



T.C.
TOKAT GAZİOSMANPAŞA ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI
MEKATRONİK MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI

VGG16 TEMELLİ MİMARİ İLE GÖRÜNTÜLERDE
GÜRÜLTÜ TAHMİNİ

YÜKSEK LİSANS TEZİ

Aybüke GÜNEŞ

Danışman: Dr. Öğr. Üyesi Yasemin Çetin Kaya

TOKAT-2022

ETİK SÖZLEŞME

Tokat Gaziosmanpaşa Üniversitesi Lisansüstü Eğitim Enstitüsü tez yazım kılavuzuna göre, Dr. Öğr. Üyesi Yasemin Çetin Kaya danışmanlığında hazırlamış olduğum “VGG16 Temelli Mimari ile Görüntülerde Gürültü Tahmini” adlı Yüksek Lisans tezinin bilimsel etik değerlere ve kurallara uygun, özgün bir çalışma olduğunu, aksinin tespit edilmesi halinde her türlü yasal yaptırımını kabul edeceğimi beyan ederim.

.../.../...

Aybüke GÜNEŞ

JÜRİ KABUL VE ONAY

Aybüke GÜNEŞ tarafından hazırlanan “VGG16 Temelli Mimari ile Görüntülerde Gürültü Tahmini” adlı tez çalışmasının savunma sınavı 10/06/2022 tarihinde yapılmış olup aşağıda verilen Jüri tarafından Oy Birliği ile Tokat Gaziosmanpaşa Üniversitesi Lisansüstü Eğitim Enstitüsü Mekatronik Anabilim Dalı’nda Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Üyeleri (Unvanı, Adı Soyadı)

İmzası

Üye (Başkan) :.....

.....

Üye :.....

.....

Üye :.....

.....

Üye :.....

.....

Üye :.....

.....

ONAY

...../...../.....

Lisansüstü Eğitim Enstitüsü Müdürü

ÖZET

VGG16 TEMELLİ MİMARİ İLE GÖRÜNTÜLERDE GÜRÜLTÜ TAHMİNİ

Güneş, Aybuke

Yüksek Lisans, Mekatronik Anabilim Dalı

Tez Danışmanı: Dr. Öğr. Üyesi Yasemin Çetin Kaya

Haziran 2022, x + 106 sayfa

Gürültü, görüntü elde etme esnasında görüntüye eklenen istenmeyen sinyallerdir. Görüntü işleme çalışmalarının en temel sorunlarından biri gürültüdür. Bir görüntüden gürültünün arındırılması için kullanılan filtre yöntemlerinin başarılı olabilmesi için gürültü türünün doğru şekilde analiz edilmiş olması gerekmektedir. Literatürde yaygın olan gürültü türleri gauss, benek ve tuz-biber gürültü türleridir. Orijinal görüntülere Matlab programa dili vasıtasıyla eklenen bu üç gürültü türünün ve gürültüsüz görüntülerin en doğru ve pratik bir şekilde sınıflandırılması amaçlanmıştır.

Görüntü üzerine yapılan çalışmalarda Evrişimsel Sinir Ağları (ESA) sıklıkla kullanılmaktadır. Çalışmada ESA ile oluşturulan 5 farklı model için önce eğitim sonra test işlemleri gerçekleştirilmiştir. Çalışmadaki ESA mimarileri; VGG16 ağı, transfer öğrenme ve Kök Ortalama Kare Yayılımı (RMSProp), Stokastik Gradyan İniş (SGD), Uyarlanabilir Gradyan(Adagrad), Adadelata ile Adaptif Moment(Adam) optimizasyon algoritmaları kullanılarak oluşturulmuştur. Görüntülerdeki gürültü türünün ve gürültüsüz görüntülerin doğru ve pratik şekilde tahmin edilmesi hedeflenmiştir. Görüntülerdeki gürültü tahmininde başarılı olunması halinde gürültülü görüntülerin, gürültülerinden arındırılması için daha doğru filtrelerin kullanılmasına olanak sağlanmış olacaktır.

RMSProp optimizasyon algoritması kullanılarak oluşturulan model toplamda 12 hatalı gürültü türü tespiti yapmıştır. RMSProp optimizasyon algoritması kullanılarak oluşturulan model %98.75 doğruluk oranı ile en iyi doğruluk oranına sahiptir. Diğer optimizasyon algoritmalarının ise doğruluk oranları sırasıyla Adam optimizasyon algoritması için %98.44, Adagrad optimizasyon algoritması için %97.29, Adadelata optimizasyon algoritması için %89.38 ve SGD optimizasyon algoritması için %57.71'dir.

Bu çalışma ile gürültü türünün ESA mimarileri ile daha doğru ve pratik bir şekilde belirlenebilmesi için hangi optimizasyon algoritmasının tercih edilebileceğine ışık tutulmaya çalışılmıştır. ESA'nın görüntü işleme çalışmalarındaki başarısı göz önüne alındığında ileride görüntü ile yapılan çalışmaların daha doğru ve güvenilir filtreler ile yapılabileceği düşünülmektedir.

Anahtar Kelimeler: Evrişimsel Sinir Ağları, VGG16, Gürültü, Gürültü Tahmini, Optimizasyon Algoritmaları

ABSTRACT

NOISE PREDICTION IN IMAGES WITH VGG16 BASED ARCHITECTURE

Güneş, Aybüke

Master's Thesis, Department of Mechatronics Engineering

Advisor: Assit.Prof.Dr. Yasemin Çetin Kaya

June 2022, x + 106 pages

Noise is unwanted signals added to the image during image acquisition. One of the most fundamental problems of image processing studies is noise. In order for the filter methods used to remove noise from the image to be successful, the noise type must be analyzed correctly. The types of noise that are common in the literature are gaussian, speckle and salt-pepper types of noise. It is aimed to classify these three noise types and noiseless images in the most accurate and practical way, which are added to the original images via the Matlab programming language.

Convolutional Neural Networks (CNN) are frequently used in studies on images. In the study, first training and then testing were carried out for 5 different models created with CNN. ESA architectures in the study; The VGG16 network was created using transfer learning and Root Mean Square Propagation (RMSProp), Stochastic Gradient Descent (SGD), Adaptive Gradient (Adagrad), Adadelata and Adaptive Moment (Adam) optimization algorithms. It is aimed to predict the noise type and noiseless images in an accurate and practical way. If the noise prediction in the images is successful, it will be possible to use more accurate filters to remove the noise of the noisy images.

The model created using the RMSProp optimization algorithm has detected a total of 12 incorrect noise types. The model created using the RMSProp optimization algorithm has the best accuracy rate with 98.75% accuracy. The accuracy rates of other optimization algorithms are 98.44% for Adam optimization algorithm, 97.29% for Adagrad optimization algorithm, 89.38% for Adadelata optimization algorithm and 57.71% for SGD optimization algorithm, respectively.

