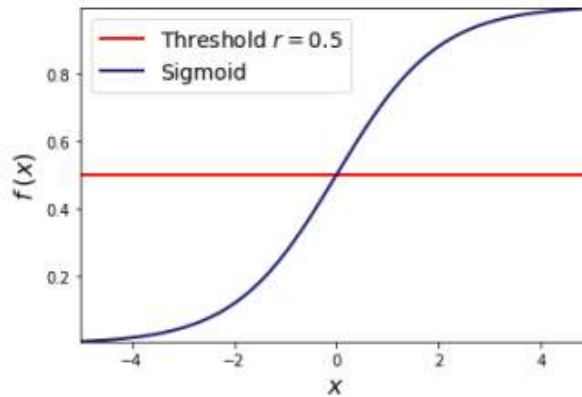
Denetimli öğrenmenin yardımıyla model, önceki deneyimlere dayanarak çıktıyı tahmin edebilir. Bu öğrenme çeşidinde nesnelerin sınıfları hakkında kesin bir fikre sahip olabiliriz. Ayrıca denetimli öğrenme modeli, dolandırıcılık tespiti, spam filtreleme vb. gibi gerçek dünyadaki çeşitli sorunları çözmemize yardımcı olur. Bunun aksine denetimli öğrenme modelleri, karmaşık görevlerin üstesinden gelmek için uygun değildir. Test verileri eğitim veri kümesinden farklıysa doğru çıktıyı tahmin edemez. Bu nedenle eğitim çok sayıda hesaplama zamanı gerektirir.

Denetimli MÖ’de çok popüler olan ikili sınıflandırma algoritmalarından birisi ise yukarıda listelenen Lojistik Regresyon’dur. ‘Çatlak’ ve ‘Normal’ etiketine sahip veri setimiz ikili sınıflandırma örneğidir. Bu nedenle bu bölümde Lojistik Regresyon ve Aktivasyon Fonksiyonları ile ilgili kısaca bilgi verilmiştir.

Lojistik regresyon, ikili sınıflandırma için bir sınıflandırma tekniğidir. Model W ağırlıkları ile parametrelendirilir ve sınıflandırmaya doğrusal regresyonun bir uzantısı olarak düşünülebilir. Doğrusal regresyonda, modelin çıktısı, W ağırlıklarının ve x girdilerinin iç çarpımıdır. Doğrusal regresyonun çıktısı, lojistik fonksiyondan geçirilerek $(0, 1)$ aralığına eşlenir. Lojistik işlev bazen sigmoid işlevi olarak adlandırılır ve Denklem 3.3’te sunulmuştur.

$$\text{sigmoid}(W * x) = \frac{1}{1 + e^{-x}} \quad (3.3)$$

İkili tahmin edicinin çıktısı, $(0, 1)$ aralığındaki değer, genellikle 1 olarak etiketlenen pozitif sınıfa ait bir örneğin olasılığı olarak yorumlanır. Skaler karar eşiği $r \in [0, 1]$ örneğin hangi sınıfa ait olduğunun tahmin edildiğini belirler. Sigmoid fonksiyonu, dikey eksende $r = 0,5$ eşiği işaretlenmiş olarak Şekil 3.6’da çizilmiştir. Bu eşikte, modelin çıktısı 0,5’ten büyükse örnek pozitif sınıfa ait olarak sınıflandırılır.



Şekil 3.6. Sigmoid fonksiyonunun grafiği. $R = 0,5$

Eşik değeri dikey ekseninde 0,5 olarak işaretlenmiştir. Bir numunenin 1 etiketli pozitif sınıfa ait olduğunu tahmin etmek için modelin çıktısı 0,5'ten büyük olmalıdır. Bu, sigmoid fonksiyonunun 0'dan büyük argümanına karşılık gelir. Lojistik regresyon için, sigmoid fonksiyonuna iletilen argüman, model ağırlıkları ve giriş değişkenleri arasındaki iç çarpımdır.

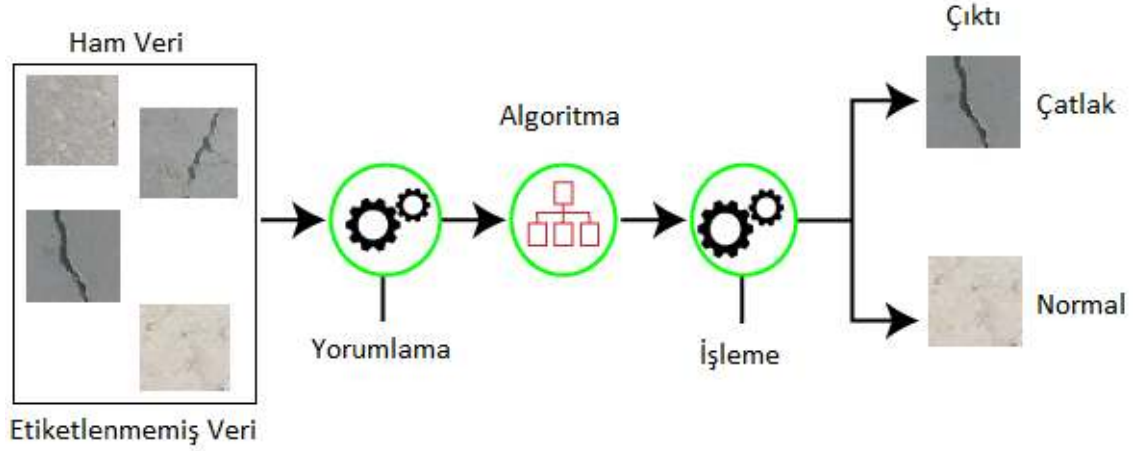
Burada sadece sigmoid fonksiyonunun denklemi ve grafiği açıklanmıştır. Diğer aktivasyon fonksiyonları, denklemleri ve tanım aralıkları Tablo 3.1'de gösterilmiştir.

Tablo 3.1. Aktivasyon fonksiyonlarının matematiksel gösterimi

Aktivasyon Fonksiyon	Denklem	Aralık
Doğrusal Fonksiyon	$f(x) = x$	$(-\infty, \infty)$
Basamak Fonksiyonu	$f(x) = \begin{cases} 0 & \text{ için } x < 0 \\ 1 & \text{ için } x \geq 0 \end{cases}$	$\{0,1\}$
Sigmoid Fonksiyonu	$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$	$(0,1)$
Hiperbolik Tanjant Fonksiyonu	$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{e^x + e^{-x}}$	$(-1,1)$
DDB	$f(x) = \begin{cases} 0 & \text{ için } x < 0 \\ x & \text{ için } x \geq 0 \end{cases}$	$[0, \infty)$
Leaky (Sızıntı) DDB	$f(x) = \begin{cases} 0.01 & \text{ için } x < 0 \\ x & \text{ için } x \geq 0 \end{cases}$	$(-\infty, \infty)$
Swish Fonksiyonu	$f(x) = 2x\sigma(\beta x) = \begin{cases} \beta = 0 & \text{ için } f(x) = x \\ \beta \rightarrow \infty & \text{ için } f(x) = 2\max(0, x) \end{cases}$	$(-\infty, \infty)$

3.1.2. Denetimsiz Makine Öğrenmesi

Etiketlenmiş verilere sahip olmadığımız zamanlarda makine öğreniminde denetimsiz öğrenme kullanılır. Modellerin eğitim verisi kullanılarak denetlenmediği bir MÖ tekniğidir. Bunun yerine modelin kendisi veri setinden gizli kalıpları bulmaktadır. Denetimli öğrenmenin aksine çıktı verilerine sahip olmadığımız için denetimsiz öğrenme bir regresyon ya da sınıflandırma problemine doğrudan uygulanamaz. Algoritma hiçbir zaman verilen veri kümesi üstünden eğitilmez bu da veri kümesi hakkında hiçbir fikri olmadığı anlamına gelir. Denetimsiz öğrenme algoritmasının görevi imge özelliklerini tek başlarına tanımlamaktır. Denetimsiz öğrenme algoritması bu görevi, görüntü veri setini, görüntüler arasındaki benzerliklere göre gruplandırarak gerçekleştirir.



Şekil 3.7. Denetimsiz MÖ tasarımı

Etiketlenmemiş olarak alınan veri seti ve buna karşılık çıktılar da belirlenmemiştir. Bu etiketlenmemiş giriş verileri, onu eğitmek için MÖ modeline beslenir. İlk olarak, verilerden gizli kalıpları bulmak için ham veriler yorumlanır ve ardından k-ortalama kümeleme, karar ağacı vb. gibi uygun algoritma kullanılır. Uygun algoritmayı uyguladıktan sonra, algoritma veri nesnelerini nesneler arasındaki benzerlik ve farklılıklara göre gruplara ayırır. Şekil 3.7’de denetimsiz öğrenme tasarımı gösterilmiştir. Denetimsiz öğrenme ayrıca kendi içerisinde 2 alt başlığa ayrılmaktadır.

Kümeleme

En çok benzerliğe sahip nesnelerin bir grupta kalacağı ve başka bir grubun nesneleriyle daha az benzerliği olacak veya hiç benzerliği olmayacak şekilde nesneleri kümeler halinde gruplandırma yöntemidir. Kümeleme analizi, veri nesneleri arasındaki ortak noktaları bulur ve bunları bu ortak noktaların varlığına ve yokluğuna göre sınıflandırır.

İlişkilendirme

Birliktelik kuralı, büyük veri tabanındaki değişkenler arasındaki ilişkileri bulmak için kullanılan denetimsiz bir öğrenme yöntemidir. Veri kümesinde birlikte oluşan öğelerin kümesini belirler. Birliktelik kuralı, pazarlama stratejisini daha etkili hale getirir. Bazı denetimsiz öğrenme algoritmaları listelenmiştir:

- K-Ortalamalar ile kümeleme
- Hiyerarşik Kümeleme
- Temel Bileşen Analizi (TBA)

- Bağımsız Değişken Analizi (BDA)

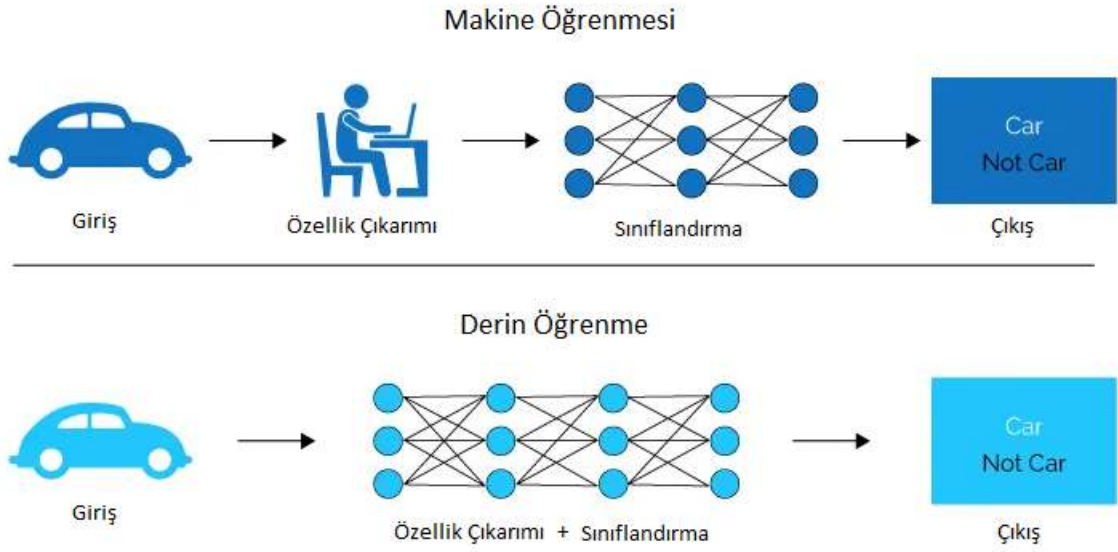
Denetimsiz öğrenme, denetimli öğrenmeye kıyasla daha karmaşık görevler için kullanılır, çünkü denetimsiz öğrenmede etiketlenmiş girdi verileri bulunmamaktadır. Etiketli verilere kıyasla etiketlenmemiş verileri almak daha kolay olduğu için denetimsiz öğrenme tercih edilir. Bunun aksine denetimsiz öğrenme, karşılık gelen çıktıya sahip olmadığı için denetimli öğrenmeye göre özünde daha zordur. Denetimsiz öğrenme algoritmasının sonucu, girdi verileri etiketlenmediğinden ve algoritmalar kesin çıktıyı önceden bilmediğinden daha az doğru olabilir.

3.3. Derin Öğrenme

Bu bölümde temel de MÖ işlemlerine dayanan derin öğrenmeden kısaca bahsedilmiş ve MÖ'den ayrılan kısımlarına değinilmiştir.

DÖ, yapay zekânın bir alt kümesi olan makine öğrenimi dalına dayanmaktadır. Sinir ağları insan beynini taklit ettiğinden ve DÖ yapacaktır. DÖ'de hiçbir şey açıkça programlanmaz. Temel olarak, öznitelik çıkarımı ve dönüşümü gerçekleştirmek için çok sayıda doğrusal olmayan işlem biriminden yararlanan bir makine öğrenimi sınıfıdır. Her bir önceki katmanın çıktısı, ardışık katmanların her biri tarafından girdi olarak alınır. DÖ modelleri, programcıdan küçük bir rehberlik gerektirerek doğru özelliklere odaklanabilecek kadar yeteneklidir ve boyut sorununu çözmede çok yardımcı olur. DÖ algoritmaları, özellikle çok sayıda girdi ve çıktıya sahip olduğunda kullanılır. DÖ, Sinir Ağları yardımıyla gerçekleştirilir ve Sinir Ağı'nın motivasyonunun ardındaki fikir, bir beyin hücresinden başka bir şey olmayan biyolojik nöronlardır.

Temelde DÖ MÖ tekniklerinden oluşmaktadır. DÖ daha karmaşık veriler ve yüksek çözünürlüğe sahip işlemler için kullanılmaktadır. MÖ'nün tasarımında giriş katmanında sonra bir özellik çıkarıcı katman bulunmaktadır. Bu katmanın çıkışında ise sınıflandırma işlemi yapılmaktadır. Fakat DÖ'de özellik çıkarımı yani bilgi elde edinimi işlemi ile sınıflandırma işlemi Şekil 3.8'de ifade edildiği gibi aynı zamanda gerçekleştirilir.



Şekil 3.8. DÖ ve MÖ farkının gösterimi

DÖ günümüzde kendi kendini süren arabalarda, büyük miktarda veri işleyerek etrafındaki görüntüleri yakalayabilir ve ardından sola veya sağa gitmek için hangi eylemlerin dâhil edilmesi gerektiğine veya durması gerektiğine karar verir. Otomatik resim yazısı oluşturmada hangi resmi yüklerseniz yükleyin, algoritma, buna göre başlık oluşturacak şekilde çalışacaktır. Mavi renkli göz dersiniz, resmin altında bir resim yazısı bulunan mavi renkli bir göz görüntülenir. Otomatik makine çevirisinde, DÖ yardımıyla bir dili diğerine dönüştürebiliyoruz. Daha çok günümüzde oluşan karmaşık problemlere çözüm aramak için DÖ algoritmaları kullanışlı ve performanslıdır.

3.4. Derin Öğrenme Modelleri

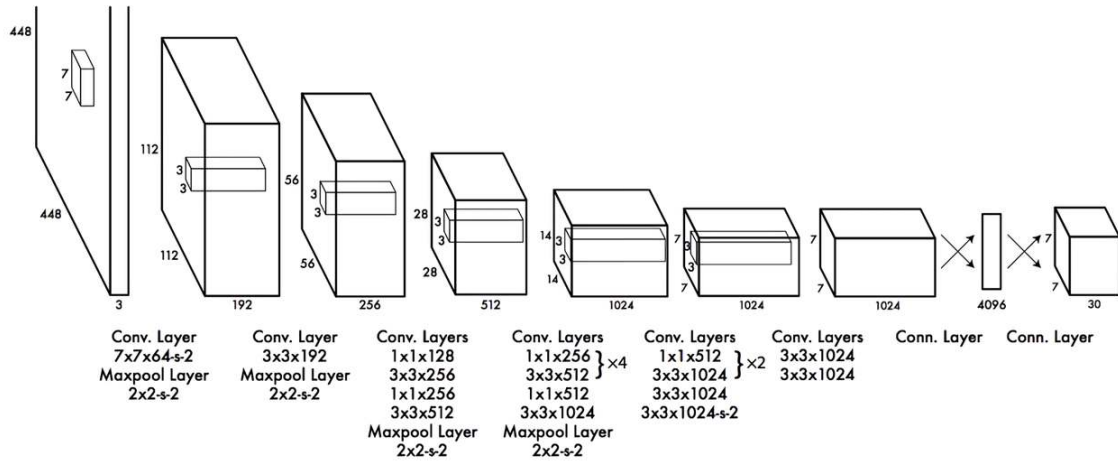
DÖ, belirli görevleri yerine getirmek için YSA kavramları üzerinde çalışan bir makine öğrenimi alt alanıdır. DÖ temelde 4 temel aşamadan oluşmaktadır. Bunlar; Mimarinin tanımlanması, Modelin derlenmesi, Modelin Eğitimi ve son aşama olan Değerlendirme ve Tahminde Bulunma. DÖ' nün temel basamakları sırasıyla üstte sıralandığı gibidir. Araştırmacılar tarafından popüler olarak kullanılan bazı DÖ mimarilerinin kısaca açıklanması bu bölümde yer almaktadır.