With this study, it has been tried to shed light on which optimization algorithm can be preferred in order to determine the noise type in a more accurate and practical way with CNN architectures. Considering the success of CNN in image processing studies, it is thought that future studies with images can be done with more accurate and reliable filters.

Keywords: Convolutional Neural Networks, VGG16, Noise, Noise Prediction, Optimization Algorithms

ÖNSÖZ

Derin öğrenme, son zamanlarda araştırma alanlarında sıkça kullanılan bir çalışma alanıdır. Doğal dil işlemden konuşma/ses işlemeye, görüntü işlemeden büyük veri analizlerine kadar hayatımızın her alanında yer edinmiştir. Bu çalışmanın amacı da, derin öğrenme mimarilerinden biri olan Evrişimsel sinir ağlarından VGG16 kullanılarak geliştirilen model ile görüntülerdeki gürültü türlerinin tahminini doğru ve pratik şekilde bulabilmektir.

Bu tez Gaziosmanpaşa Üniversitesi, Mekatronik Mühendisliği Anabilim Dalı Yüksek Lisans Programı'nda hazırlanmıştır.

Yüksek lisans eğitimim boyunca bana her türlü desteği sağlayan, vakit ayırıp her daim bana yol gösteren değerli danışmanım Sayın Dr. Öğr. Üyesi Yasemin ÇETİN KAYA'ya teşekkürlerimi sunarım.

Eğitim ve öğretim hayatım boyunca benden maddi ve manevi desteklerini esirgemeyen her zaman yanımda olan bugünlere gelmem de büyük emekleri olan aileme sevgi ve saygılarımı sunarım.

Her daim yanımda olup desteklerini hiçbir zaman esirgemeyen, ailemden ayırmadığım manevi kız kardeşim Harita Mühendisi Hilal KAYHAN'a ve Bilgi Teknolojileri ve İletişim Kurumunda Bilişim Uzm. Yard. olarak görev yapan kıymetli dostum Yasemin YAMAK'a teşekkürü bir borç bilirim.

Aybüke GÜNEŞ
Haziran 2022

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	ii
ÖNSÖZ	iii
İÇİNDEKİLER	iv
ÇİZELGELER LİSTESİ	vi
ŞEKİLLER LİSTESİ	vii
KISALTMALAR	x
1. GİRİŞ	1
2. GENEL BİLGİLER	5
2.1 Gürültü Tahmini	5
2.2 Derin Öğrenme	7
2.2.1 Tarihsel eğilimler	13
2.2.2 Evrimsel sinir ağları	17
2.2.3 VGGNet mimarisi	25
2.2.4 Transfer öğrenme	26
2.2.5 Optimizasyon algoritmaları	27
2.2.6 Başarım ölçütleri	30
3. LİTERATÜR ÖZETLERİ	33
4. MATERYAL ve YÖNTEM	40
4.1 Veri Seti Oluşturulması	41
4.2 Derin Öğrenme Geliştirme Ortamı	47
4.3 Uygulama Ağının Oluşturulması	48
4.4 Değerlendirme	51
5. BULGULAR ve TARTIŞMA	52
5.1 Optimizasyon Algoritmalarının Karşılaştırması	52
5.1.1 RMSProp	53
5.1.2 Adam	55
5.1.3 Adagrad	57
5.1.4 Adadelat	58
5.1.5 SGD	60
5.2 Katman Karşılaştırması	66
5.2.1 Bir (1) katman eklenerek oluşturulan VGG16 mimarisi	67
5.2.2 İki (2) katman eklenerek oluşturulan VGG16 mimarisi	68
5.2.3 Üç (3) katman eklenerek oluşturulan VGG16 mimarisi	70
5.2.4 Beş (5) katman eklenerek oluşturulan VGG16 mimarisi	71

6. SONUÇ ve ÖNERİLER	79
KAYNAKÇA	81
EKLER	89
Ek 1. Optimizasyon Algoritmaları için Hatalı Bulunan Görüntüler	89
Ek 2. RMSProp Optimizasyon Algoritması için Katman Farklılıklarında Hatalı Bulunan Görüntüler	97
Ek 3. Yazarın Özgeçmişi	106



ÇİZELGELER LİSTESİ

Çizelge No	Sayfa
Çizelge 2.1 Derin öğrenme için adımlar	8
Çizelge 2.2 Derin öğrenme ağ yapıları	10
Çizelge 2.3 Derin öğrenme kütüphaneleri	11
Çizelge 2.4 Çalışmada yer alan optimizasyon algoritmalarının açıklamaları ve formülleri	27
Çizelge 2.5 İkili sınıflandırma için karışıklık matrisi yapısı (Öğündür, 2019)	31
Çizelge 4.1 Çalışmada takip edilecek aşamalar	40
Çizelge 4.2 Model karışıklık matrisi gösterimi	51
Çizelge 5.1 Çalışmada kullanılan optimizasyon algoritmalarının doğrulukları	62
Çizelge 5.2 Optimizasyon algoritmalarının karışıklık matrisleri	63
Çizelge 5.3 Modelin performans ölçümleri	66
Çizelge 5.4 Kullanılan katman sayısına göre RMSProp optimizasyon algoritmasının doğrulukları	73
Çizelge 5.5 Katman sayısına bağlı modellerin karışıklık matrisleri	74
Çizelge 5.6 Katman sayısına göre modellerin performans ölçümleri	76
Çizelge 5.7 Literatürde yer alan bazı örneklerin yapılan çalışma ile karşılaştırması	76

ŞEKİLLER LİSTESİ

Şekil No	Sayfa
Şekil 1.1 Makine öğrenmesi ve derin öğrenme arasındaki temel fark (Gülal,2022).....	1
Şekil 1.2 Derin öğrenme yapısı (Savaş, 2022).....	2
Şekil 2.1 Gürültülü görüntülere örnekler: a) Orijinal Kameraman görüntüsü, b) Gauss gürültüsü eklenmiş Kameraman görüntüsü, c) Benek gürültüsü eklenmiş Kameraman görüntüsü, ç) Tuz-biber gürültüsü eklenmiş Kameraman görüntüsü.....	7
Şekil 2.2 Yapay zekâ, makine öğrenmesi ve derin öğrenme arasındaki ilişki (Murat, 2021).....	8
Şekil 2.3 Derin öğrenme modeli genel mimarisi (Çınar, 2017).....	9
Şekil 2.4 Basit bir Perceptron yapısı (Saporito, 2019).....	15
Şekil 2.5 Yapay zekâ, makine öğrenmesi ve derin öğrenme tarihsel süreci (Kayaalp & Süzen, 2018)	16
Şekil 2.6 (a) İnsan sinir hücresi (b) Yapay sinir hücresi (Murat, 2021).....	16
Şekil 2.7 Geleneksel bir ESA mimarisi (Fırıldak & Talu, 2019).....	18
Şekil 2.8 Görüntüleri verimli bir şekilde işlemek için birkaç evrişim ve alt örnekleme katmanından oluşan ESA'nın temel mimarisi (Ravi ve ark., 2017).....	18
Şekil 2.9 Evrişim İşlemi(Doğan & Türkoğlu, 2019).....	19
Şekil 2.10 Lena görüntüsü kenar bulma işlemi için özellik çıkarma örneği(Anonim a, 2021)... ..	20
Şekil 2.11 Kenarlara sıfır ekleme yöntemi ile piksel ekleme örneği (Setiawan ve ark., 2019) ..	21
Şekil 2.12 Relu aktivasyon fonksiyonu grafiği (Liu, 2017).....	22
Şekil 2.13 En büyük değer havuzlama yöntemi örneği (Kın, 2019).....	23
Şekil 2.14 Ortalama değer havuzlama yöntemi örneği (Kın, 2019).....	23
Şekil 2.15 Tam bağlı katman (Kın,2019).....	23
Şekil 2.16 a) Dropout olmayan standart ağ görünümü, b) Dropout kullanılan ağ görünümü (Uysal ve ark., 2018).....	24
Şekil 2.17 VGGNet model yapısı (Simonyan & Zisserman, 2015)	25
Şekil 2.18 Tam bir VGG16 mimarisi (Hu, 2020)	26
Şekil 2.19 Transfer öğrenmenin temelleri. (Li ve ark., 2021).....	26
Şekil 2.20 Optimizasyon algoritmalarının evrimsel haritası (Seyyarer ve ark., 2020)	30
Şekil 4.1 Gürültü ekleme işlemine örnek	41
Şekil 4.2 Eğitim, doğrulama ve test seti arasındaki ilişki	42
Şekil 4.3 Veri seti klasör yapısı.....	43
Şekil 4.4 Gauss gürültüsü eklenmiş eğitim veri seti örnekleri	43
Şekil 4.5 Orijinal görüntüler eğitim veri seti örnekleri	44
Şekil 4.6 Benek gürültüsü eklenmiş eğitim veri seti örnekleri	44
Şekil 4.7 Tuz-biber gürültüsü eklenmiş eğitim veri seti örnekleri	45

Şekil 4.8 Gauss gürültüsü eklenmiş test veri seti örnekleri.....	45
Şekil 4.9 Orijinal görüntü test veri seti örnekleri	46
Şekil 4.10 Benek gürültüsü eklenmiş eğitim veri seti örnekleri	46
Şekil 4.11 Tuz-biber gürültüsü eklenmiş eğitim veri seti örnekleri	47
Şekil 4.12 Google Colaboratory platformu	47
Şekil 4.13 Model için transfer öğrenme kullanımının temsili gösterimi.....	49
Şekil 4.14 Model mimarisi diyagramı.....	50
Şekil 4.15 Model mimarisi.....	50
Şekil 5.1 RMSProp optimizasyon algoritması için; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği.....	53
Şekil 5.2 RMSProp optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntülere örnekler	54
Şekil 5.3 Adam optimizasyon algoritması için; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği.....	55
Şekil 5.4 Adam optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntülere örnekler	56
Şekil 5.5 Adagrad optimizasyon algoritması için; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği	57
Şekil 5.6 Adagrad optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntülere örnekler	57
Şekil 5.7 Adadelta optimizasyon algoritması için; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği.....	58
Şekil 5.8 Adadelta optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntülere örnekler	59
Şekil 5.9 SGD optimizasyon algoritması için; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği.....	60
Şekil 5.10 SGD optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntülere örnekler	61
Şekil 5.11 Çizelge 5.1'in grafik gösterimi	62
Şekil 5.12 RMSProp optimizasyon algoritması için bir katmanlı mimari; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği.....	67
Şekil 5.13 RMSProp optimizasyon algoritması ile oluşturulan bir katmanlı modelin test sonucunda hatalı bulunan görüntülere örnekler	67
Şekil 5.14 RMSProp optimizasyon algoritması için iki katmanlı mimari; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği.....	68
Şekil 5.15 RMSProp optimizasyon algoritması ile oluşturulan iki (2) katmanlı modelin test sonucunda hatalı bulunan görüntülere örnekler	69
Şekil 5.16 RMSProp optimizasyon algoritması için üç katmanlı mimari; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği.....	70

Şekil 5.17 RMSProp optimizasyon algoritması ile oluşturulan üç (3) katmanlı modelin test sonucunda hatalı bulunan görüntülere örnekler	70
Şekil 5.18 RMSProp optimizasyon algoritması için beş katmanlı mimari; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği.....	71
Şekil 5.19 RMSProp optimizasyon algoritması ile oluşturulan beş (5) katmanlı modelin test sonucunda hatalı bulunan görüntülere örnekler	72
Şekil 5.20 Çizelge 5.4'ün grafik gösterimi	73

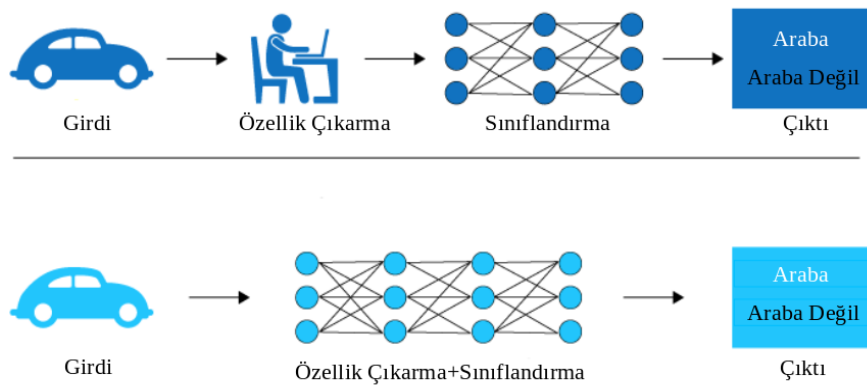


KISALTMALAR

AdaGrad: Uyarlanabilir Gradyan
Adam : Adaptif Moment
BM3D: Blok Eşleştirme ve 3D Filtreleme
DN: Doğru Negatifler
DnCNN: İleri Beslemeli Evrişimsel Sinir Ağları
DP: Doğru Pozitifler
DRUNet: Genişletilmiş - artık UNet
DWT: Ayrık Dalgacık Dönüşümü
EMD: Deneysel Mod Ayırıştırması
ESA: Evrişimsel Sinir Ağları
fBm: Kesirli Brownian Hareketi
FFDNet: Hızlı ve Esnek Evrişimsel Sinir Ağı
MLP: Çok Katmanlı Algılayıcı
MRI: Manyetik Rezonans Görüntüsü
MRF: Markov Rastgele Filtre Yöntemi
MSE: Ortalama Kare Hatası
PSNR: Tepe Sinyal Oranı
Relu: Düzeltilmiş Doğrusal Birim
RMSProp: Kök Ortalama Kare Yayılımı
SGD: Stokastik Gradyan İniş
VGG: Görsel Geometri Grubu
VDM: Varyasyonel Mod Ayırışımı
YN: Yanlış Negatifler
YP: Yanlış Pozitifler