3.4.1. Yolo Mimarisi

Tüm yapısı basittir ve yüksek çıkarım hızı nedeniyle, You Only Look Once (YOLO) birçok gerçek zamanlı algılama görevinde uygulanabilir. Bununla birlikte, dezavantaj açıktı, çünkü hücre

sayısı azdı ve her hücrede sadece iki sınırlayıcı kutu öngörülüyordu, YOLO için haritalama, özellikle küçük ve yoğun hedefleri tespit etmede Faster-RCNN [53] kadar iyi değildi.

Bir yıl sonra Redmon ve Farhadi, YOLO'nun [54] ikinci versiyonunu amaçladı. YOLO'nun hedefleri tespit etmede hatalı olduğu ve nispeten düşük bir geri çağırma oranına sahip olduğu sorununu çözmek istediler, bu nedenle Single-Shot-Multi-Box Detector'e (SSD) [55] başvurdular ve algılama için çok ölçekli özellik haritalarını kullandılar. Dahası, YOLO v2, girdi görüntü boyutunu 416×416 'ya büyötmüş ve bunları 13×13 ızgara hüccresine bölmüşdür (her hüccrede 5 tahmini kutu). Sonuç olarak, hem mAP hem de geri çağırma oranları iyileştirildi. İkinci sürüme dayanarak, 2018'de Redmon ve Farhadi, modellerini yeniden geliştirdiler [56].



Şekil 3.9. YOLO Mimarisinin Genel Yapısı [57]

YOLO modeli, yalnızca bir nesneyi sınıflandırmakla kalmayan, aynı zamanda bir görüntüdeki birden çok nesneyi sınıflandırabilen ve bunları kutulayarak kullanıcıya nesnenin görüntüde nerede tanındığını söyleyen en gelişmiş görüntü tanıma algoritmalarından biridir. Bu, bu tür bir nesneye uygulanacak en etkili model olacaktır, çünkü cephede bulunabilen belirli özelliklere sahip farklı nesneyi içeren birçok sınıf oluşturulabilir.

YOLOv3'ü diğer modellerden alınan ağırlıklarla uygulama imkânı da vardır, bu bir avantaj olarak kullanılabilir, çünkü bunu özellik çıkarıcı yöntemiyle uygulamak mümkün olabilir, bu büyük olasılıkla uygun eğitim verileriyle daha yüksek doğrulukta eğitilebilir. Şekil 3.9'da YOLO mimarisi gösterilmektedir.

3.4.2. SSD

SSD, YOLO'nun hızlı çıkarım süresi avantajını benimsedi [55] ve farklı ölçeklerdeki özellik haritalarını tespit ederken RPN'de [58] uygulanan bağlantı kutusu mekanizmasına referans verdi [53]. SSD, son derece hızlı algılama hızını korurken, Faster-RCNN ile benzer bir

performans elde etti. SSD ve Daha Hızlı-RCNN arasında iki büyük fark vardı: SSD, sınırlayıcı kutuları algılamak için farklı ölçeklerdeki özellik haritalarında birkaç bağlantı kurarken, Daha Hızlı-RCNN yalnızca tek bir ölçek dâhilinde işlem yaptı ve SSD doğrudan çok ölçekli algılama ve sınırlama gerçekleştirdi sırasıyla farklı ölçeklerdeki özellik haritalarında kutu regresyonu.

3.4.3. Retina-Net

Geleneksel ÇEK üzerine revize edilen kayıp fonksiyonu olarak odak kaybını kullanmıştır. Hedeflerin ve arka plan verilerinin dengesiz dağılımının, entegre ESA'lara dayalı algoritmaların nispeten düşük doğruluğu ile sonuçlandığını belirtti [59]. Bu nedenle, odak kaybı, Retina-Net'in bu zor örneklerle odaklanabilmesi için eğitim sırasında basit arka plan örneklerinin öğrenme ağırlıklarını azaltabildi.

3.4.4. R-CNN

Nesne önerilerine dayanan ilk algılama algoritmasıydı [60] ve basit bir algılama stratejisi kullandı. Öncelikle, seçmeli arama yoluyla görselden obje önerileri çıkarılmış ve her teklif aynı boyutta ölçeklendirilmiştir. İkinci olarak, özellikleri çıkarmak için ImageNet üzerinde eğitilmiş Alexnet ağı [61] kullanıldı. Son olarak, nesneleri sınıflandırmak için bir DVM sınıflandırıcı kullanıldı. R-CNN, VOC2007 veri setinde [62] şaşırtıcı sonuçlar elde etti, Ortalama Hassasiyet (mAP)% 58,5'e yükseldi. Bununla birlikte, RCNN büyük ilerleme kaydetmiş olsa da, dezavantajları da çok açıktı: eğitimi çok aşamalıydı, bu da sıkıcı ve zaman alıcıydı. Ayrıca, yüksek yoğunluklu önerilen bölgelerde tekrarlanan öznetelik çıkarımı nedeniyle, tespiti son derece yavaştı.

3.4.5. Fast R-CNN

R-CNN [60] ve SPPNet [61] 'e dayanıyordu. Fast-RCNN'nin temel özelliği, çok görevli bir öğrenme yöntemi uygulamasıydı. Hedef sınıflandırmanın ve sınırlayıcı kutu regresyonunun senkron eğitimini sağladı. Eğitim hızı R-CNN'nin 9 katı ve algılama hızı R-CNN'nin 200 katı idi. VOC2007 veri setinde [62], Fast-RCNN mAP'yi% 58.5 'den% 70.0'a çıkardı. Fast R-CNN'nin dezavantajı, daha önce önerilen bölgeleri çıkarmak için hala harici bir algoritmaya ihtiyaç duymasıydı, bu nedenle uçtan uca işlemeyi gerçekleştiremiyordu.

3.4.6. Faster R-CNN

Farklı aşamalarındaki algoritmaların işlevlerini tek bir bütünleşmiş çerçevede birleştirmek istiyoruz, yani hedeflerimizin sınıfını ve konumunu doğrudan tahmin etmek istiyoruz [53]. Nesne önerisi ve sınıflandırmayı ayrı ayrı uygulayan [89, 81] bahsedilen iki aşamalı algoritmalar

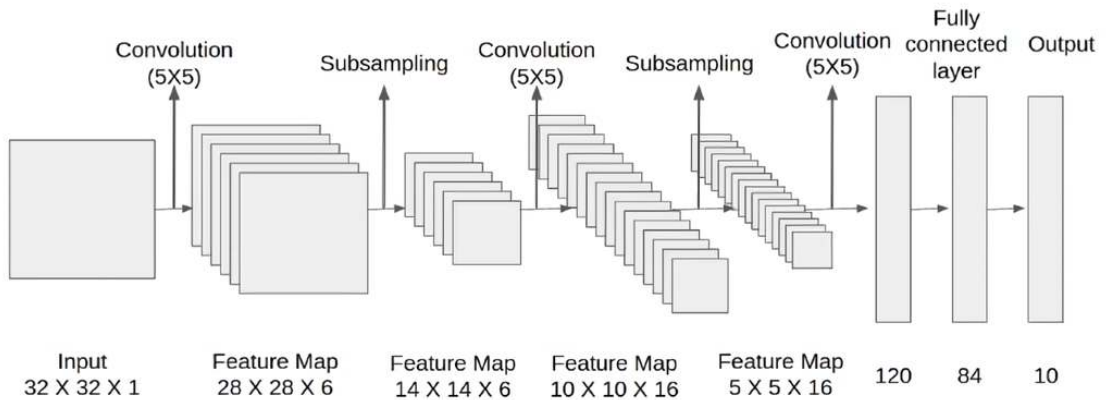
farklı olarak, bu bölümde, bölge önerisi problemini bir gerileme görevi olarak gören tek aşamalı ağlara odaklanıyoruz.

3.4.7. SPPNet

Geleneksel ESA modelleriyle karşılaştırıldığında, evrişimli katman ile tam bağlantılı katman arasına bir uzamsal piramit havuz katmanı ekleyerek tüm ağın giriş boyutunun sabit bir sayıya çarpılmamasını sağladılar [63]. Böylelikle, SPPNet, önerilen bölgeleri ölçeklendirmeden herhangi bir boyutta ve herhangi bir en boy oranı alanında özellik çıkarımını gerçekleştirebilir. Ancak sorun, eğitiminin hala çok aşamalı olmasıydı.

3.4.8. LeNet-5

LeNet-5, orijinal olarak MNIST veri toplama için el yazısı sayıları sınıflandırmak için oluşturulmuş oldukça basit bir evrişimsel ağıdır. ESA'nın ilk adımlarında önemli bir kilometre taşıdır ve orijinal formun önerildiği 1989 yılına kadar uzanır [52]. Model, aralarında maks. Havuzlama bulunan iki evrişimden oluşur, dizi daha sonra düzleştirilir ve daha sonra, sınıflandırma, Şekil 3.10'da gösterildiği gibi bir softmax vektörüne giden iki yoğun katman tarafından yapılır.



Şekil 3.10. LeNet-5 Mimarisinin Genel Yapısı [82]

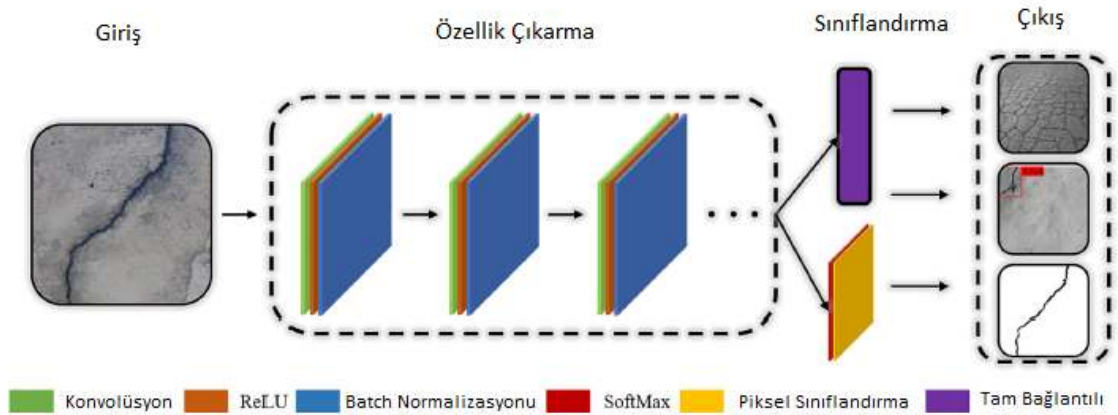
4. MATERYAL VE METOT

Bu bölümde tezin amacına uygun gerçekleştirilen deneysel analizlerde kullanılan veri kümeleri hakkında detaylı bilgiler yer almaktadır. Genel olarak kullanılan yöntemler, teknikler, deneysel analizler ve kullanılan filtrelerin detaylı bilgileri verilmektedir.

4.1. Çatlak Tespiti

Görüntülerden çatlak tespit etmek için genel olarak piksel tabanlı ve kenar algılamaya yönelik çalışmalar yapılmıştır. Dalgacık dönüşümü tabanlı yaklaşım, kaplama görüntülerinin sinyalini Dalgacık Dönüşümü aracılığıyla frekans alanına dönüştürür ve çatlak tespiti için yüksek frekanslı çatlak sinyalini ve çatlak olmayan düşük frekanslı sinyali bölmek için uygun bir eşik değeri ayarlar. Görüntü eşiklemeye yönelik çatlakların gri değerinin, kaplama görüntüsündeki arka plandan genellikle daha düşük olduğu vurgulanır. Gri histogramın olasılık ağırlık faktörünü değiştirerek çatlak tespiti için geliştirilmiş Otsu eşik yöntemini önermektedir [63]. Kenar algılama tabanlı yaklaşımda ise kaplama görüntüsündeki bariz çatlak kenarlarını dikkate alarak, beton çatlak tespiti için filtre tabanlı kenar tespitinin performansını araştırılmıştır [64]. Son olarak yapılan uygulamada çatlak tespiti tamamen denetimsiz bir yöntemle gerçekleştirilmeye çalışıldı [65]. Gri bir görüntü, pikselleri grafikte düğümler olarak kabul ederek bir grafik olarak görüntülendi. Bu şekilde, bir kaplama görüntüsündeki çatlaklar genellikle düşük yoğunluklu sürekli eğriler olarak görünür.

Yapılan bu çalışmalardan hareketle, çalışmamızda ReLU Aktivasyon fonksiyonunu ve Batch normalizasyonunu kullandık. Oluşturmuş olduğumuz ESA mimarisinin katmanları ve çıktıları Şekil 4.1’de gösterilmiştir.



Şekil 4.1. Çatlak Tespiti'nde ESA Mimarisi

4.2. Veri Setleri

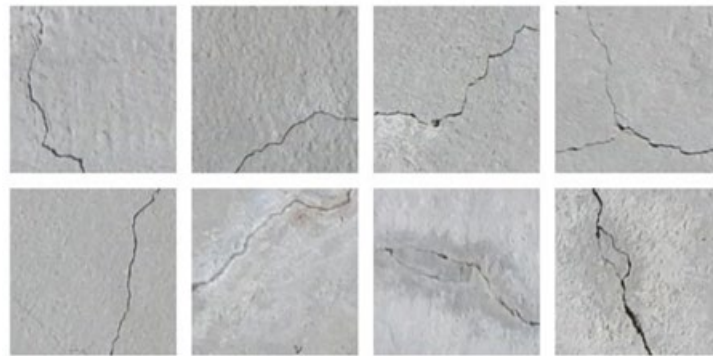
Deneyisel çalışma sırasında, hem modelleri eğitmek hem de modelleri test etmek için veri seti seçmek, değerlerin tutarlılığı açısından son derece önemlidir. Veri setinde bulunan imgelerin formatları (RGB,BGR,HSV) ve piksel sayıları öğrenme ile ters orantılıdır. Bu nedenle yapılan çalışmalarda kullanılan imgeler öncelikle 224x224 boyutunda yeniden boyutlandırıldı. Bu imgeler modelin daha basit öğrenebileceği şekil olan gri formata dönüştürüldü [66]. Bu veri koleksiyonları bu bölümde sunulmuştur.

4.2.1. Beton Çatlak Görüntüleri

227x227 [68] formatında 20.000 çatlaklı görüntü ve 20.000 çatlaksız görüntü içerir. Görüntülerde büyük çatlaklar var ve görüntüler arasındaki renk farklılıkları, SDNET2018 alt kümesine kıyasla nispeten büyük. Bu veri toplama, daha yüksek uç prototipleri eğitmek için değil, deneysel çalışma için tasarlanmıştır.

4.2.2. Basit Veri Kümesi

SDNET2018, beton için yapay zekâ tabanlı çatlak algılama algoritmalarının eğitimi, doğrulanması ve kıyaslaması için açıklamalı bir görüntü veri setidir. SDNET2018, 56.000'den fazla çatlaklı ve çatlaksız beton köprü tabliyesi, duvar ve kaldırım görüntüsü içerir. Veri seti, 0,06 mm kadar dar ve 25 mm kadar geniş çatlakları içerir. Veri kümesi ayrıca gölgeler, yüzey pürüzlülüğü, ölçekleme, kenarlar, delikler ve arka plan döküntüleri dâhil olmak üzere çeşitli engellere sahip görüntüleri içerir. Veri setinde bulunan örnek imgeler Şekil 4.2'de gösterilmiştir. SDNET2018, yapısal sağlık izleme alanında sürekli araştırma konusu olan DÖ, ESA'ya dayalı somut çatlak algılama algoritmalarının sürekli geliştirilmesi için araştırmacılara faydalı olmaktadır [67].



Şekil 4.2. Basit test görüntülerinin alt kümesi, çatlaklar

4.2.3. Gerçekçi Veri Kümesi

Bir modelin ne kadar iyi olduğunu karşılaştırabilmek için gerçeğe bağlanabilecek gerçekçi görüntülere ihtiyaç vardır. Bu veriler, sınıflandırıcının görüntüdeki çatlaklar veya diğer nesneler üzerinde değerlendirmesini ve daha iyi bir genel bakışı sağlamalıdır. Gerçek yol çatlak görüntülerinden alınan bazı resimler Şekil 4.3'te gösterilmektedir.



Şekil 4.3. Gerçekçi veri kümeleri

Beton ve karayolu çatlaklarının da içinde bulunduğu popüler veri setlerinden bazıları Tablo 4.1’ de gösterilmiştir. İçerisinde yüksek çözünürlüklü imgelerin yer almaktadır. Modelin testi ve gerçekleştirimi sırasında tutarsız sonuçların ortaya çıkmasını önlemek için veri büyütme ve döndürme gibi işlemler yapılmamıştır. Tez çalışmasında sınıflandırma, filtreleme gibi birçok diğer işlemde bu veri seti kullanılmıştır [66].

Tablo 4.1. Veri Setlerinin Özellikleri

İsim	Özellikleri
CCD1500	Sahte çatlak, çatlak, yapay çizik, sağlam yüzey ve bitki olmak üzere beş kategoriye ayrılan yaklaşık 1500 görüntüden oluşur.
SDNET2018	56.000'den fazla çatlak görüntüsünden oluşur.
CFD	Gürültü, net olmayan çatlaklardan oluşur. 118 görüntüden oluşur. Tüm görüntü boyutu 448 x 448 pikseldir.
DeepCrack	Karmaşık arka plana sahip 537 çatlak görüntüsü ve net olmayan çatlaklar içerir. Tüm görüntü boyutu 448 x 448 pikseldir.
Cracktree200	Gölgeleri ve düzensiz aydınlatması olan 206 beton kaplama çatlağı görüntüsü içerir. Tüm görüntü boyutu 448 x 448 pikseldir.
Concrete Crack Images[66]*	• Veri seti, çatlaklara sahip somut görüntüler içerir. Veriler çeşitli ODTÜ Kampüs Binalarından toplanmıştır. • Veri seti, görüntü sınıflandırması için negatif ve pozitif çatlak görüntüleri olarak ikiye ayrılmıştır. • Her sınıf, RGB kanallı 227 x 227 piksel ile toplam 40000 görüntü ile 20000 görüntüye sahiptir. • Veri seti, tarafından önerilen yöntemle 458 yüksek çözünürlüklü görüntüden (4032x3024 piksel) oluşturulmuştur. • Yüksek çözünürlüklü görüntüler, yüzey kalitesi ve aydınlatma koşulları açısından farklılıklara sahiptir. • Rastgele döndürme veya çevirme açısından hiçbir veri büyütme uygulanmamıştır.

4.3. Araçlar

Tez çalışması sırasında resimlere ön işlemlerin yapılması, yeniden boyutlandırılması, filtrenin oluşturulması ve modellerin oluşturulması gibi birçok işlemde bazı araçlar kullanılmıştır. Bu bölümde, deneysel çalışma sırasında kullanılan tüm farklı araçların kısa bir açıklaması yapılmaktadır.

Yapılan tez çalışmasında yol anormalisi ve çatlak çeşitleri, çatlak tespiti uygulama alanları ve çatlak tespitinin yöntemleri araştırılmıştır. Çatlak tespitinin gerçekleştirildiği modelin tasarımı için VGG-16 mimarisi ve ESA Mimarisi kullanılmıştır. İlgili kodlar NVIDIA GeForce 4 GB grafik kartı, Intel® i7 -Core 2.80 GHz işlemci ve 16 GB RAM'a sahip bilgisayarda Windows 10 işletim sistemi üzerinde kurulan Python 3.6.4 programında PyCharm ara yüzü üzerinden derlenmiştir. Literatürde ise Çatlak Tespiti için ResNet mimarisi, AlexNet mimarisi, LeNet-5 Mimarisi gibi birçok farklı mimari kullanılmış ve farklı veri setleriyle oluşturulan modellerle farklı yaklaşımlarda bulunulmuştur.

4.3.1. Tensorflow 2.0 Kütüphanesi

Tensorflow (TF), uçtan uca bir MÖ platformudur. Tüm gelişmiş ve zor işlevleri bilmek zorunda kalmadan bir MÖ algoritması oluşturmayı kolaylaştıran geniş bir araçlar, kitaplıklar ve ortak kaynaklar ekosistemi içerir. TF, farklı yapı ve parametrelerle denemeyi oldukça kolaylaştırır ve eğitim süresini büyük ölçüde azaltmak için modeli bir GPU üzerinde eğitmeyi kolaylaştırır [69].

4.3.2. Keras Kütüphanesi

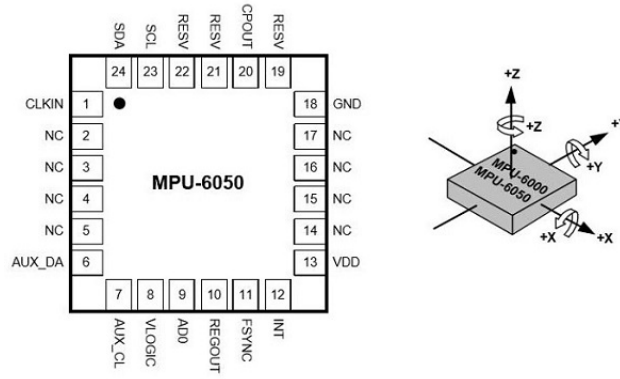
Keras, TF, CNTK veya Theano üzerinde çalıştırılabilen üst düzey bir sinir ağı uygulama programlama arayüzüdür. MÖ ile deney yapmayı kolaylaştırmak ve fikirden sonuca hızla geçebilmek için geliştirilmiştir. Keras modelleri sıralı biçimde oluşturulmuştur ve basit kodlama tasarımları, ek işlevleri, modülleri, kayıp işlevlerini ve etkinleştirme işlevlerini daha değiştirmeyi kolaylaştırır. Modeller, kodu kompakt, hata ayıklaması ve geliştirmesi kolay hale getiren Python'da açıklanmıştır. Keras API ayrıca ImageNet üzerinde, sınıflandırma katmanı [70] ile veya bunlar olmadan kullanılabilen önceden eğitilmiş modellerin kullanımını sunar.

4.3.3. OpenCV Kütüphanesi

Açık kaynaklı bir bilgisayarlı görü ve makine öğrenimi yazılım kitaplığıdır. OpenCV, bilgisayarlı görü uygulamaları için ortak bir altyapı sağlamak ve ticari ürünlerde makine algısının kullanımını hızlandırmak amacıyla oluşturulmuştur. Kütüphane, hem klasik hem de son teknoloji bilgisayarlı görme ve makine öğrenimi algoritmalarından oluşan kapsamlı bir set içeren 2500'den fazla optimize edilmiş algoritmaya sahiptir. Bu algoritmalar, yüzleri algılamak ve tanımak, nesneleri tanımlamak, videolarda insan eylemlerini sınıflandırmak, kamera hareketlerini izlemek, hareketli nesneleri izlemek, nesnelerin 3B modellerini çıkarmak, stereo kameralardan 3B nokta bulutları üretmek, yüksek çözünürlük elde etmek için görüntüleri bir araya getirmek için kullanılabilir [71].

4.3.4. 6 Eksenli ivme sensörü

Piyasada hareket algılamak için birçok sensör ve bu sensörleri çalıştırabilecek birçok yazılım bulunmaktadır. Bu tez çalışmasında araca yerleştirilen (MPU-6050) ivme sensörü ile aracın karayolunun yüzeyindeki çatlaklardan geçişi esnasındaki bilgiler okunmuş ve anlık olarak grafikler oluşturulmuştur.



Şekil 4.4. İvme Sensörünün yapısı

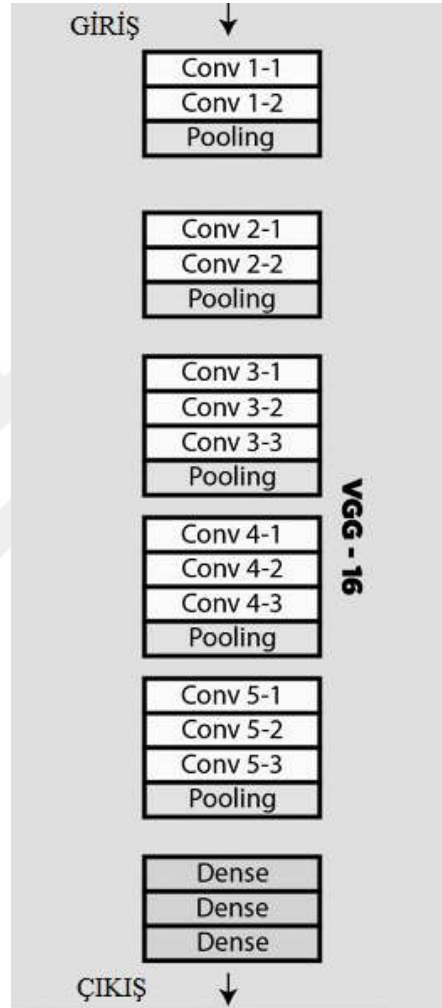
İvme sensörü harekete ilişkin Şekil 4.4’da belirtilen yönlerde (A_x , A_y , A_z , G_x , G_y , G_z) değerler elde etmek mümkündür. Bu değerlerden A_x (Acceleration X) X yönünde meydana gelen hızlanmayı, A_y (Acceleration Y) Y yönünde meydana gelen hızlanmayı, A_z (Acceleration Z) Z yönünde meydana gelen hızlanmayı ifade etmektedir. G değerlerini ise 3 eksenli koordinat düzlemi gibi düşündüğümüzde; G_x , X ekseninde meydana gelen değişikliği, G_y , Y ekseninde meydana gelen değişikliği, G_z , Z ekseninde meydana gelen değişikliği göstermektedir. Şekil 4.4’de detaylı olarak MPU-6050 6 eksenli ivme sensörünün pin numaraları ve her bir pinin ismi gösterilmiştir.

4.4. VGG-16 Mimarisi

Karen Simonyan ve Andrew Zisserman tarafından 2014 yılında duyurulmuştur [72]. VGG-16 modeli 13 konvolüsyon, 3 tam bağlı katman olmak üzere 16 katmandan oluşmaktadır. Katmanlar arasında maksimum havuzlama uygulanmıştır. Şekil 4.5’de VGG-16 mimarisi gösterilmiştir. 224x224x3 boyutlarında resimleri girdi olarak kabul etmektedir. Kullanılan veritabanının %20’sini sınıflandıracak şekilde modellenmiştir.

Sınıflandırmanın yapılabilmesi için kullanılan VGG-16 mimarisi ImageNet veri setinde en iyi performans gösteren model olarak belirlendi. R, G ve B olmak üzere üç kanallı sabit boyutlu 224x224 görüntü olduğu kabul edilir. Yapılan tek ön işleme, her piksel için RGB değerlerinin normalleştirilmesidir. Bu, her pikselden ortalama değer çıkarılarak elde edilir. Görüntü, 3x3’lük çok küçük alıcı boyutundaki 2 evrişim katmanının ilk yığınınan geçirilir, ardından ReLU aktivasyonları gelir. Bu iki katmanın her biri 64 filtre içerir. Evrişim adımı 1 pikselde sabitlenir ve dolgu 1 pikseldir. Bu konfigürasyon, uzaysal çözünürlüğü korur ve çıktı aktivasyon haritasının boyutu, giriş görüntüsü boyutlarıyla aynıdır. Aktivasyon haritaları daha sonra 2 piksellik bir adımla 2x2 piksellik bir pencere üzerinde uzamsal maksimum havuzlamadan geçirilir. Bu, aktivasyonların boyutunu yarıya indirir. Böylece ilk

yığının sonundaki aktivasyonların boyutu 112x112x64'tür. Aktivasyonlar daha sonra benzer bir ikinci yığından geçer, ancak ilkinde 64'e kıyasla 128 filtre bulunur. Sonuç olarak, ikinci yığından sonraki boyut 56 x 56 x 128 olur. Burada uygulanan filtrelerin sayısı 256'dır ve yığının çıktı boyutunu 28 x 28 x 256 yapar. Bunu, her biri 512 filtre içeren üç evrişim katmanından oluşan iki yığın izler. Bu yığınların her ikisinin de sonundaki çıktı 7 x 7 x 512 olacaktır [73].



Şekil 4.5. VGG-16 Mimari Haritası [73]

4.5. Yönlü Gabor Filtresi

Gabor Filtresi, adını Dennis Gabor'dan alan kenar algılama, doku analizi, özellik çıkarma vb. için sayısız görüntü işleme uygulamasında kullanılan lineer bir filtredir. Gabor Filtresi'nde bahsi geçen filtreleme, filtrelenecek imge ile filtre matrisinin konvolüsyonudur. Denklem 4.1' de iki boyutlu Gabor Filtresi çekirdek matrisinin formülü verilmiştir.

Fourier dönüşümü, belirli bir sinyalin frekans özelliklerinin analizi için sinyal işlemede en yaygın kullanılan araçtır. Ancak sinyal dönüşümünden sonra görüntünün farklı bölümlerinden gelen özellikler birbirine karışır ve sinyalin zaman bilgisi kaybolur ve frekansın ve çalışmadaki konumunun belirlenmesi neredeyse imkânsızdır. Bu soruna karşı koymak için ortak alan analizi kullanılabilir. 1946'da Gabor tarafından ortak bir etki alanı işlevi, yani Gabor Filtresi önerildi. Gabor fonksiyonunun alt sınırı elde ettiği kanıtlanmıştır ve ortak alanda en iyi analitik çözünürlüğü gerçekleştirir [74].

Gabor Filtre tabanlı teknikler, doku analizi ve bölütleme [75, 76], kenar algılama [77], nesne tanıma [78,79], görüntülerin gösterimi [80] ve el yazısı rakamlar ve alfabe tanıma [81] gibi çok çeşitli görüş tabanlı uygulamalarda yaygın olarak kullanılmaktadır. Gabor Çekirdekleri bant geçiren filtrelerdir ve görsel yapısındaki çoklu kanal filtrelemesi ile temel görme bilgisi işleme anlayışı tarafından yönetilir [83].

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x^2 + y^2 + y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right) \quad (4.1)$$

Euler formülünden yararlanılarak Denklem 4.1 düzenlenirse Denklem 4.2'de ifade edilen g fonksiyonunun cosinüs formu ve Denklem 4.3'de ifade edilen sinüs formu elde edilir.

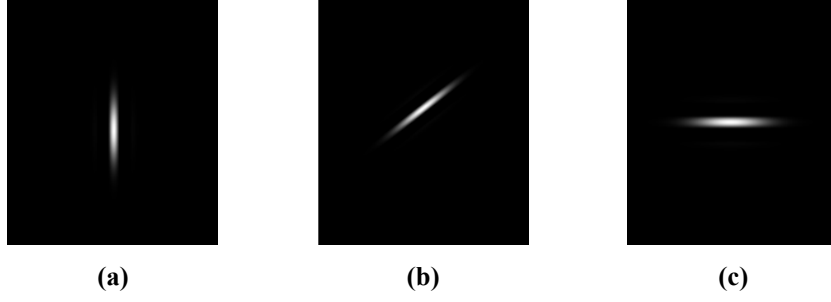
$$g(x, y; \lambda, \theta, \varphi, \sigma, \gamma) = \exp\left(-\frac{x^2 + y^2 + y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \psi\right) \quad (4.2)$$

$$g(x, y; \lambda, \theta, \varphi, \sigma, \gamma) = \exp\left(-\frac{x^2 + y^2 + y'^2}{2\sigma^2}\right) \sin\left(2\pi\frac{x'}{\lambda} + \psi\right) \quad (4.3)$$

İşlemin gerçekleştirilebilmesi için temelde iki adım bulunmaktadır. Bunlardan birincisi iki filtrenin (çekirdek matrisin) hesaplanması ardından imgeye uygulanması ve imgelerin karelerinin toplamının karekökleri alınmasıdır. İkinci adım ise gerçel kısmının hesaplanmasıdır.

λ (Lambda): Gabor fonksiyonunun şeritlerinin genişliğini yönetir. Burada dalga boyunu artırmak daha kalın şeritler üretir ve dalga boyunu azaltmak daha ince şeritler üretir. Diğer parametreleri değiştirmeden ve lambdayı 60 ve 100 olarak değiştirerek şeritler kalınlaşır [84].

θ (Teta): Teta doğrudan formül içerisinde görünmemektedir. Gabor çekirdeğinin yönelim açısıdır. Gabor işlevinin yönünü kontrol eder. Sıfır derece teta, Gabor fonksiyonun dikey konumuna karşılık gelir. Teta açısının değişmesi sonucu Gabor çekirdeğinde meydana gelen değişimler Şekil 4.6' da gösterilmiştir. x' ve y' değişkenleri değişen teta değerleri Denklem 4.4' te incelenmiştir.



Şekil 4.6. a) Gabor çekirdeğinin Theta 0° ‘daki durumu **b)** Gabor çekirdeğinin Theta 45° ‘deki durumu **c)** Gabor çekirdeğinin Theta 90° ‘daki durumu

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (4.4)$$

φ (Fi): Çekirdek matrisinin faz kayma açısıdır. Bu değerin değiştirilmesi sonucunda sinüzoidin x ekseninde kayması sağlanır.

σ (Sigma): Bant genişliği veya Sigma Gabor zarfının genel boyutunu kontrol eder. Aynı zamanda fonksiyonun standart sapmasını belirleyen katsayıdır. Bu değerin yüksek seçildiğinde Gabor dalgacıklarının birbirinden uzaklaşması, düşük seçildiğinde ise Gabor dalgacıklarının birbirine yaklaşması beklenir.