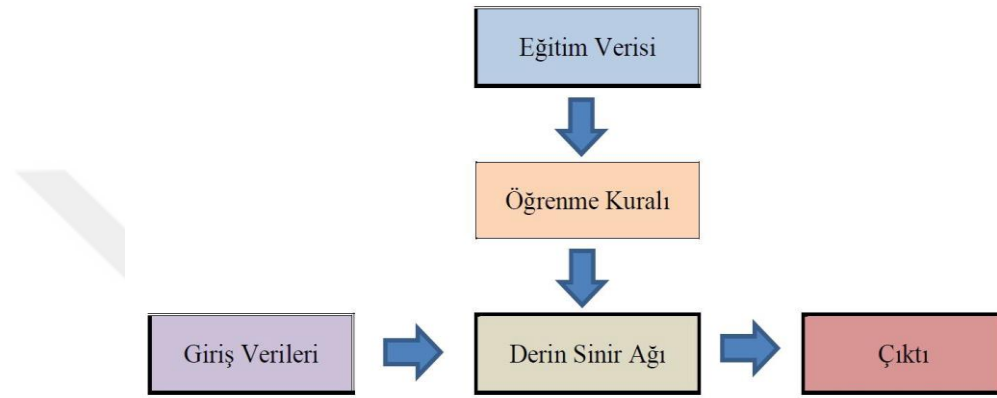
1. GİRİŞ

Yapay zekânın bir alt dalı olan makine öğrenmesinde, yapay sinir ağları kullanılarak birçok problemin çözümü sağlanmıştır. Kullanılan yapay sinir ağı “Perceptron” olarak adlandırılan tek katmanlı bir işlem ünitesiydi (Kın, 2019). 1960’larda bu tek katmanlı ünitenin karmaşık problemlere çözüm oluşturamadığı ortaya çıkmıştır. Bu dönemde yaşanan donanımsal kısıtlamalar da bilim insanlarının çalışmalarına ara vermelerine neden olmuştur. 1980’li yıllarda bir grup insan, yapay zekânın tekrardan popülerlik kazanmasını sağlamış ve çok katmanlı sinir ağları ile karmaşık problemlere çözümler üretilebileceğini belirtmiştir (Karakuş, 2018). 2000’li yıllarda yapay sinir ağları artık tek katmanlı ağlardan derin ağlara geçiş yapmıştır. Derin ağlar; doğal dil işlemeden, konuşma işlemeye, görüntü işlemeden büyük verilerin işlenmesine kadar birçok alanda başarıyla kullanılmaya başlanmıştır (Yapıcı ve ark., 2019). Derin öğrenme olarak adlandırdığımız derin ağlar yöntemi, son zamanlarda pek çok problemin çözümünde kullanılmaya başlanmış önemli bir yapay zekâ yaklaşımıdır. Makine öğrenmesinde insan etkisi ile yapılan tanımlamaları, verilen parametreleri insan etkisini devre dışı bırakarak kendisinin keşfedip değerlendirmelerde bulunan derin öğrenme, günümüzdeki donanımsal gelişmeler ve daha geniş veri setlerinin olması ile popülerliğini artırmıştır (Kın, 2019). Şekil 1.1’de derin öğrenme ve makine öğrenmesi arasındaki insan etkisi farkını gösteren bir görsel yer almaktadır.



Şekil 1.1 Makine öğrenmesi ve derin öğrenme arasındaki temel fark (Gülal,2022)

Derin öğrenme yöntemleri ile geliştirilen birçok algoritmanın farklı problemlerin çözümüne olanak sağlaması hedeflenmiştir. Evrimsel sinir ağları (ESA), derin oto kodlayıcılar, tekrarlayan sinir ağları gibi birçok derin öğrenme algoritması (Küçük & Arıcı, 2018) ve aynı zamanda LeNet, AlexNet, VGGNet, GoogleNet gibi birçok ağ yapısı mevcuttur (Karakuş, 2018). ESA algoritması ve VGGNet, çalışmanın metodunu oluşturmaktadır. Şekil 1.2 'de derin öğrenmenin yapısına yer verilmiştir.



Şekil 1.2 Derin öğrenme yapısı (Savaş, 2022)

Günümüzde popülerliği yüksek olan bir diğer konu ise görüntülerdeki gürültü tahminidir. Görüntü elde etme esnasında görüntüye çeşitli sebeplerden dolayı gürültüler eklenmektedir (Akar ve ark., 2015). Farklı etkenlerin görüntülerde sebep olduğu gürültüler, görüntünün kullanıldığı alana göre değişik sıkıntılara sebep olmaktadır. Ancak görüntünün olduğu her yerde görüntü kalitesinin iyi olması istenmektedir. Literatürde sıklıkla karşılaşılan gürültü türleri Gauss, benek ve tuz-biber gürültüsüdür (Akar ve ark., 2015). Bu tür gürültülerin görüntülerden arındırılması için Gauss ve uyarlamalı medyan filtresi gibi pek çok yöntem bulunmaktadır (Değirmenci ve ark., 2018). Yöntemlerin başarı oranları gürültü türüne göre farklılık göstermektedir. Gürültü türünün hatalı tahmininin görüntülerin gürültüden arındırılması için kullanılan yöntemlerin başarısını olumsuz etkileyeceği, gürültü türünün doğru şekilde tahmin edilmesinin ise görüntülere uygulanacak gürültü arındırma yöntemlerinin verimliliğinin artırılmasına katkı sağlayabileceği düşünülmektedir. Dolayısıyla gürültü türünün doğru bir şekilde tahmin edilmesi, uygun filtre kullanımını sağlamak ve başarı oranını artırmak için önem arz etmektedir (Magud ve ark., 2016).

Savunma sanayiden sağlık sektörüne, online bireysel başvurulardan güvenlik sistemlerine pek çok alanda görüntü hayatımızın içine dahil olmuştur. Güvenlik tedbirleri ve doğru tespit gibi konularda ise görüntünün kaliteli olması istenmektedir. Ancak görüntülere eklenen gürültüler görüntülerin kalitesini düşürmektedir (Hoomod & Dawood, 2017). Gürültü tahmini ve görüntüleri bu gürültülerden arındırmak için literatürde yer edinmiş farklı yöntemler mevcuttur. 2019 yılında Tassano ve ark., ESA mimarisine dayanan yeni görüntü gürültü giderme yöntemi olan hızlı ve esnek evrimsel sinir ağını (FFDNet) Adam optimizasyon algoritması ile kullanmayı önermiştir. Gri tonlu ve renkli görüntüler üzerinde denenen FFDNet gürültü gidericisi her iki renk veri seti içinde benzer değerler vermiştir. Düz ESA yapısı ile kıyaslanmış olan FFDNet küçük bir farkla başarılı olmuştur. İleri beslemeli evrimsel sinir ağıları (DnCNN) ile karşılaştırılan FFDNet'in, DnCNN'den daha hızlı ve daha etkili olduğu görülmüştür (Tassano ve ark., 2019). 2020 yılında Ponomarenko ve ark., gauss gürültüsünü görüntülerden arındırarak orijinal gürültüsüz görüntüler elde etmek için geliştirilen Blok Eşleştirme ve 3D Filtreleme (BM3D) ve Genişletilmiş-Artık U-Net (DRUNet) isimli iki adet ESA modeli geliştirmişlerdir. BM3D filtresi kullanılan model, mevcut çözümlerden daha verimli iken DRUNet kullanılan modelin mevcut yöntemlerden hem daha etkili hem de hızlı olduğu görülmüştür (Ponomarenko ve ark., 2020).

Gürültü arındırma işlemi görüntüler üzerine gerçekleştirilen çalışmalar için son derece önemlidir. Ancak görüntülerin gürültülerden arındırılmasındaki en önemli unsur gürültü türünün tespitidir. Bilinmeyen bir gürültü türüne uygulanacak filtreler istenen performansı her zaman göstermeyebilir. Bu sebeple doğru ve etkin bir şekilde gürültü türünün tahmin edilmesi gerekmektedir. Bu çalışma ile gürültü türünün ESA mimarileri ile daha doğru ve pratik bir şekilde tespit edilmesi için hangi optimizasyon algoritmasının tercih edilebileceğine ışık tutulmaya çalışılmıştır. Kaur ve Gandhi beyin görüntülerinin sınıflandırılması için VGG16 ve transfer öğrenme kullanımını önermiştir ve çalışma %100 tanıma oranı sağlamıştır (Kaur & Gandhi, 2019). Sil ve ark., ESA'ların, görüntüden gürültü gidermek için kullanılan diğer algoritmalara göre nasıl performans gösterdiğini kıyaslamak için VGG16 ve Inception-v3 algoritmalarını kullanmış ve Inception-v3 mimarisi, testte %1 oranında VGG16 mimarisinden daha

fazla başarımlı sağlamıştır (Sil ve ark., 2019). Derin öğrenmenin görüntü işlemlerindeki başarısı, ESA'nın geleneksel yöntemlerle karşılaştırıldığında çok daha basit uygulanabilmesi göz önüne alınarak görüntüde gürültü tahmini için yapılacak çalışmada ESA'nın ve VGG16 ağı'nın kullanılması kararlaştırılmıştır. VGG16'nın başarısı, 2019 yılında cinsiyet tahmini üzerine yapılan Gündüz ve Cedimoğlu'nun çalışmasında da kendini göstermiştir. Gündüz ve Cedimoğlu'nun kendi tasarladıkları ESA mimarisinin başarı oranı AlexNet'ten iyiyken, VGG16'dan kötü sonuç vermiştir (Gündüz & Cedimoğlu 2019).

2014 yılında gerçekleştirilen ImageNet yarışmasında %7.3 hata oranı (Simonyan & Zisserman, 2015) ile başarı gösteren VGG16 mimarisinin son 4 katmanı hariç diğer katmanları dondurulmuş ve üzerine yeni model eklenmiştir. Oluşturulan bu modelde VGG16'nın ImageNet yarışmasında elde edilmiş ağırlıkları kullanılarak transfer öğrenme gerçekleştirilmiştir. Dondurulan katmanlardan sonra iki blok eklenmiş, her blokta iki evrişim katmanı ve her evrişim işleminden sonra bir ortaklama katmanı eklenmiştir. Sonrasında iki adet tam bağlı katman eklenmiş ve aşırı öğrenmeyi önlemek için tam bağlı katmandan sonra 0.3 oranında dropout kullanılmıştır. Bunlara ilave çıkış katmanının da eklenmesi ile yeni model kurulmuştur.

Doğru ve etkin şekilde gürültü tahmini gerçekleştirilmesi hedeflenen çalışmanın uygulama aşında, 5 farklı optimizasyon algoritmasına (RMSProp, Adam, Adagrad, Adadelta, SGD) göre 70 turda eğitim yapılmış ve testler sonucunda açığa çıkan değerler kullanılan başarımlı ölçütleri ile kıyaslanmıştır. Gürültü tahmini için geliştirilecek ESA mimarilerinde daha pratik ve daha doğru sonuçlar elde edebilmek için ileride yapılacak çalışmalarda hangi optimizasyon algoritmalarının tercih edilebileceği hususunda bilgi verilmek istenmektedir. Gürültü nedir, gürültü tahmini neden önemlidir, literatürde hangi yöntemler kullanılmış ve başarımlı düzeyleri nelerdir, gürültü tahmini için kullanılacak yöntem ile hedeflenen nedir, derin öğrenme yöntemi seçilirken neler göz önüne alınmıştır gibi sorular bu çalışmanın temellerini oluşturmaktadır. Bu tezin sonraki bölümlerinde literatür taraması, çalışmada kullanılacak materyal ve yöntemler, çalışma için kullanılacak yöntemin nasıl kullanıldığı hakkında bilgiler, elde edilen bulgular ve son olarak sonuçlara yer verilerek önerilerde bulunulmuştur.

2. GENEL BİLGİLER

Görüntü hayatımızın birçok alanının ayrılmaz bir parçası haline gelmiştir (Hoomod & Dawood, 2017). Tıpta bir hastalığın teşhisinde, askeri alanda uydu görüntülerinde, özel anların unutulmaz kılınması istendiğinde, kimlik doğrulama işlemlerinin gerçekleştirilmesinde ve benzeri birçok durumda görüntüden ve görüntünün kusursuzluğunun istenmesinden bahsetmek mümkündür. Görüntülerde soruna sebep olan unsurlar araştırılmış ve bunların önüne geçmek için teknolojik gelişmeler doğrultusunda çözümler üretilmeye çalışılmıştır. Görüntülerde meydana alan sorunların sebebi gürültülerdir (Hoomod & Dawood, 2017) ve görüntülerdeki gürültü tahmininin doğru ve pratik bir şekilde gerçekleştirilebilmesi görüntülerin gürültüden arındırılabilmesi ve kusursuzlaştırılabilmesi için önem arz etmektedir (Tai & Yang, 2008).