γ (Gamma): En boy oranı veya Gamma, Gabor işlevinin yüksekliğini kontrol eder. Çok yüksek en boy oranı için yükseklik çok küçük olur ve çok küçük Gamma değeri için yükseklik oldukça büyük olur. Gamma değerini 0.5 ve 0.75'e yükselterek, diğer parametreleri değiştirmeden, Gabor fonksiyonunun yüksekliği azalır.

Gabor Filtresi vektörel bir filtre olması nedeniyle yüzey çatlağı algılamada modelin kolayca öğrenebileceği resimler oluşturuldu. Veri setinde bulunan öğrenme ve test sınıfı altındaki çatlak ve normal klasörlerindeki resimlerin her birine Gabor Filtre uygulandı.

5. BULGULAR VE TARTIŞMA

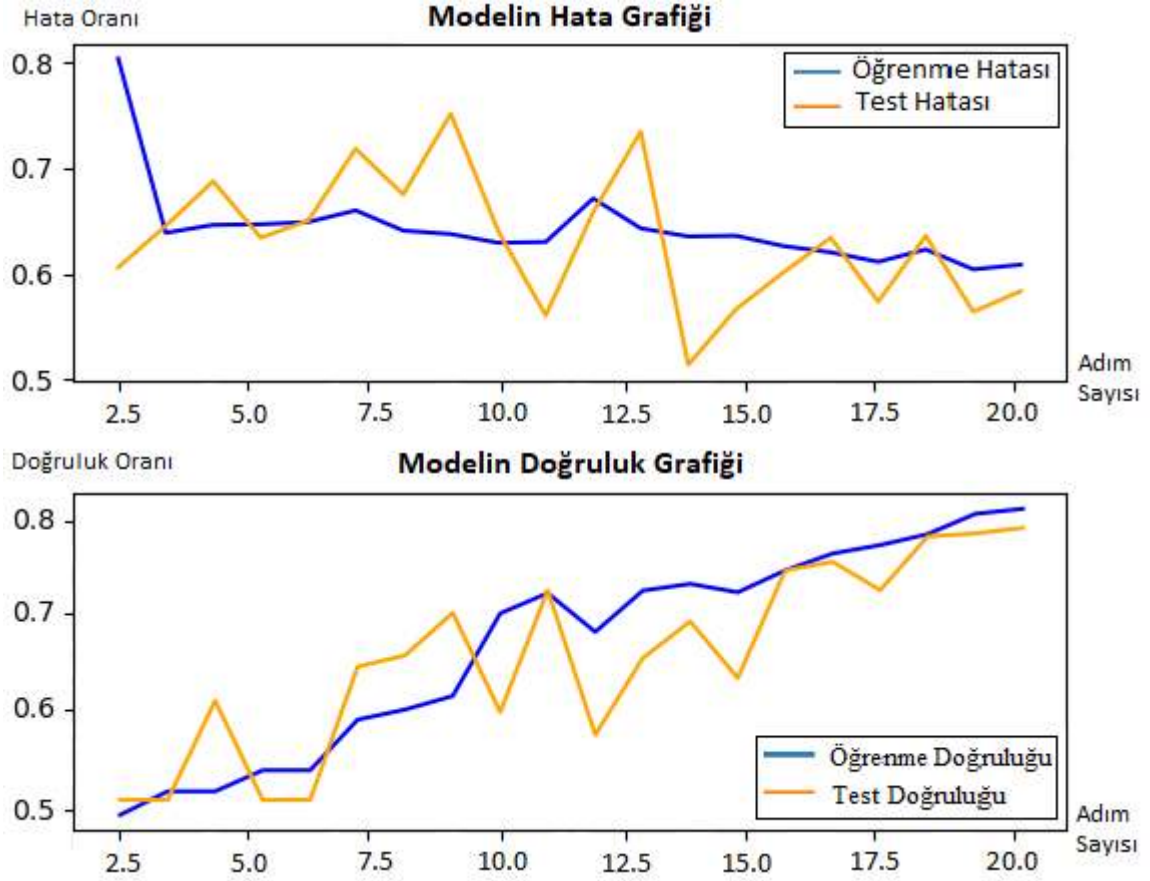
Bu bölümde gerçekleştirilen üç deney analizi hakkında bilgilere yer verilmiştir. Gerçekleştirilen ilk deney de ilgili veri seti ile [85] VGG-16 mimarisi kullanılarak oluşturulan model kullanılarak sınıflandırma yapılmıştır. Yapılan sınıflandırmanın başarı parametrelerinin artması için mimari yeniden iyileştirilmiştir. İkinci deneyde kullanılan çatlak veri seti geleneksel ESA mimarisinden geçirilmiş ve sonuçlar elde edilmiştir. Daha sonra veri setine Gabor Filtre uygulanarak sonuç elde edilmiş ve ESA ile karşılaştırılmıştır. Yapılan son çalışmada ise Gabor Filtre'nin değişen parametreleri ile oluşturulan veri setlerinden modeller test edilmiş ve sonuçlar 6 adet başarı parametresi ile kıyaslanmıştır. Tezin amacı doğrultusunda yapılan çalışmaların analizi için karmaşıklık matrisi ve bazı diğer başarı parametrelerinde faydalanılmıştır.

5.1. Çatlak Sınıflandırmada VGG-16 mimarisinin Kullanımı

VGG-16 mimarisini kullanırken tüm görüntülerin aynı şekle sahip olması için fotoğrafları modellemeyen önce yeniden boyutlandırılması gerçekleştirildi. Daha küçük girdiler, eğitilmesi daha hızlı olan bir model anlamına gelmektedir. Bu yüzden veri setinde bulunan resimleri 224×224 piksellik sabit bir boyuta yeniden biçimlendirdik. Daha sonra veri setimizde bulunan resimlerin rastgele % 25'ini test veri kümesinde geriye kalan %75'lik bölümü ise öğrenme veri kümesinde olacak şekilde tasarlandı.

5.1.1. Bir Bloklü VGG-16 Modeli

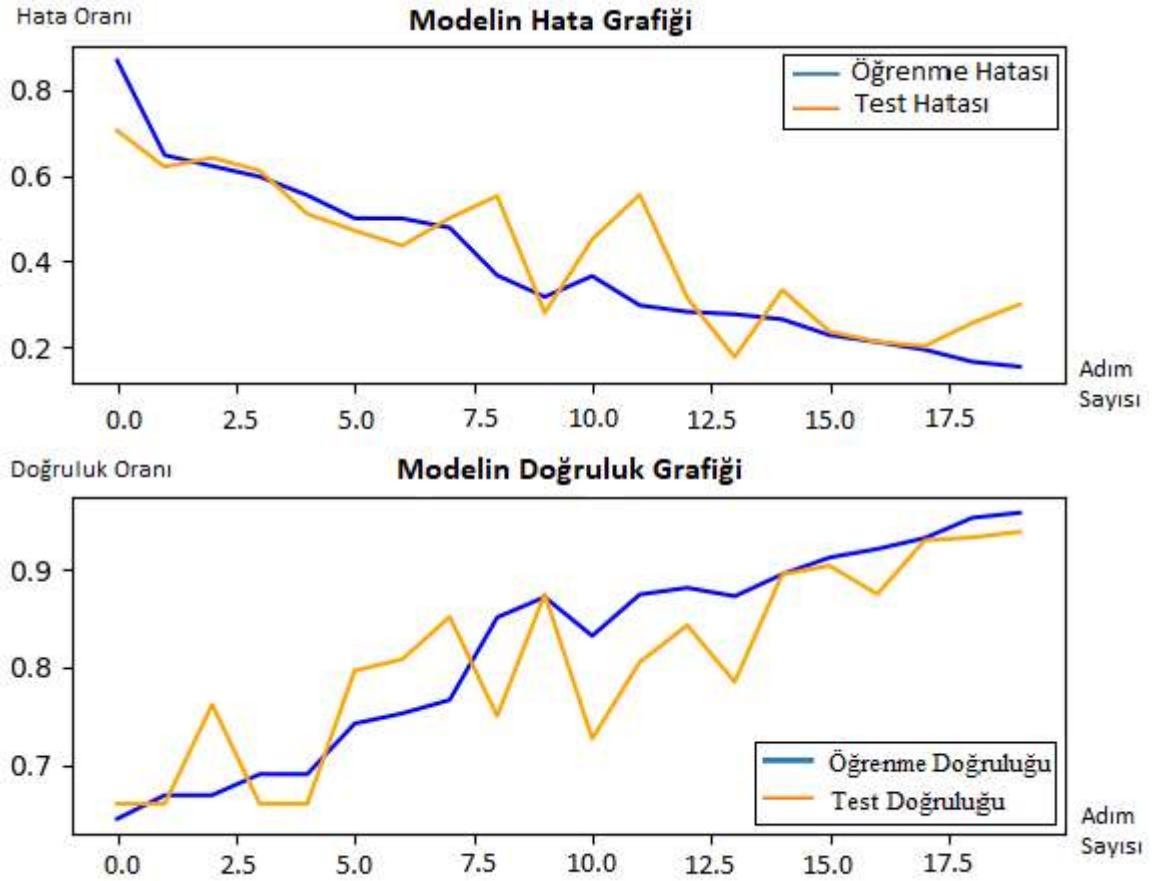
Tek bloklü VGG-16 modeli, 3 x 3 boyutunda 64 filtreye sahip iki ardışık evrişim katmanı ve ardından adım 2 ile 2 x 2 maksimum havuzlama katmanı içerir. Modelin eğitimi 20 adımda gerçekleştirilmiştir. Eğitim sonunda model %83,988'lik bir doğruluk elde etmiştir. Şekil 5.1'de ki ÇEK grafiğinde 0,8'den başlayan kayıp değeri model eğitiminin sonunda 0,6'nın altına inmeyi başarmıştır. Bu durum hem modelin veri seti üzerindeki doğruluğunun artması hem de kayıp değerinin istenilen değer olan 0,1'in altına inmesi için 1 Bloklü VGG-16 çalışmasının yeterli olmadığını göstermiştir.



Şekil 5.1. Çatlaklar ve Normaller Veri Seti’nde Bir Bloklı VGG-16 Modeli için Kayıp ve Doğruluk Öğrenme Modellerinin Çizgi Grafikleri

5.1.2. İki Bloklı VGG-16 Modeli

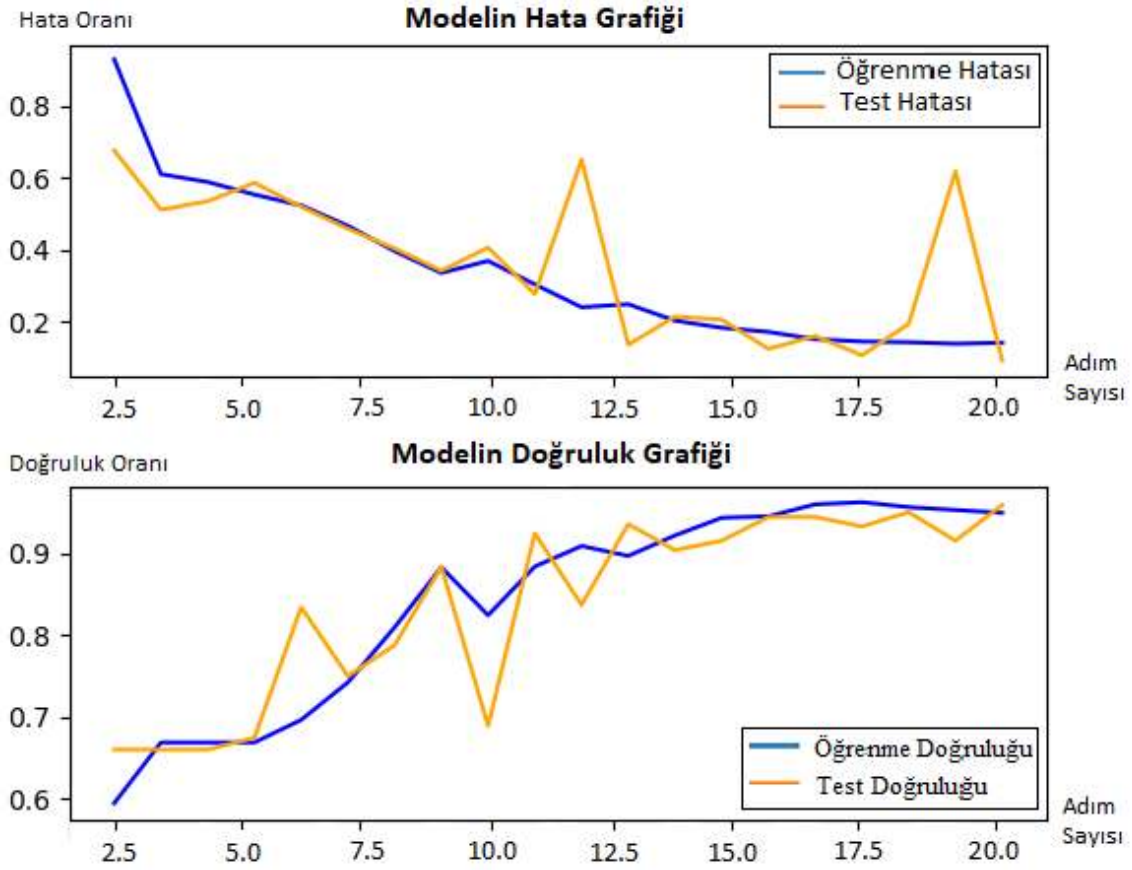
İki bloklı VGG-16 modeli bir önceki tek bloklı VGG-16 modelini genişleterek ve 32 filtreli tek bir evrişimli katmanın yanına 64 filtreli ikinci bir blok daha ekler. Modelin test veri seti üzerindeki başarısı kaydedilir. Modelin performansının bir bloklı VGG-16’ da %83,988 iken Şekil 5.2’de iki bloklı VGG-16’ da %93,895 doğruluğa ulaşmıştır. Öğrenme eğrilerinin grafiğini gözden geçirdiğimizde, modelin eğitim veri setini aştığını görebiliriz. Kayıp değerinin 1 Bloklı VGG-16’ya göre azaldığını ve doğruluk öğrenme modelinin 1 Bloklı VGG-16’ ya göre arttığını görmekteyiz. Sonuçların bu şekilde olması modelin artan kapasitesinin bir sonucudur ve bu eğilimin daha iyi olması ve ÇEK değerinin daha da düşmesi gerekmektedir. Bu nedenle veri setimiz 3 Bloklı VGG-16 mimarisine uygulanacaktır.



Şekil 5.2. Çatlaklar ve Normaller Veri Seti’nde İki Bloklı VGG-16 Modeli için Kayıp ve Doğruluk Öğrenme Modellerinin Çizgi Grafikleri

5.1.3. Üç Bloklı VGG-16 Modeli

Üç bloklı VGG-16 modeli, iki bloklı VGG-16 modelini genişletir ve 128 filtreli üçüncü bir blok ekler. Bu durumda, iki bloklı VGG-16’da %93.895 olan doğruluk performansımız, üç bloklı VGG-16’da %95.930 hassasiyete kadar artış göstermiştir. Şekil 5.3’de ki öğrenme eğrilerinin grafiğini gözden geçirdiğimizde, benzer bir aşırı uyum eğilimi görebiliriz. Kayıp değerinin 0,2’nin altına inmesi ve doğruluk değerinin ise yaklaşık olarak %96’ ya yaklaşmıştır. 4 veya 5 Bloklı VGG-16’ ya modelleri ile yapılan testlerde doğruluk oranı %96’nın üstüne çıkmamıştır. Bu durum bize en iyi modelin 3 bloklı VGG-16 modeli olduğunu gösterir [86].



Şekil 5.3. Çatlaklar ve Normaller Veri Seti’nde Üç Bloklı VGG-16 Modeli için Kayıp ve Doğruluk Öğrenme Modellerinin Çizgi Grafikleri

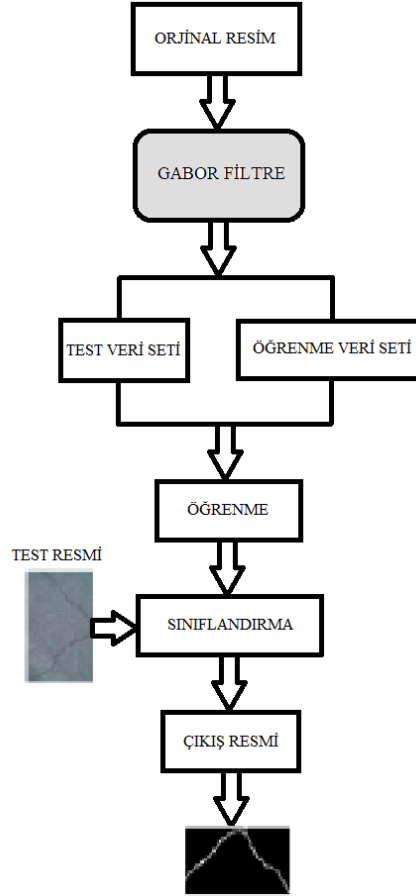
Tablo 5.1. 5 kat veri kümesinde farklı eğitilmiş modeller kullanılarak ortalama sınıflandırma başarıları

Model	En Yüksek Başarı Periyot Num.	Doğruluk Oranı					
		Kat 1	Kat 3	Kat 3	Kat 4	Kat 5	Genel
AlexNet	56	93.33	90.00	86.67	96.67	90.00	91.11
LeNet	64	93.33	86.67	83.33	96.67	83.33	87.50
VGG-16	25	83.98	93.89	95.93	-	-	91,26
RoICNN	13	96.67	93.33	90.00	100.00	96.67	95.34

Çalışmamızdan elde ettiğimiz sonuçlar, daha önce farklı DÖ modelleri eğitilerek ve farklı adımlarda çalıştırılarak yapılan çalışmalardan doğruluk oranı en yüksek olan RoICNN modelinden daha iyi olduğu Tablo 5.1’de görülmektedir [86]. Modelin muhtemelen düzenleme tekniklerinden yararlanacağını sonuçlardan anlaşılmaktadır. Bu ağırlık kaybı ve veri büyütme gibi teknikleri olabilir. Ayrıca, eğitim veri setini genişleterek modeli konumlandırmak için daha fazla değişmeyen özellikleri öğrenmeye teşvik ederek performansı da arttırabiliriz.