Derin öğrenme algoritması olan ESA ve bir derin öğrenme modeli olan VGG16 görüntü işlemlerinde yaygın olarak kullanılmaktadır (Daşkın ve ark., 2018). Bu bölümde çalışmanın ana konusu olan gürültü tahmininden ve derin öğrenmeden, ayrıca derin öğrenmenin daha iyi anlaşılabilmesi için yapay zekadan başlayarak günümüze uzanan derin öğrenme gelişmelerinden ve bir derin öğrenme algoritması olan ESA ile bir derin öğrenme modeli olan VGG16 modeli ile çalışmada yer alan transfer öğrenme, optimizasyon algoritmaları ve başarımlar ölçütlerinden bahsedilecektir.

2.1 Gürültü Tahmini

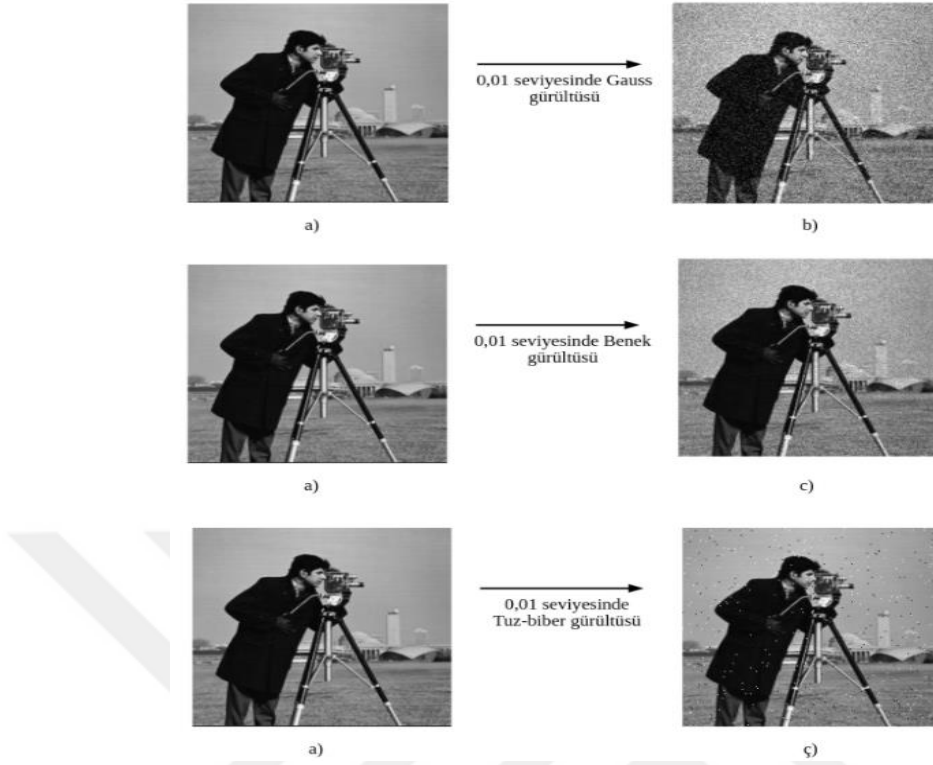
Görüntü elde etme esnasında görüntüye eklenen istenmeyen sinyaller gürültü olarak adlandırılmaktadır. Görüntü işleme tekniklerinde başarıya etki eden en önemli unsurlardan biri gürültüdür. Gürültülü görüntülere filtreleme işlemi uygulanarak görüntülerdeki gürültü azaltılmaya çalışılmakta ve bu sayede görüntü işleme tekniklerinin başarısının artırılması hedeflenmektedir (Değirmenci, 2018). Görüntüler elde edilirken veya bir yerden bir yere aktarılırken görüntülerde meydana gelen gürültülere farklı faktörler sebep olmaktadır. Görüntülerde gürültüye sebep olan bazı faktörler;

- Görüntü sensörünün çevresel etkenlerden etkilenmesi,
- Işığın yetersiz seviyede olması ve sensördeki sıcaklığın görüntüde parazite sebep olması,
- İletim kanalında olan parazitlerin görüntüyü bozabilmesi,
- Tarayıcı ekranında var olması muhtemel toz parçacıklarının görüntüde parazite neden olabilmesi,

şeklinde sıralanabilir. (Verma & Ali, 2013).

Görüntülerde meydana gelen gürültüler görüntülerin kalitesini düşürmek ile birlikte görüntü aktarma ve işleme üzerindeki verimliliği de olumsuz etkilemektedir (Hu, 2020). Gürültünün görüntüyü farklı şekilde etkilediği ve gürültüler farklı etkenlerden dolayı oluştuğu için birden fazla gürültü türü mevcuttur. Gürültü türleri; gauss gürültüsü, tuz-biber gürültüsü, benek gürültüsü, çekim gürültüsü, saçma gürültüsü, film tanecik gürültüsü, periyodik gürültü, eş yönlü gürültü ve tek biçimli gürültü olarak sınıflandırılabilir (Verma & Ali, 2013). Literatürde en sık rastlanan gürültü türleri tuz-biber, gauss ve benek gürültüsüdür.

Tuz-biber gürültüsü, veri iletimi sırasında yaşanan hatalardan kaynaklı oluşan bir gürültü türüdür. Ani ve keskin sinyal değişimleri sonucu ortaya çıkmaktadır (Kaur, 2015). Görüntüde tuz ve biber serpiştirilmiş gibi duran siyah ve beyaz renkli noktaların oluşmasına neden olmaktadır. Gauss gürültüsü genellikle görüntü elde edilirken oluşan sinyal hatalarından kaynaklanmaktadır. Sıcaklığın yüksek olması, az miktarda aydınlatma olması ve transferden kaynaklı görüntüye eklenen bir gürültü türüdür. Gürültülü görüntü, gauss gürültüsünde gerçek piksel değerlerinin rastgele gauss gürültü değerleri ile toplanması sonucunda ortaya çıkmaktadır (Küpeli & Bulut, 2020). Benek gürültüsü ise gauss ve tuz-biber gürültülerinden farklı şekilde piksel değerleri ile rastgele değerlerin çarpılması sonucu oluşmaktadır (Kaur, 2015). Şekil 2.1’de orijinal Kameraman görüntüsüne gauss, benek ve tuz-biber gürültüleri eklendiğinde oluşan gürültülü görüntülere birer örnek verilmiştir.



Şekil 2.1 Gürültülü görüntülere örnekler: a) Orijinal Kameraman görüntüsü, b) Gauss gürültüsü eklenmiş Kameraman görüntüsü, c) Benek gürültüsü eklenmiş Kameraman görüntüsü, ç) Tuz-biber gürültüsü eklenmiş Kameraman görüntüsü

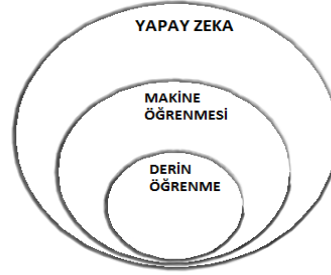
2.2 Derin Öğrenme

Derin öğrenme yapay zekânın birçok yaklaşımında kullanılan makine öğrenmesinin bir sınıfıdır. Şekil 2.2’de yapay zekâ, makine öğrenmesi ve derin öğrenme arasındaki ilişkiye dair bir görsel yer almaktadır. Çok katmanlı yapay sinir ağlarından oluşan (Aktaş, 2020) derin öğrenme;

- Bir veriden anlamlı ve istenen bilgilerin bilgisayarın anlayabileceği şekilde anlamlar çıkarılmasının sağlandığı Doğal Dil İşleme (Küçük & Arıcı, 2014),
- Konuşma cümlelerini yüksek doğruluk oranıyla metne dönüştürme işlemi olan Konuşma ve Ses İşleme (Büyük, 2018),
- Verilerin gittikçe artış göstermesi sonucu ilgili veriye erişimin zorlaştığı ve bu zorluğun üstesinden gelebilmek için derin öğrenme ile bilgi erişimi çalışmalarının yapıldığı Bilgi Erişimi,

- Görme duyumuzla yaptığımız algılama biçimini bilgisayara yaptırma amacı taşıyan Bilgisayarlı Görme ve günümüzde pek çok alanda kullanılan bilgisayarlı görü uygulamaları için önemli bir gereksinim alan Nesne Tanıma,
- Elektronik ortam kullanımının artmasıyla geniş kitlelere ulaşan büyük miktardaki verileri ifade eden derin öğrenme uygulamalarını çok daha kolay hale getiren Büyük Veri (Aktan,2018)

alanlarında başarı göstermektedir.



Şekil 2.2 Yapay zekâ, makine öğrenmesi ve derin öğrenme arasındaki ilişki (Murat, 2021)

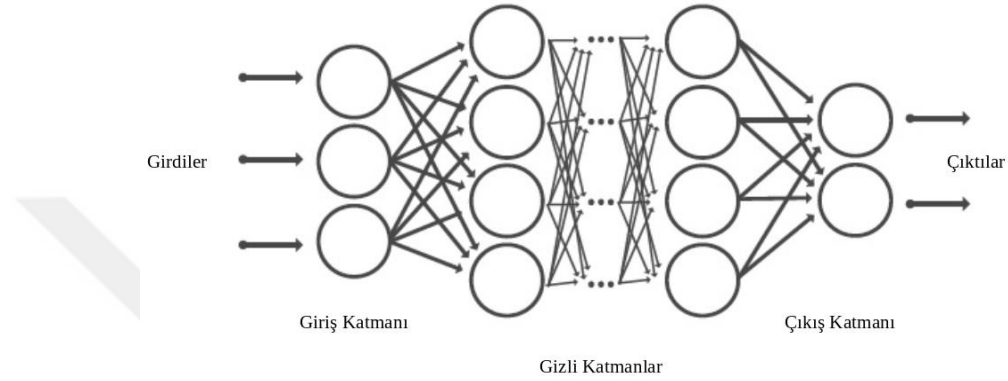
Bir problemin çözümünde derin öğrenme kullanımına karar verilmesinden sonuca kadar temelde beş adımdan bahsetmek mümkündür. Çizelge 2.1’de günümüzde çoğu problemin çözümünde kullanımı tercih edilen derin öğrenme için adımlar verilmiştir (Kayaalp & Süzen, 2018).

Çizelge 2.1 Derin öğrenme için adımlar

Adım 1:	Problemi Tanımla ve Derin Öğrenme ile Çözülüp Çözilemeyeceğini Tespit Et
Adım 2:	Veri Kümelerini Tanımla ve Verileri Analiz için Hazırla
Adım 3:	Kullanılacak Derin Öğrenme Algoritmasını Seç
Adım 4:	Tanımlı Verileri Seçilen Algoritma ile Eğit
Adım 5:	Eğitilmiş Modeli Tanımsız Verilerle Test Et

Derin sinir ağları olarak da isimlendirilen derin öğrenmede, girdi katmanı ile çıktı katmanı arasında birçok gizli katman bulunmaktadır (Bayrakdar & Doğan, 2021). Bu ağ, çok fazla katman ve parametre içerdiğinden, girdi ve çıktı arasındaki karmaşık

ilişkileri çok daha kolay öğrenebilmektedir. Art arda gelen katmanlar, sinir ağı olarak nitelendirdiğimiz yapıdır ve girdi verileri bu katmanlarda ağırlıklar olarak depolanmaktadır. Bir derin öğrenme modeli, ağıdaki tüm katmanların ağırlıkları için bir dizi değer bulmaya ve sonrasında bu ağırlıkları kullanarak girdi verilerini ilişkili olduğu gerçek sınıflarla doğru eşleştirmeye çalışmaktadır (Korkmaz, 2019). Şekil 2.3'te derin öğrenme modeline ait genel mimariye yer verilmiştir.



Şekil 2.3 Derin öğrenme modeli genel mimarisi (Çınar, 2017)

Derin öğrenmenin ESA, sınırlı Boltzman makineleri, derin oto kodlayıcılar, tekrarlayan sinir ağıları, uzun-kısa vadeli hafıza ağıları ve derin inanç ağıları olmak üzere birden fazla derin öğrenme algoritması bulunmaktadır (Şeker ve ark., 2017). ESA, iki boyutlu görseller ve çok boyutlu girdi verileri için tavsiye edilen bir derin öğrenme yöntemidir (Küçük & Arıcı, 2018). Sınırlı Boltzman makineleri, iki katmanlı, veri kümesindeki olasılıksal dağılımları öğrenebilen rastlantısal bir sinir ağıdır (Kayaalp & Süzen, 2018). Derin oto kodlayıcılar, etiketi olmayan verilerin benzer şekilde çıktı olarak bulunmasını amaçlayan sinir ağıdır. Tekrarlayan Sinir ağıları, veri akışlarını analiz edebilen gizli katmanlara sahip bir sinir ağı olması nedeniyle çıktının bir önceki hesaplamalara bağlı olduğu metin ve konuşma analizi gibi problemlerin çözümde büyük başarılar göstermektedir (Ravi ve ark., 2017). Uzun-kısa vadeli hafıza ağıları, tekrarlayan sinir ağılarında yaşanan bazı sıkıntıların giderilmesi için çıkarılmış bir sinir ağıdır (Kayaalp & Süzen, 2018). Derin inanç ağıları, düğümler arasında bağlantı olmayan fakat katmanlar arasında bağlantı bulunan birden fazla Sınırlı Boltzman Makinesinin bileşkesinden oluşan bir sinir ağıdır (Ravi ve ark., 2017).