5.2. ESA ve Gabor Filtre Kullanarak Çatlak Tespiti

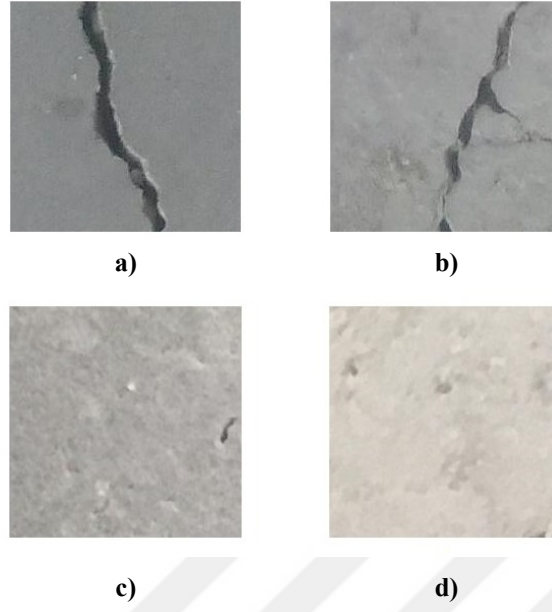
Yüzey çatlaklarının tespiti ile ilgili çalışmalar genellikle tek tip problemlerin çözümüne ve büyük boyutlu imgeler üzerine odaklanmaktadır. Bu çalışmadaki imgelere çeşitli ön işlemler uygulandıktan sonra kullanılan ESA mimarisinde direkt olarak filtre kullanmaksızın öğrenme yapılmaktadır. Orijinal imgelere Gabor Filtre uygulanarak oluşturulan veri seti daha sonra aynı ESA mimarisinden geçmektedir. Sistemin temel adımları Şekil 5.4' de gösterilmiştir.



Şekil 5.4. Sistemin temel adımları

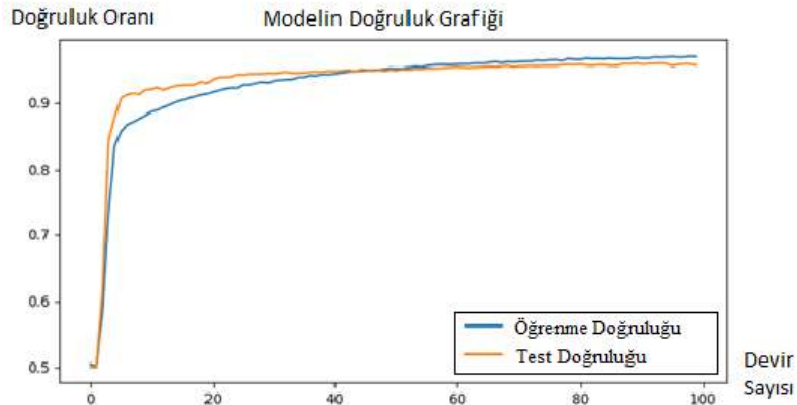
5.2.1. ESA Modeli İle Çatlak Tespiti

ESA modelinin eğitimi için kullanılan resimler veri ön işleme esnasında boyutları 224x224 olarak yeniden boyutlandırılmıştır. İnternet ortamından alınan verisetinin %20'si kullanılarak model oluşturulmuştur [89]. Toplamda veri seti çatlaklar ve normal imgeler içeren 8000 adet görüntüden oluşmaktadır. Veri kümesindeki tüm görüntüler, doğru ölçüm için manuel olarak etiketlenmiştir. Çatlak ve normal olarak etiketlenen veri setindeki bazı resimler Şekil 5.5' de gösterilmektedir. Varolan veri setinin %20'si test verisi %80'i ise öğrenme verisi olarak ayrılmıştır. Test klasörü ve öğrenme klasörü veri setindeki resimlerden rastgele olarak seçilmiştir.

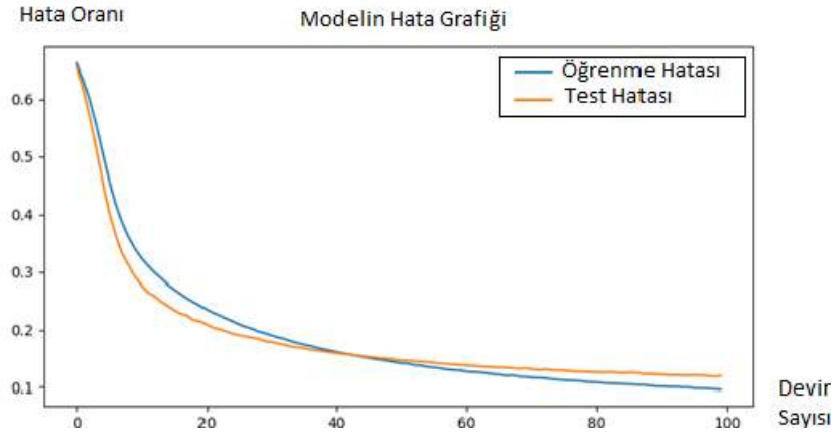


Şekil 5.5. a), b) Yüzey çatlak görüntüleri c), d) Yüzey normal görüntüleri

Bu makalede çatlaklar ve çatlak olmayan görüntüler üzerinden eğitilen modelin test verileri ve öğrenme verileri üzerinde göstermiş olduğu doğruluk oranı Şekil 5.6' da gösterilmiştir. Eğitilen modelin test verileri ve öğrenme verileri üzerinde göstermiş olduğu hata oranı Şekil 5.7' de gösterilmiştir. Model çatlak ve normal görüntülerden oluşan 1600 tane test verisinde ve yine çatlak ve normallerden oluşan 6400 adet öğrenme verisinde %96 başarı göstermiştir. Öte yandan 1600 tane test verisinde ve 6400 adet öğrenme verisinde 1.adımda %69 olan hata oranını modelin eğitimin sonunda %0.1 seviyesine indirmeyi başarmıştır. Model eğitimi 100 adımda gerçekleştirilmiştir.



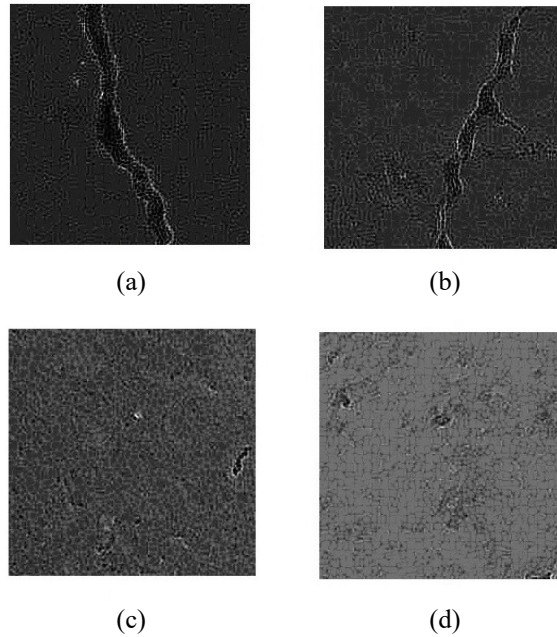
Şekil 5.6. Çatlaklar ve normaller veri setinde ESA Modeli için doğruluk grafiği



Şekil 5.7. Çatlaklar ve normaller veri setinde ESA Modeli için hata grafiği

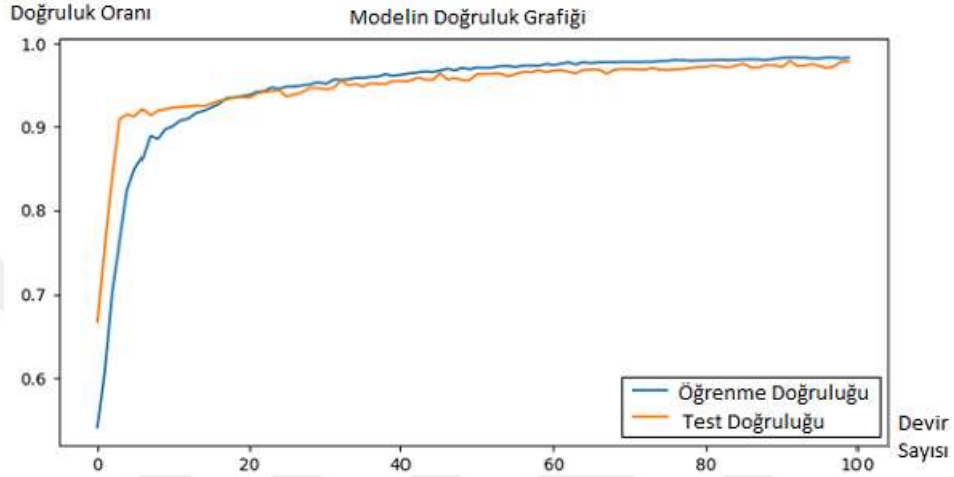
5.2.2. Gabor Filtreli ESA Modeli İle Çatlak Tespiti

Görüntü Sınıflandırma Gabor Filtresi'nin frekansı ve yön gösterimleri, insanların görsel algılama sistemine benzemektedir. Veri setinde bulunan görüntüler farklı Gabor Filtre çekirdeklerinin gerçek kısımları kullanılarak filtrelenir. Filtrelenen imgelerin ortalaması ve varyansı daha sonra basitlik için en küçük kare hatasına dayanan sınıflandırma için öznitelikler olarak kullanılır. Gabor Filtresi vektörel bir filtre olması nedeniyle yüzey çatlakları algılamada modelin kolayca öğrenebileceği resimler oluşturuldu. Veri setinde bulunan öğrenme ve test sınıfı altındaki çatlak ve normal klasörlerindeki resimlerin her birine Gabor Filtre uygulandı. Veri setinde bulunan bazı resimlerin Gabor Filtre çıktısı Şekil 5.8' de gösterilmiştir.

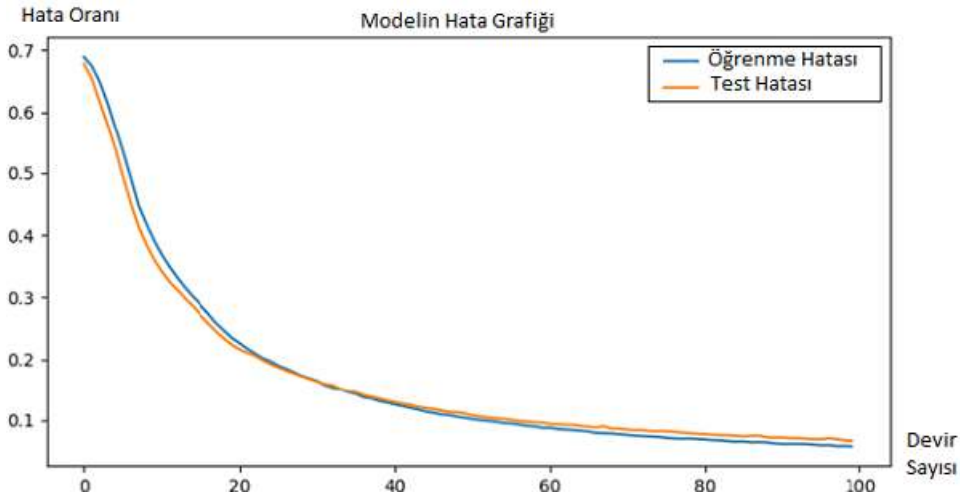


Şekil 5.8. a), b) Yüzey çatlak görüntülerinin Gabor Filtre çıktısı c), d) Yüzey normal görüntülerinin Gabor Filtre çıktısı

Veri setinde bulunan görüntülere 2 boyutlu Gabor Filtre uygulandıktan sonra tekrar aynı ESA modelinden geçirilerek eğitimleri tamamlanmıştır. Model, çatlak ve normal görüntülerden oluşan Gabor Filtre uygulanmış 1600 tane test verisinde ve yine çatlak ve normallerden oluşan Gabor Filtre uygulanmış 6400 adet öğrenme verisinde Şekil 5.9’da gösterildiği gibi %98 başarı göstermiştir. Diğer yandan Şekil 5.10’da hata oranı ESA modeline kıyasla daha da azalarak %0.1’ in altına inmiştir.



Şekil 5.9. Gabor Filtre uygulanmış veri setinin doğruluk grafiği



Şekil 5.10. Gabor Filtre uygulanmış veri setinin hata grafiği

Gabor Filtresi'nin kullanımı, çatlakların algılamasına potansiyel bir alternatif sağlar. Yaklaşımımız, bu filtreyi yüzey çatlakları algılama sorunlarına özel yönelimlerle kullanmaktır. Gabor Filtresi, Gauss zarfı tarafından modüle edilen sinüzoidal bir fonksiyondur. Sinüzoidal fonksiyon, frekansı ve yönü ile karakterize edilir. Bu nedenle, bir Gabor Filtresi, sinüsün yayılma

yönüne dik olan kenarlara tepki verdiğinden, belirli yönelim kenarlarının bir detektörü olarak görülebilir.

Bu çalışmanın performansının ölçülmesi; doğruluk, duyarlılık ve F1-skor ile doğrudan ilişkilidir. F1-skoru hesaplanması için, Doğru Pozitif (DP), Doğru Negatif (DN), Yanlış Pozitif (YP), Yanlış Negatif (YN) değerleri kullanılmaktadır. Performans metriklerinin hesaplanması için kullanılan formüller Doğruluk Denklem 5.1’de, Duyarlılık Denklem 5.2’de F1-Skor ise Denklem 5.3’de verilmiştir.

$$\text{Doğruluk} = ((DP + DN)) / ((DP + YN) + (YP + DN)) \quad (5.1)$$

$$\text{Duyarlılık} = ((DP)) / ((DP + YN)) \quad (5.2)$$

$$\text{F1 - skor} = ((2 \times DP)) / ((2 \times DP + YP + YN)) \quad (5.3)$$

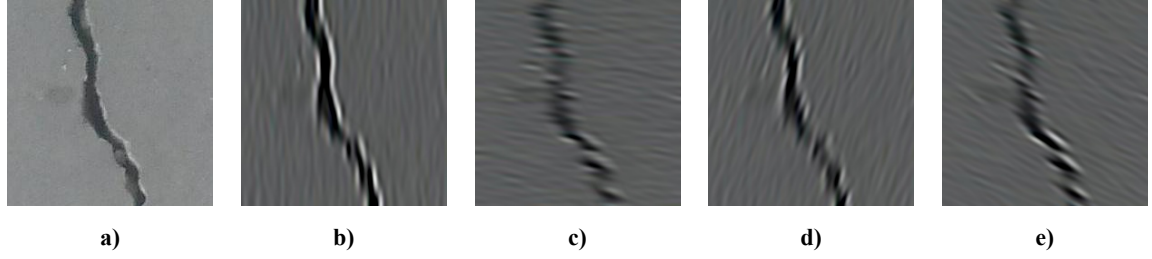
Tablo 5.2. Yalın ESA ve Gabor Filtreli Veri Setlerinin Başarı Ölçütleri Tablosu

	ESA Modeli			Gabor Filtre+ESA Modeli		
	Doğruluk	Duyarlılık	F1-Skor	Doğruluk	Duyarlılık	F1-Skor
Çatlak	0.97	0.94	0.96	0.99	0.97	0.98
Normal	0.94	0.97	0.96	0.97	0.99	0.98
Ağırlıklı Ort.	0.96	0.96	0.96	0.98	0.98	0.98

Bu formüllerden yola çıkarak performans metriklerinde Gabor Filtre çıktısının sıradan ESA modeline kıyasla daha başarılı olduğunu söyleyebiliriz. Hem ESA modelinin hem de Gabor Filtre uygulanmış veri setinin ESA modelindeki başarısı tablosunu Tablo 5.2’ de gösterilmektedir.

5.3. Gabor Filtresi’nde Değişen Sigma ve Teta Değerlerinin Çatlak Tespitine Etkisi

Gabor Filtresi’nin çekirdek parametrelerinde bulunan Teta ve Sigma açısının değerleri değiştirilerek yeni veri tabanları oluşturuldu. Teta açısının 0°, 30° 60° ve 90° veri tabanı [86] ayrı ayrı yeniden oluşturuldu. Teta değerine göre değişen veri tabanı resimlerinin örnekleri Şekil 5.11’ de gösterilmiştir.



Şekil 5.11. a) Normal çatlak görüntüsü, **b)** Teta parametresi 0° iken oluşturulan veri seti **c)** Teta parametresi 30° iken oluşturulan veri seti **d)** Teta parametresi 60° iken oluşturulan veri seti **e)** Teta parametresi 90° iken oluşturulan veri seti

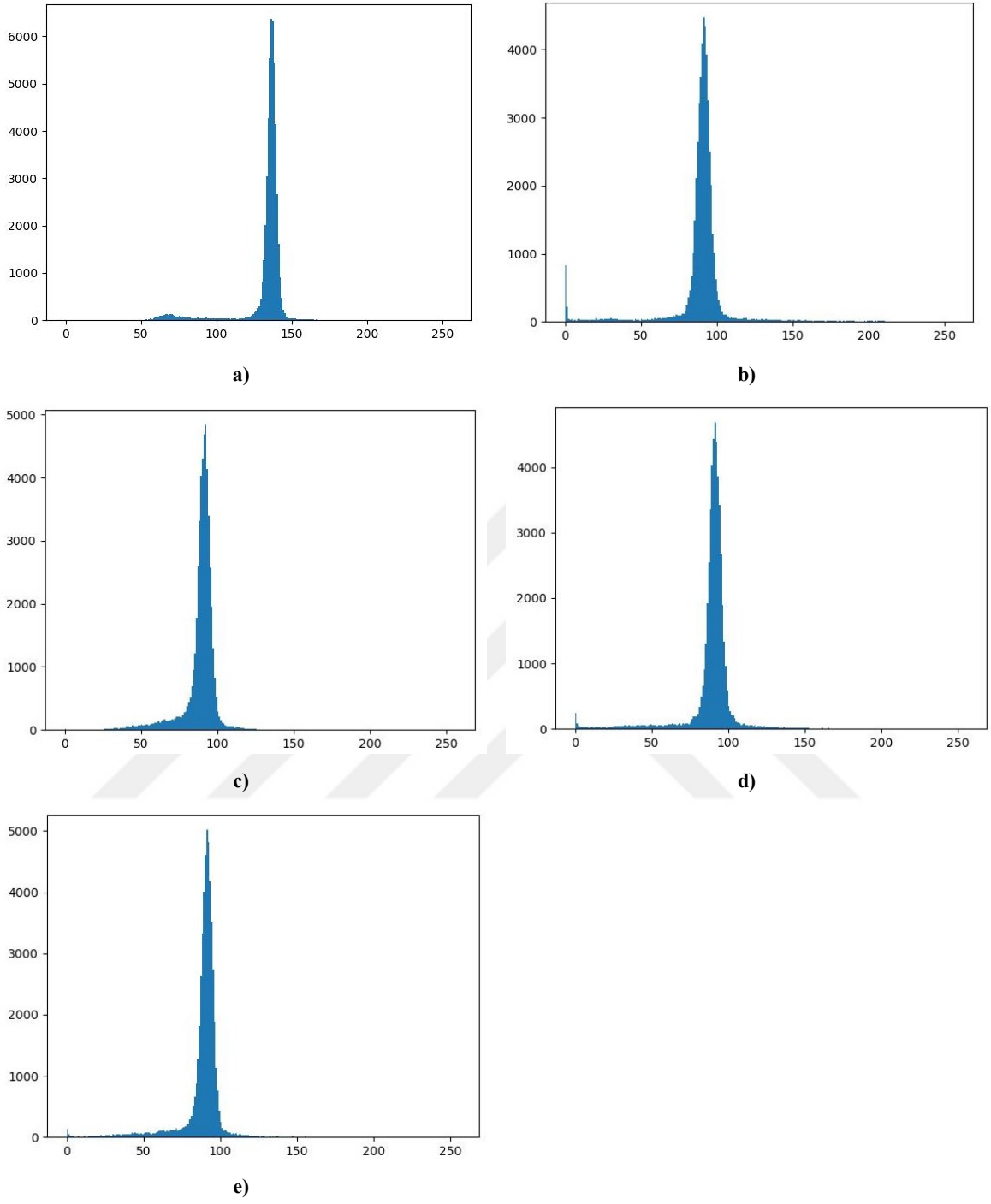
Bir histogram grafiği, verilerin bir çubuk grafik temsilidir. X eksenı boyunca sütun oluşumuna bir dizi sonucun bir temsilidir. Aynı histogramda, her sütun için verilerdeki sayı sayısı veya çoklu oluşumlar y eksenı ile temsil edilir. Veri dağılımlarını görselleştirmek için kullanılabilecek en kolay yoldur. Görüntünün histogramında ise olay buna benzerdir. Öncelikle her bir pikselin resimdeki frekansı bulunur, bu değerler y eksenini oluşturur. Resim formatına göre değişkenlik gösterse de gri renkli resimleri için siyah ile beyaz pikseller 0-255 arası değer almaktadır. Bu değerler ise x ekseninde yer alır. Bu çalışmada yol çatlak görüntüsü ve bu görüntünün Gabor fonksiyonun parametresinde bulunan Teta değerinin değişmesiyle oluşan şekiller Şekil 5.11’de gösterilmişti. Bu resimlerin histogram grafiği Şekil 5.12’de gösterilmektedir.

5.3.1. Gabor Filtre Yönelim Açısı (Teta)

Bu açı, tespit edilmek istenen ayrıtların yönünü belirler. Açı 0° değerinde iken, x eksenine dik olan ayrıtlar tespit edilir. Bunun yerine yatay ayrıtlar tespit edilmek istenirse teta açısı $+90^\circ$ olarak kullanılır. Teta açısı arttırıldıkça, rotasyon matrisinin döndürme yönüne göre artan açılara sahip ayrıtlar tespit edilir.

5.3.2. Gabor Filtre Standart Sapma (Sigma)

Gabor Filtresi’nin çekirdek fonksiyondaki bu değer bir gaussian fonksiyonudur. Band genişliği ve dalga boyu parametreleri kullanılarak hesaplanabilir. Burada bulunan sigma_x ve sigma_y değerleri Gabor Filtresi’nin gauss zarfının eliptik şeklini belirler. Eğer her iki sigma da aynıysa, gauss zarfımız daire şeklinde, aksi takdirde eliptik bir forma sahip olur. Burada hangi değerlerin seçileceğine karar vermek, algılamanın doğruluğunu ve tutarlılığını doğrudan etkiler. Eğer aynı değerler kullanılırsa, frekans uzayında belirli açılarda algılanmayan özellikler ortaya çıkar.



Şekil 5.12. a) Çatlak görüntüsünün histogram grafiği, **b)** Teta parametresi 0 iken oluşan çatlak resminin histogram grafiği, **c)** Teta parametresi 30 iken oluşan çatlak resminin histogram grafiği, **d)** Teta parametresi 60 iken oluşan çatlak resminin histogram grafiği, **e)** Teta parametresi 90 iken oluşan çatlak resminin histogram grafiği

Tablo 5.3. Karmaşıklık matrisi

	TAHMİN EDİLEN VERİ	
	DP (750)	YP (50)
GERÇEK VERİ	YN (16)	DN (784)

Karmaşıklık matrisi bir modelin ya da sınıflandırmanın değerlendirilmesinde önemli rol oynar. Çalışmada çatlak ve normal adında iki adet sınıf bulunmaktadır. Bu nedenle bu ikili sınıflandırma karmaşıklık matrisine iki sütun ve iki satır olarak yansımıştır. Karmaşıklık matrisinde sütunlar tahmin edilen veri sayısını satırlar ise gerçek veri sayısını göstermektedir. Sınıflandırmaların başarı kriterleri incelendiğinde Gabor Filtresi'ndeki parametrelerden Teta=90 Sigma=4 olduğunda ortaya çıkan modelin karmaşıklık matrisindeki değerlerin diğer sonuçlara kıyasla daha başarılı olduğu görülmüştür. Tablo 5.3'de Teta=90 Sigma=4 olduğunda ortaya çıkan karmaşıklık matrisi görülmektedir [87].

- DP = 750: Burada 1600 tane resimden 750 tane çatlak resmi model tarafından doğru bir şekilde çatlak olarak sınıflandırıldı.
- YP = 50: 50 tane normal resim model tarafından yanlış bir şekilde çatlak olarak sınıflandırıldı.
- DN = 784: 784 tane normal resim model tarafından doğru bir şekilde normal olarak sınıflandırıldı.
- YN = 16: 16 tane çatlak resim model tarafından yanlış bir şekilde normal olarak sınıflandırıldı [88].

Tablo 5.4. Başarı ölçütlerinin Teta°'na göre değişimi

Teta°	Doğruluk	Kesinlik	Duyarlılık	F1-Skor	Macro Ort.	Ağırlıklı Ort.
0°	%95.0	%95.5	%95.5	%95.0	%95.0	%95.0
30°	%95.0	%95.0	%91.0	%94.5	%95.0	%95.0
60°	%94.0	%94.5	%94.5	%94.0	%94.0	%94.0
90°	%96.0	%96.0	%96.0	%96.0	%96.0	%96.0

Sigma=4 olarak sabit tutulduğu ve Teta açısının değişen değerlerine göre sınıflandırmanın başarısı Tablo 5.4'te gösterilmektedir.

Tablo 5.5. Başarı ölçütlerinin Sigma'ya göre değişimi

Teta°	Sigma	Doğruluk	Kesinlik	Duyarlılık	F1-Skor	Cohen's Kappa	ROC-AUC
90	0.1	%97.0	%94.7	%99.5	%97.0	%94.0	%97.0
90	0.2	%97.6	%96.2	%99.2	%97.7	%95.3	%97.6
90	0.3	%98.0	%97.5	%99.0	%98.0	%96.0	%98.0
90	0.4	%96.6	%94.4	%99.2	%96.7	%93.3	%96.6
90	0.5	%96.3	%93.6	%99.3	%96.4	%92.6	%96.3
90	0.6	%96.7	%94.5	%99.2	%96.8	%93.5	%96.7
90	0.7	%97.6	%96.2	%99.1	%97.6	%95.2	%97.6
90	0.8	%97.0	%95.1	%99.1	%97.1	%94.1	%97.0
90	0.9	%96.5	%94.0	%99.2	%96.5	%93.0	%96.5
90	1	%95.1	%91.6	%99.3	%95.3	%90.3	%95.1
90	2	%95.7	%92.8	%99.1	%95.8	%91.5	%95.7
90	3	%95.9	%93.5	%98.6	%96.0	%91.8	%95.9
90	4	%96.0	%96.0	%96.0	%96.9	%91.7	%95.8

Buradan yola çıkarak ikinci bir çalışma olarak doğruluk kesinlik gibi diğer başarı parametrelerini daha da iyileştirmek için Teta'nın en başarılı olduğu açı olan 90° kullanılmalıdır. Başarı değerlerini artırmak için Sigma'yı 0'a yaklaşması gerekmektedir. Sigma_x ve Sigma_y'den oluşan ve gaussian zarfının eliptik şeklini belirleyen bu fonksiyon için yeni veri setleri oluşturulmuştur. Sigma değeri 4 - 1 ve 0,9 - 0,1 aralığında ki değerlerle yapılan testlerden elde edilen sonuçlar Tablo 5.5'te gösterilmiştir. Kullanılan sütun başlıkları sınıflandırmanın başarısını belirlemektedir.

Doğruluk: Modelde doğru tahmin edilen verilerin toplam veri kümesine oranı ile hesaplanmaktadır.

Kesinlik: Pozitif olarak tahmin edilen verilerin gerçekte kaç tanesinin pozitif olduğunu göstermektedir.

Duyarlılık: Pozitif olarak tahmin etmemiz gereken işlemlerin kaç tanesini pozitif olarak tahmin ettiğimizi gösterir.

F1-Skor: Duyarlılık ve Kesinlik değerlerinin harmonik ortalamasıdır [89].

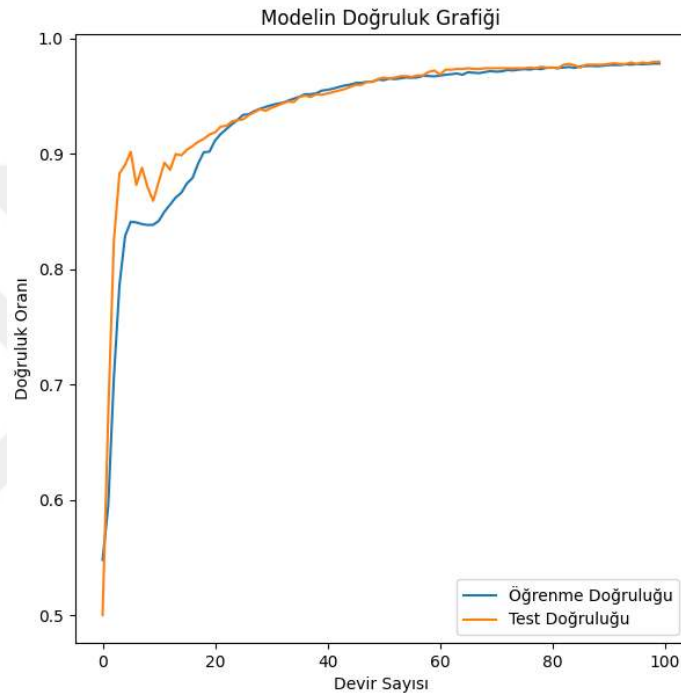
ROC-AUC Eğrisi: İşlem karakteristiği eğrisi olup, bu eğrinin altında kalan alanın büyüklüğü değerlendirme için önemlidir. Bu eğri MÖ modellerinin performansını görsel anlamda değerlendirmede kullanışlı bir yöntemdir. İşlem karakteristiği eğrisi bir olasılık eğrisidir. Teta'nın 90 değerini Sigma'nın ise 0.3 değerini aldığı son çalışmada ROC-AUC eğrisinin değeri %98,0 olarak hesaplanmıştır. Çatlak ve Normal etiketlerine sahip sınıflarımız ROC eğrisinde yüksek yer kapladığından model başarısının yüksek olduğunu söyleyebiliriz.

Cohen's Kappa: MÖ modellerinde hem çok sınıflı hem de dengesiz sınıf problemlerinin çok iyi çözümleyen bir istatistiktir. P0'ın gözlemlenen durumu, Pe'nin ise beklenen durumu gösterdiğinde Kappa $P0-Pe/1-Pe$ olarak hesaplanır [90].

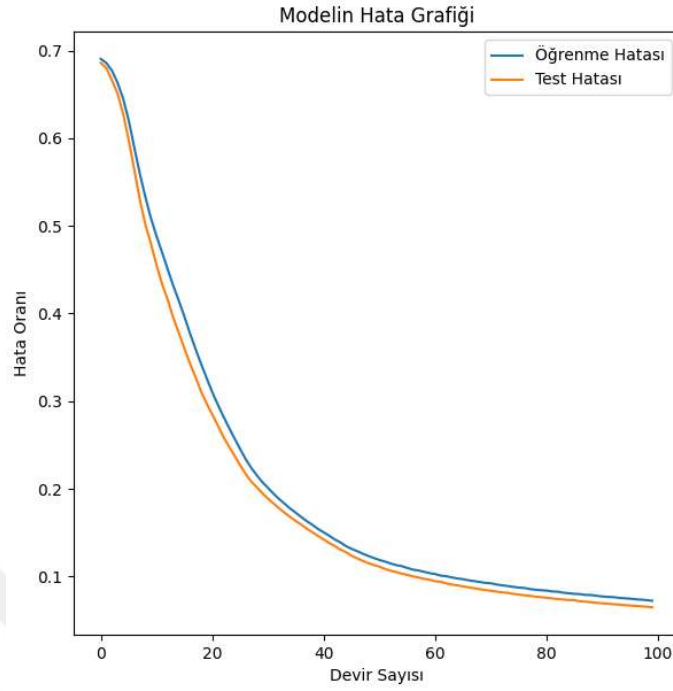
Makro Ortalama: Genellikle sınıf sayısı ikiden fazla olunca daha çok tercih edilen bir ölçüttür. Tüm sınıfların duyarlılıklarının aritmetik ortalaması alınarak bulunmaktadır.

Ağırlıklı Ortalama: Farklı sınıf etiketleriyle ilgili farklı sayıda örnek olduğu durumlarda ağırlıklı ortalama kullanılır. Elde edilen değerleri tüm sınıf oranlarıyla çarparak duyarlılıklarının toplanmasıdır.

Yapılan çalışmalar sonucunda Teta değerinin 90, Sigma değerinin ise 0.3 olduğu durumda tüm değerler optimum düzeydedir. Bu şekilde oluşturulan modelin test verileri üzerinde göstermiş olduğu doğruluğun grafiği Şekil 5.13’de gösterilmiştir. Diğer yandan hata değeri de diğer çalışmalara kıyasla en düşük değerini almıştır. Modelin hata grafiği Şekil 5.14’te gösterilmiştir.



Şekil 5.13. En iyi başarı gösteren modelin doğruluk grafiği

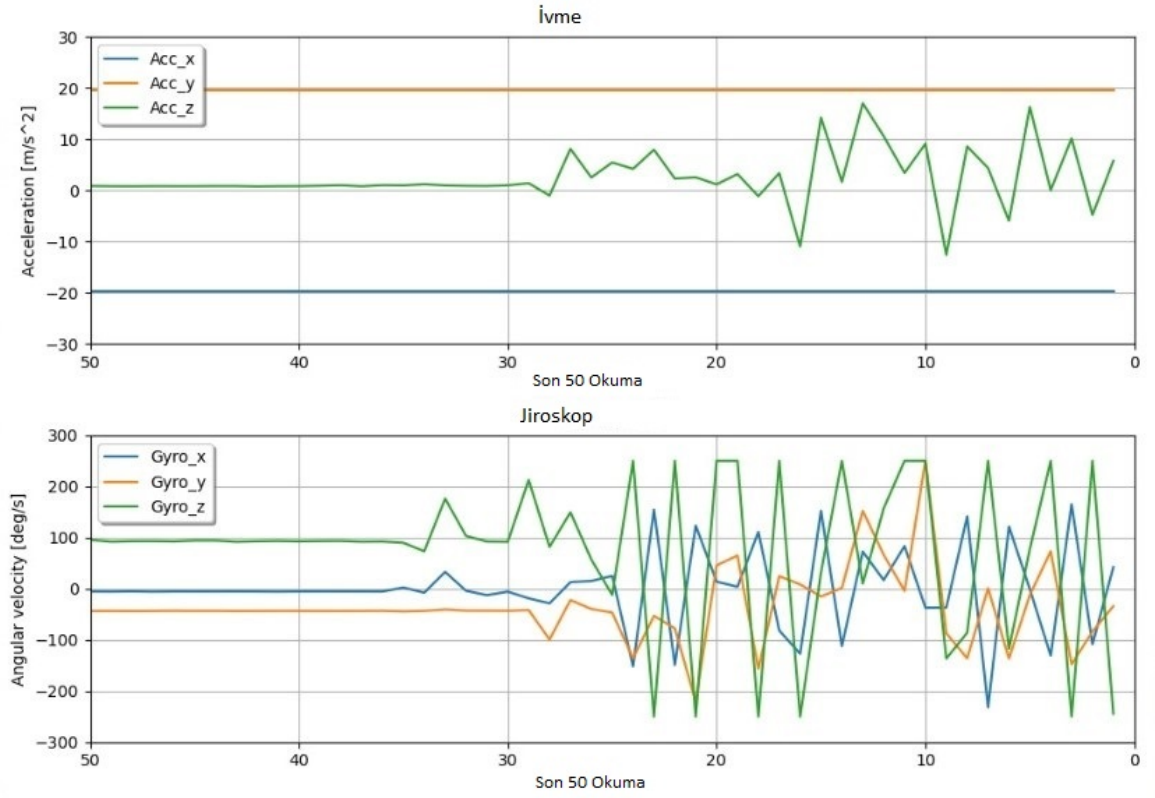


Şekil 5.14. En düşük hata payına sahip modelin grafiği

5.4. 6 Eksenli İvme Sensörü

İvmeölçerler, dijital cihazlar ve analog cihazlar olarak mevcuttur. İvmeölçerler farklı yöntemler kullanılarak tasarlanmıştır. Piezoelektrik, piezodirençli ve kapasitif bileşenler genellikle ivmeölçerlerde oluşan mekanik hareketi elektrik sinyaline dönüştürmek için kullanılır. Piezoelektrik ivmeölçerler tek kristallerden oluşur. Bunlar ivmeyi ölçmek için piezoelektrik etkiyi kullanır. Strese uygulandığında, bu kristaller hız ve yönelimi belirlemek için yorumlanan bir voltaj üretir. Kapasitif ivmeölçerler, silikon mikro işlenmiş bir eleman kullanır. Burada hızlanma algılandığında kapasitans üretilir ve bu kapasitans hız değerlerini ölçmek için bir voltaja çevrilir. Oluşturulan sistemde MPU6050 6 eksenli ivme sensörünü Rapberry Pi Geliştirme Modülü üzerine entegre ederek kullanılmaktadır. Raspberry Pi Geliştirme Modülü üzerinde dâhili olarak bulunan Esp8266 Wifi modülü ile IOT, Smart City ve Drone Sistemleri gibi birçok sistemde de tercih edilmektedir. Klasik Arduino UNO R3'ün aksine Python gibi üst düzey dillerle programlanabilir.

Tablo 5.6'da MPU6050 6 eksenli ivme sensörü yardımıyla bir karayolunda seyir halinde giden bir aracın Şekil 5.16'da görülen çatlak üzerinden geçerken okunan Gx, Gy, Gz değerleri gösterilmektedir. Bu değerlerin anlık grafiği Şekil 5.15'de sunulmuştur.



Şekil 5.15. İvme Sensörü'nden gelen değerlerin anlık grafiği



Şekil 6.16. Yol çatlak görüntüsü

Tablo 5.6. İlk 20 adım için İvme Sensör değerleri

	Gx	Gy	Gz
1	-5.15	-42.07	86.56
2	-5.23	-41.98	88.47
3	-5.29	-42.39	88.94
4	-5.24	-42.20	87.33
5	-5.15	-41.81	88.38
6	-35.27	-42.35	100.55
7	-26.04	-42.92	99.02
8	-31.76	-41.68	99.73
9	-52.63	-42.58	109.21
10	-46.15	-42.40	103.94
11	-36.47	-42.49	104.31
12	-25.22	-41.85	99.37
13	-30.97	-41.95	99.85
14	-26.39	-40.49	96.05
15	-24.09	-41.74	96.08
16	-17.09	-41.47	96.44
17	-5.19	-42.16	88.72
18	-4.01	-43.38	87.92
19	-1.89	-42.18	87.05
20	-4.54	-41.90	87.03

6. SONUÇLAR

Yapılan tez çalışmasında beton ve yol görüntülerinden çatlak tespitini insan muhakemesi olmadan gerçekleştiren yöntemler uygulanmıştır. Önceki çalışmalardan elde edilen sonuçlar ile çalışmamızdan elde edilen sonuçlar karşılaştırılmış ve çalışmamızın başarısı ortaya çıkmıştır.

İlk deneyde yeni bir çatlak tespit yaklaşımı önerilmiştir. Veri setinde bulunan imgeler öncelikle gri forma daha sonra 224x224 boyutuna dönüştürüldü. Ön işlemden geçen görüntüler daha sonra eğitim ve test adında iki gruba %20 test %80 eğitim olacak biçimde gruplandırıldı, üzerinde çatlak bulunan ve üzerinde çatlak bulunmayan imgeler her iki gruba da rastgele dağıtıldı. Bu sayede, veri ön işleme ve ayırma işlemleri tamamlandı ve sınıflandırılmaya hazır duruma getirildi. Veri setine 1 Bloklü VGG-16 mimarisi uygulandı ve doğruluk değeri %83,988 olarak kaydedildi. Bu işlemde elde edilen başarının yeterli olmamasından dolayı sırasıyla, 2 Bloklü VGG-16 ve 3 Bloklü VGG-16 modelleri kullanılarak %93,895 ve %95,930 değerleri elde edildi. Bu modellerden elde edilen sonuçların grafikler yardımıyla gösterilmesi sağlandı. Doğruluğun yanı sıra hata payının da %0 'a yaklaştığı gözlemlendi. Oluşturulan modelin tahmini için oluşturulan predict() metodu kullanılarak seçtiğimiz resmin çatlak barındırıp barındırmadığını tahmin etmesi sağlandı. Eğitilen model tahmini başarıyla gerçekleştirdi ve doğru sınıfı seçti.

İkinci deneyde, betonarme yapılarda yüzey çatlaklarının tespiti için Gabor Filtrelemesi kullanan ESA mimarili bir tespit yöntemi önerilmiştir. Çalışma yüzey çatlaklarının bulunduğu tüm veri setinin %20 'sine uygulanmıştır. Öncelikli olarak veri seti test ve öğrenme olarak gruplandırıldı. Daha sonra elde edilen veri seti ön işleme aşamalarından geçerek yeniden boyutlandırıldı ve manuel olarak düzenlendi. ESA modeli eğitimi sonrasında modelimiz %96 başarı gösterdi. Bu eğitim esnasında elde edilen başarı ve hata grafikleri gösterildi. İkinci aşama olarak veri setine Gabor Filtre uygulandı. Daha önceleri bölütleme, kenar algılama, nesne tanıma gibi birçok algorithmada kullanılan yön tabanlı Gabor Filtresi'nin, yüzey çatlağı imgelerinin üzerindeki etkisi gözlemlendi. Gabor Filtresi kullanılması durumunda %98 oranında başarı elde edilmiştir. Önerilen Gabor tabanlı çatlak tespit sistemi, literatürde verilen çatlak tespit tekniklerine göre daha başarılı bir alternatif yöntem olduğu görülmüştür.

Üçüncü deneyde Gabor Filtresinin çekirdeğinde bulunan parametrelerin değişmesinin çatlak tespitine etkisi üzerinde durulmuştur. Modelde kullanılacak veri seti ilk olarak Teta değerlerine göre oluşturuldu. Bu kısımda diğer parametre değerleri sabit tutuldu ve sistemin başarısı gözlemlendi. Teta 0°, 30°, 60° ve 90° değerlerine göre oluşturulan veri tabanları tek tek modelden geçirildi ve en iyi sonuç Teta 90° iken olduğu görüldü. Teta'nın optimize değerinin bulunması aşamasında elde edilen sonuçlar grafikler ve tablolar ile gösterildi. Daha sonra diğer bir parametre olan Sigma parametresi ile Teta açısı 90° de sabitlenerek en iyi başarı gösteren

Sigma deęerinin bulunması amaçlandı. Buradan Sigma deęeri başarı deęerlerinin arttıęı gözlemlendi. Sigma deęerinin 1 ‘den başlanarak ve 0,1 azalarak 0,1 ‘e kadar olan tüm deęerleri için ayrı ayrı veri setleri oluşturuldu ve eğitilen model uygulandı. Modelin başarısını karşılaştırmak için karmaşıklık matrisleri ve tablolar ve grafikler ile sonuçlar gösterildi. Bu sayede Sigma deęeri 0,3 olduğunda Doğruluk %98, Duyarlılık %99, Kesinlik %97,5, Cohen’s Kappa %96, F1-skor %98 ve ROC-AUC deęerinin ise %98 olduğü görüldü. Hata oranı ise Sigma deęeri 0,3 olduğunda 0,1’in altına düşmüştür.

Dördüncü çalışmada gerçek yol koşullarından alınan veriler ile ölçümler gerçekleştirildi. MPU-6050 ivme sensörünün araca yerleştirilmesi ile yolun durumu hakkında bilgi elde edildi. Bu sensör çıktı olarak x yönünde ivme A_x , y yönünde ivme A_y , z yönünde ivme A_z ve x yönünde hareket G_x , y yönünde hareket G_y , z yönünde hareket G_z vermektedir. Yolda bulunan çukur çatlak gibi anormalinin şiddetini ölçmek ve anlık olarak gözlemleyebilmek için gerçek zamanlı çalışan sistem oluşturuldu.

Gelecek çalışmalarda doku analizi, karakter tanıma gibi birçok uygulamada yetkinlięi daha önceden test edilmiş olan Gabor Filtresi’nin uygulandığı veri seti genişletilerek daha fazla resim verisi ile derin bir sinir ağı kullanılarak eğitilebilir, ortaya koyduğumuz parametreleri kullanarak gerçek zamanlı çalışmalar yapılabilir. Bu yöntemler ile daha sonra ön işleme adımları ve sınıflandırma mimarisi geliştirilerek başarı oranının artırılabilceęi öngörülmektedir.

KAYNAKLAR

- [1] H. Cho, H.-J. Yoon, J.-Y. Jung, 2018. Image-based crack detection using crack width transform (CWT) algorithm. IEEE Access, 6(1), 60100–60114. <https://doi.org/10.1109/ACCESS.2018.2875889>
- [2] T. C. Su, M. D. Yang, Morphological segmentation based on edge detection-II for automatic concrete crack measurement. Computers and Concrete, 21(6), (2018), 727–739. <https://doi.org/10.12989/cac.2018.21.6.727>
- [3] A. Pikaz, A. Averbuch, “An efficient topological characterization of gray-levels textures using a multiresolution representation,” Graphical Models and Image Processing, 59, (1997), pp. 1-17.
- [4] I. Abdel-Qader, Analysis of edge-detection techniques for crack identification in bridges. Journal of Computing in Civil Engineering, 17(4), (2003), 255 – 263. [https://doi.org/10.1061/\(ASCE\)0887-3801\(2003\)17:4\(255\)](https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(255))
- [5] L.-C. Chen , Y.-C. Chao, H.-H. Jan, C.-W. Huang, Y.-M. Tien, Measuring System for Cracks in Concrete Using Multitemporal Images. Journal of Surveying Engineering, 132, (May 2006), 77 – 82. [http://doi.org/10.1061/\(ASCE\)0733-9453\(2006\)132:2\(77\)](http://doi.org/10.1061/(ASCE)0733-9453(2006)132:2(77))
- [6] T. C. Hutchinson, M. Asce, Z. Chen, Improved Image Analysis for Evaluating Concrete Damage, 20 (June 2006), 210 – 216. [https://doi.org/10.1061/\(ASCE\)0887-3801\(2006\)20:3\(210\)](https://doi.org/10.1061/(ASCE)0887-3801(2006)20:3(210))
- [7] T. Yamaguchi, S. Nakamura, R. Saegusa, S. Hashimoto, Image-based crack detection for real concrete surfaces, IEEJ Trans. Electr. Electron. Eng. 3 (2008) 128–135, <https://doi.org/10.1002/tee.20244>.
- [8] N. Gehri, J. Mata-Falc'ón, W. Kaufmann, Automated crack detection and measurement based on digital image correlation, Constr. Build. Mater. 256 (2020) 119383, <https://doi.org/10.1016/j.conbuildmat.2020.119383>.
- [9] R.S. Adhikari, O. Moselhi, A. Bagchi, Image-based retrieval of concrete crack properties for bridge inspection, Autom. Constr. 39 (2014) 180–194, <https://doi.org/10.1016/j.autcon.2013.06.011>.
- [10] B. Xu, Y. Huang, Automatic inspection of pavement cracking distress, Appl. Digital Image Proc. XXVIII, (2005), 5909-590901, <https://doi.org/10.1117/12.613770>.
- [11] Y.-C. Tsai, V. Kaul, R.M. Mersereau, Critical assessment of pavement distress segmentation methods, J. Transp. Eng. 136 (2009) 11–19, [https://doi.org/10.1061/\(asce\)te.1943-5436.0000051](https://doi.org/10.1061/(asce)te.1943-5436.0000051).
- [12] Q. Zou, Y. Cao, Q. Li, Q. Mao, S. Wang, CrackTree: Automatic crack detection from pavement images, Pattern Recogn. Lett. 33 (2012) 227–238, <https://doi.org/10.1016/j.patrec.2011.11.004>.
- [13] A. Arena, C. Delle Piane, J. Sarout, A new computational approach to cracks quantification from 2D image analysis: Application to micro-cracks description in rocks, Comput. Geosci. 66 (2014) 106–120, <https://doi.org/10.1016/j.cageo.2014.01.007>.
- [14] H.N. Koutsopoulos, A.B. Downey, Primitive-based classification of pavement cracking images, J. Transp. Eng. 119 (2006) 402–418, [https://doi.org/10.1061/\(asce\)0733-947x\(1993\)119:3\(402\)](https://doi.org/10.1061/(asce)0733-947x(1993)119:3(402)).
- [15] A. Cubero-Fernandez, F.J. Rodriguez-Lozano, R. Villatoro, J. Olivares, J. M. Palomares, Efficient pavement crack detection and classification, EURASIP J. Image Video Proc. 39 (2017) 1–11, <https://doi.org/10.1186/s13640-017-0187-0>.
- [16] S.K. Sinha, P.W. Fieguth, Automated detection of cracks in buried concrete pipe images, Autom. Constr. 15 (2006) 58–72, <https://doi.org/10.1016/j.autcon.2005.02.006>.
- [17] C. Koch, S.G. Paal, A. Rashidi, Z. Zhu, M. K'önig, I. Brilakis, Achievements and challenges in machine vision-based inspection of large concrete structures, Adv. Struct. Eng. 17 (2014) 303–318, <https://doi.org/10.1260/1369-4332.17.3.303>.

- [18] J.K. Oh, G. Jang, S. Oh, J.H. Lee, B.J. Yi, Y.S. Moon, J.S. Lee, Y. Choi, Bridge inspection robot system with machine vision, *Autom. Constr.* 18 (2009) 929–941, <https://doi.org/10.1016/j.autcon.2009.04.003>.
- [19] P. Prasanna, K. Dana, N. Gucunski, B. Basily, Computer-vision based crack detection and analysis, in: *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, International Society for Optics and Photonics 8345, (2012), p. 834542, <https://doi.org/10.1117/12.915384>.
- [20] L. Li, L. Sun, G. Ning, S. Tan, Automatic pavement crack recognition based on BP neural network, *PROMET - Traffic Transp.* 26 (2014) 11–22, <https://doi.org/10.7307/ptt.v26i1.1477>.
- [21] M. Salman, S. Mathavan, K. Kamal, M. Rahman, Pavement crack detection using the Gabor filter, in: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, IEEE*, 2013, pp. 2039–2044, <https://doi.org/10.1109/ITSC.2013.6728529>.
- [22] A.M.A. Talab, Z. Huang, F. Xi, L. Haiming, Detection crack in image using Otsu method and multiple filtering in image processing techniques, *Optik.* 127 (2016) 1030–1033, <https://doi.org/10.1016/j.ijleo.2015.09.147>.
- [23] H. Oliveira, P.L. Correia, CrackIT - An image processing toolbox for crack detection and characterization, *IEEE International Conference on Image Processing 2014, ICIP*, (2014), pp. 798–802, <https://doi.org/10.1109/ICIP.2014.7025160>.
- [24] D. H. Hubel, T. N. Wiesel. “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex”. In: *The Journal of physiology* 160.1 (1962), pp. 106–154. <https://doi.org/10.1113/jphysiol.1962.sp006837>
- [25] D. Barina, “Gabor wavelets in image processing”. In: *arXiv preprint arXiv:1602.03308* (2016) <https://doi.org/10.48550/arXiv.1602.03308>.
- [26] T. P. Weldon, W. E. Higgins, D. F Dunn. “Efficient Gabor filter design for texture segmentation”. In: *Pattern recognition* 29.12 (1996), pp. 2005–2015. [https://doi.org/10.1016/S0031-3203\(96\)00047-7](https://doi.org/10.1016/S0031-3203(96)00047-7)
- [27] R. Mehrotra, K. R. Namuduri, N. Ranganathan, “Gabor filter-based edge detection”. In: *Pattern recognition* 25.12 (1992), pp. 1479–1494. [https://doi.org/10.1016/0031-3203\(92\)90121-X](https://doi.org/10.1016/0031-3203(92)90121-X)
- [28] T. S. Lee. “Image representation using 2D Gabor wavelets”. In: *IEEE Transactions on pattern analysis and machine intelligence* 18.10 (1996), pp. 959–971. <https://doi.org/10.1109/34.541406>.
- [29] V. Krüger, G. Sommer. “Gabor wavelet networks for object representation”. In: *Multi-Image Analysis*. Springer, 2001, pp. 115–128. https://doi.org/10.1007/3-540-45134-X_9
- [30] H. Zhang, B. Zhang, W. Huang, Q. Tian “Gabor wavelet associative memory for face recognition”. In: *IEEE transactions on neural networks* 16.1 (2005), pp. 275–278. <https://doi.org/10.1109/TNN.2004.841811>
- [31] A. Kaushal, J. Raina, “Face detection using neural network & Gabor wavelet transform”, In: *IJCST* 1.1 (2010), <https://doi.org/10.1109/SOCPAR.2014.7007975>
- [32] S. Choubey, D. Banchhor. “A Literature Survey of various Fingerprint De-noising Techniques to justify the need of a new De-noising model based upon Pixel Component Analysis”, (2015) In: *arXiv preprint arXiv:1512.00939*.
- [33] L. Zhang, F. Yang, Y. Daniel Zhang, Y.J. Zhu, Road crack detection using deep convolutional neural network, in: *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3708–3712, <https://doi.org/10.1109/ICIP.2016.7533052>.
- [34] L. Pauly, H. Peel, S. Luo, D. Hogg, R. Fuentes, Deeper networks for pavement crack detection, in: *34th International Symposium on Automation and Robotics in Construction*, 2017, pp. 479–485, <https://doi.org/10.22260/ISARC2017/0066>.

- [35] Y.J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, *Comp. Aided Civil Infrastr. Eng.* 32, (2017) 361–378, <https://doi.org/10.1111/mice.12263>.
- [36] S. Yokoyama, T. Matsumoto, Development of an automatic detector of cracks in concrete using machine learning, *Proc. Eng.* 171 (2017) 1250–1255, <https://doi.org/10.1016/j.proeng.2017.01.418>.
- [37] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, S. Wang, DeepCrack: learning hierarchical convolutional features for crack detection, *IEEE Trans. Image Process.* 28 (2018) 1498–1512, <https://doi.org/10.1109/TIP.2018.2878966>.
- [38] Y. Cha, W. Choi, Vision-based concrete crack detection using a convolutional neural network, in: *Dynamics of Civil Structures 2*, Springer, Cham, (2017), pp. 71–73, https://doi.org/10.1007/978-3-319-54777-0_9.
- [39] K. Chaiyasarn, M. Sharma, L. Ali, W. Khan, N. Poovarodom, Crack detection in historical structures based on convolutional neural network, in: *International Journal of Geomate* 15, 2018, pp. 240–251, <https://doi.org/10.21660/2018.51.35376>.
- [40] M. Wang, J.C.P. Cheng, Development and improvement of deep learning based automated defect detection for sewer pipe inspection using faster R-CNN, in: *Advanced Computing Strategies for Engineering*, Springer, Cham, 2018, pp. 171–192, https://doi.org/10.1007/978-3-319-91638-5_9.
- [41] F. Fang, L. Li, Y. Gu, H. Zhu, J.-H. Lim, A novel hybrid approach for crack detection, *Pattern Recogn.* 107 (2020) 107474, <https://doi.org/10.1016/j.patcog.2020.107474>.
- [42] E. Ibragimov, H.J. Lee, J.J. Lee, N. Kim, Automated pavement distress detection using region based convolutional neural networks, in: *International Journal of Pavement Engineering*, 2020, pp. 1–12, <https://doi.org/10.1080/10298436.2020.1833204>.
- [43] V. Nair, G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines”. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.
- [44] Yann LeCun and Corinna Cortes. “MNIST handwritten digit database”. In: (2010). URL: <http://yann.lecun.com/exdb/mnist/>.
- [45] M. I. Jordan, T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*. (2015), <https://doi.org/10.1126/science.aaa8415>.
- [46] E. Alpaydin, “Introduction to Machine Learning Ethem Alpaydin” *Introduction to Machine Learning*, Third Edition, 2014.
- [47] G. Shobha, S. Rangaswamy, “Chapter 8 - Machine Learning,” in *Handbook of Statistics*, vol. 38, V. N. Gudivada and C. R. Rao, Eds. Elsevier, 2018, pp. 197–228.
- [48] S. Purushotham, B. K. Tripathy, “Evaluation of classifier models using stratified tenfold cross validation techniques,” in *Communications in Computer and Information Science*, 2012, https://doi.org/10.1007/978-3-642-29216-3_74.
- [49] P. Geurts, L. Wehenkel, “Investigation and reduction of discretization variance in decision tree induction” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1810, 2000, pp. 162–170.
- [50] A. V. Miceli Barone, B. Haddow, U. Germann, and R. Sennrich, “Regularization techniques for fine-tuning in neural machine translation,” in *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2017, <https://doi.org/10.18653/v1/d17-1156>.
- [51] J. Lever, M. Krzywinski, N. Altman, “Classification evaluation,” *Nature Methods*, 2016, <https://doi.org/10.1038/nmeth.3945>.
- [52] S. Ren, K. He, R. Girshick, J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems*, 2015, pp. 91–99 <https://doi.org/10.48550/arXiv.1506.01497>

- [53] J. Redmon, A. Farhadi. "YOLO9000: better, faster, stronger". In: Proceedings of the IEEE conference on computer vision and pattern recognition, (2017), pp. 7263–7271. <https://doi.org/10.48550/arXiv.1612.08242>.
- [54] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A. C. Berg "Ssd: Single shot multibox detector". In: European conference on computer vision. Springer, (2016), pp. 21–37. <https://doi.org/10.48550/arXiv.1512.02325>.
- [55] J. Redmon, A. Farhadi. "Yolov3: An incremental improvement". In: arXiv preprint (2018). <https://doi.org/10.48550/arXiv.1804.02767>.
- [56] J. Redmon, S. Divvala, R. Girshick, A. Farhadi "You only look once: Unified, real-time object detection". In: Proceedings of the IEEE conference on computer vision and pattern recognition, (2016), pp. 779–788. <https://doi.org/10.48550/arXiv.1506.02640>.
- [57] Yiğit Mesci. Yolo Mimarisi. <https://medium.com/deep-learning-turkiye/yolo-algoritmas%C4%B1n%C4%B1-anlamak-290f2152808f>. (Erişim Tarihi: 23.04.2022).
- [58] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar "Focal loss for dense object detection". In: Proceedings of the IEEE international conference on computer vision, (2017), pp. 2980–2988. <https://doi.org/10.48550/arXiv.1708.02002>.
- [59] R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation". In: Proceedings of the IEEE conference on computer vision and pattern recognition, (2014), pp. 580–587 <https://doi.org/10.48550/arXiv.1311.2524>.
- [60] K. He, X. Zhang, S. Ren, J. Sun "Spatial pyramid pooling in deep convolutional networks for visual recognition". In: IEEE transactions on pattern analysis and machine intelligence 37.9 (2015), pp. 1904–1916. <https://doi.org/10.48550/arXiv.1406.4729>.
- [61] R. Girshick. "Fast r-cnn". In: Proceedings of the IEEE international conference on computer vision. (2015), pp. 1440–1448. <https://doi.org/10.48550/arXiv.1504.08083>.
- [62] Mark Everingham, J. Winn "The PASCAL visual object classes challenge 2007 (VOC2007) results". In: (2007) <https://doi.org/10.1007/s11263-009-0275-4>.
- [63] Y. Quan, J. Sun, Y. Zhang, H. Zhang, "The method of the road surface crack detection by the improved otsu threshold," in Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA), Aug. (2019), pp. 1615–1620. <https://doi.org/10.1109/ICMA.2019.8816422>.
- [64] H. Nhat-Duc, Q.-L. Nguyen, V.-D. Tran, "Automatic recognition of asphalt pavement cracks using Metaheuristic optimized edge detection algorithms and convolution neural network," Autom. Construct., vol. 94, Oct. (2018), pp. 203–213, <https://doi.org/10.1016/j.autcon.2018.07.008>
- [65] R. Amhaz, S. Chambon, J. Idier, V. Baltazart, "Automatic crack detection on 2D pavement images : An algorithm based on minimal path selection," IEEE Trans. Intell. Transp. Syst., vol. 17, no. 10, Oct. (2016), pp. 2718–2729, <https://doi.org/10.1109/TITS.2015.2477675>.
- [66] F. Ç. Özgenel, Concrete Crack Images for Classification. Mendeley Data (2018), <https://doi.org/10.17632/5y9wdsg2zt.1>.
- [67] M. Maguire, S. Dorafshan, R. J. Thomas, SDNET2018: A concrete crack image dataset for machine learning applications. Utah State University, (2018), <https://doi.org/10.15142/T3TD19>.
- [68] B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubard, L. D. Jackel, Y. LeCun, in Backpropagation Applied to Handwritten Zip Code Recognition, MIT, (1989), pp. 541–551, <https://doi.org/10.1162/neco.1989.1.4.541>.
- [69] F. Chollet, Keras, <http://keras.io>, (Erişim Tarihi: 14.04.2022).
- [70] F. Ç. Özgenel, A. S. Gönenç, "Performance Comparison of Pretrained Convolutional Neural Networks on Crack Detection in Buildings," ISARC, Berlin, (2018), <https://doi.org/10.22260/ISARC2018/0094>.
- [71] OpenCV team. Opencv. <https://opencv.org/about/>. (Erişim Tarihi: 21.04.2022).

- [72] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, Published as a conference paper at ICLR, (2015) <https://doi.org/10.48550/arXiv.1409.1556>.
- [73] Pawangfg. 2020. VGG-16 CNN Model. <https://www.geeksforgeeks.org/vgg-16-cnn-model>. (Erişim Tarihi: 21.05.2021).
- [74] D. J. Gabor, Theory of communication. IEEE, 93(26), (1946), 429–457.
- [75] P. P. Raghu, B. Yegnanarayana, Segmentation of Gabor-filtered textures using deterministic relaxation. IEEE Trans. Image Processing, 5(12), (1996), 1625-1636. <https://doi.org/10.1109/83.544570>.
- [76] M. Clark, A. C. Bovik, Texture segmentation using Gabor modulation/ demodulation. Pattern Recognition Letters 6(4), (1987), 261-267. [https://doi.org/10.1016/0167-8655\(87\)90086-9](https://doi.org/10.1016/0167-8655(87)90086-9).
- [77] K. R. Namuduri, R. Mehrotra, N. Ranganathan, Efficient computation of Gabor filter based multiresolution responses. Pattern Recognition, 27(7), (1994), 925-938. [https://doi.org/10.1016/0031-3203\(94\)90158-9](https://doi.org/10.1016/0031-3203(94)90158-9).
- [78] D. P. Casasent, J. S. Smokelin, A. Ye, Wavelet and Gabor transforms for detection. Optical Engineering 31(9), (1992), 1893-1898. <https://doi.org/10.1117/12.59913>.
- [79] A. K. Jain, N. K. Ratha, S. Lakshmanan, Object detection using Gabor filters. Pattern Recognition, 30(2), (1997), 295-309. [https://doi.org/10.1016/S0031-3203\(96\)00068-4](https://doi.org/10.1016/S0031-3203(96)00068-4).
- [80] M. Porat, Y. Y. Zeevi, The generalized Gabor scheme of image representation in biological and machine vision. IEEE Trans. Pattern Analysis Mach. Intelligence, 10(4), (1988), 452-468. <https://doi.org/10.1109/34.3910>.
- [81] Y. Hamamoto, S. Uchimura, M. Watanabe, T. Yasuda, Y. Mitani, S. Tomita, A Gabor filter-based method for recognizing handwritten numerals. Pattern Recognition 31(4), (1998), 395-400. [https://doi.org/10.1016/S0031-3203\(97\)00057-5](https://doi.org/10.1016/S0031-3203(97)00057-5).
- [82] Shipra Saxena. Lenet 5. <https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/>. (Erişim Tarihi: 07.05.2022).
- [83] R. L. D. Valois, D. G. Albrecht, L. G. Thorell, Spatial-frequency selectivity of cells in macaque visual cortex. Vision Research 22(5), (1982), 545- 559. [https://doi.org/10.1016/0042-6989\(82\)90113-4](https://doi.org/10.1016/0042-6989(82)90113-4).
- [84] A. Shah, Gabor Filtre, (2018), https://medium.com/@anuj_shah/through-the-eyes-of-gabor-filter-17d1fdb3ac97 (Ziyaret tarihi: 21 Eylül 2021).
- [85] E. Deveci, B. Ergen, Yol Yüzey Anormalisinin Tespiti ve Analizi. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi , 10 (3), (2021), 1187-1194 . <https://doi.org/10.17798/bitlisfen.942386>.
- [86] T. Vedat, E. Burhan. Intersections and crosswalk detection using deep learning and image processing techniques. Physica A: Statistical Mechanics and its Applications, (2019), <https://doi.org/10.1016/j.physa.2019.123510>.
- [87] Aniruddha, Karmaşıklık Matrisi, (2020), <https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/> (Ziyaret tarihi: 20 Aralık 2021).
- [88] Ş. Yiğit Karmaşıklık Matrisi. (2020), <https://yigitsener.medium.com/veri-bilimi-s%C4%B1n%C4%B1fland%C4%B1rma-model-%C3%A7%C4%B1kt%C4%B1lar%C4%B1n%C4%B1-de%C4%9Ferlendiren-metrikler-confusion-matrix-accuracy-437f5633c82b>. (Ziyaret tarihi: 10 Ocak 2022).
- [89] S. T. Lainier. Doğruluk ve F1 skor, (2020), <https://towardsdatascience.com/choosing-performance-metrics-61b40819eae1> (Ziyaret tarihi: 5 Şubat 2022).
- [90] S. Kampakis, Cohens kapps, (2020), <https://thedata scientist.com/performance-measures-cohens-kappa-statistic/>. (Ziyaret tarihi: 9 Mart 2022).

ÖZGEÇMİŞ

Erkan DEVECİ

Category	Percentage
Very good	10%
Good	20%
Not good	30%
Very bad	40%

[REDACTED] [REDACTED]

-
- | Response | Percentage |
|--|------------|
| U.S. should take action to address climate change | 95% |
| U.S. should take strong action to address climate change | 90% |
| U.S. should take moderate action to address climate change | 5% |
| U.S. should not take action to address climate change | 5% |