Derin öğrenmenin ilk ağ yapısı sayılan LeNet 1990 yılında Yann LeCun tarafından geliştirilmiş rakam tanıma için kullanılmıştır bir ağıdır (Uçar, Bingöl, 2018). Derin öğrenme ağlarının aslen tanınması dünyaca ünlü ImageNet yarışması ile olmuştur. AlexNet, VggNet, ZFNet, GoogleNet ve ResNet derin öğrenme ağlarının her biri ImageNet yarışmasında elde ettiği başarılar ile görüntü işleme konusunda isimlerini duyurmuş ağlardır (Daşkın ve ark., 2018). Derin öğrenme mimarilerinin oluşturulmasında bahsi geçen ağların büyük önemi bulunmaktadır. Çizelge 2.2 ‘de derin öğrenme ağlarına ilişkin açıklamalar yer almaktadır.

Çizelge 2.2 Derin öğrenme ağ yapıları

Yıl	Ağ Yapısı	Geliştirici	Açıklamalar
1990	LeNet	Yann LeCun	- Başarılı olarak uygulanan ilk derin öğrenme ağ yapısıdır. - Rakam tanıma için kullanılmaktadır (Uçar & Bingöl, 2018). - LeNet mimarisi, giriş katmanı, evrişim katmanı, relu, havuzlama, tam bağlı katman ve yumuşak bağlı katmanlardan oluşmaktadır (Arora ve ark., 2020).
2012	AlexNet	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever	- 2012 yılındaki ImageNet yarışmasında birinci olarak en çok duyulan derin öğrenme ağ yapılarından biri haline gelmiştir (Alom ve ark., 2018). - 8 katman içeren AlexNet’in, 5 evrişim, 3 tam bağlı katmanı bulunmaktadır (Arora ve ark., 2020).
2013	ZFNet	Matthew Zeiler ve Rob Fergus	- 2013 yılında ImageNet yarışmasının kazananı olan ZFNet, AlexNet’in geliştirilmiş halidir. - AlexNet’ten farkı filtre boyutu 7x7 olarak, havuzlamadaki adım kaydırma miktarı ise 2 olarak belirlenmesidir (İnik & Ülker, 2017).

Çizelge 2.2 (Devam) Derin öğrenme ağ yapıları

2014	VggNet	Simonyan, Zisserman	- 2014 yılında Oxford Üniversitesinde geliştirilmiş ve 2014 ImageNet yarışmasında iyi performans göstermiş bir ağ yapısıdır (Simonyan & Zisserman, 2015). - Birbirine bağlı 3 tam bağlı katman bulunmaktadır. Sondaki bağlı katman 1000 nörona sahip olup çıkıştaki sınıflandırma katmanında ise yumuşak bağlı katman yer almaktadır (Kurt, 2018).
2014	GoogleNet	Google	- 2014 yılında ImageNet yarışmasının birincisi olan GoogleNet, 22 katmandan oluşan bir ağıdır (Szegedy ve ark, 2014). - Diğerlerinden farklı olarak ardışık şekilde evrişim ve havuzlama katmanları yer almamaktadır (İnik & Ülker, 2017).
2015	ResNet	Kaiming He	- 2015 yılında ImageNet yarışmasında başarı sağlamış bir ağıdır. - Diğerlerinden daha derin bir yapıya sahip olup 152 adet katman sayısı ile hepsinden fazla katmana sahiptir (Das, 2017).

Derin öğrenme için geliştirilmiş birçok kütüphane bulunmaktadır. Python dilinde yazılan kütüphaneler mevcut olduğu gibi farklı programlama dillerinde yazılan derin öğrenme kütüphaneleri de mevcuttur (Karakuş, 2018). Çizelge 2.3 'de bazı derin öğrenme kütüphaneleri hakkında kısa bilgiler verilmiştir.

Çizelge 2.3 Derin öğrenme kütüphaneleri

Kütüphane	Yazıldığı Dil	Geliştiriciler	Açıklamalar
TensorFlow	Python	Google	- Esnek bir kütüphanedir. - Derin öğrenme mimarileri için birçok algoritmanın ifade edilmesinde kullanılabilmektedir (Abadi ve ark., 2016). - Hesaplamaları çoklu GPU sunucularına dağıtır ve büyük sinir ağlarının verimli çalışmasına imkân sağlar (Géron, 2019).

Çizelge 2.3 (Devam) Derin öğrenme kütüphaneleri

Caffe	C++	Kaliforniya Üniversitesi Berkeley Vizyon ve Öğrenim Merkezi	- GPU ile görüntü işleme çalışmalarında kullanılmıştır (NVIDIA, 2017). - Modülerlik sağlaması endüstride kullanımını yaygınlaştırmıştır (Jia ve ark., 2014).
Theano	Python	MILA Lab	- GPU desteği vardır. - Gerekli olmayan değişkenleri yok sayarak karmaşık ifadelerin hesaplamasını Python dilinin de desteğiyle kolay ve hızlı yapabilmektedir (Zhang, 2016).
Torch	Lua	Ronan Collobert ve ark.	- Açık kaynaklı bir derin öğrenme ve makine öğrenmesi kitaplığıdır. - Birkaç satır kod ile kolay bir şekilde mimari oluşturulabilmektedir (Collobert ve ark., 2002).
DeepLearning4j	Java	Adam Gibson	- Dağıtık CPU ve GPU platformlarında kullanılmak için geliştirilmiştir (Heller, 2020).
Keras	Python	François Chollet	- Sinir ağlarını eğitime ve çalışma işlemini son derece basit hale getiren önemli bir derin öğrenme kütüphanesidir (Géron, 2019). - Derin öğrenme uygulamaları için, kolay ve hızlı tasarlanabilmesi ile daha iyi bir kullanıcı deneyimi sunar (Madhavan ve ark., 2021). - Theano, CNTK veya Tensorflow kütüphanelerini kendi bünyesinde çalıştırabilen bir kütüphanedir (Géron, 2019).
Lasagne	Python		- GPU destekli, Theano'nun kullanımını kolaylaştırmak için geliştirilmiş kütüphanedir. - Theano ile aynı performanstadır (Lasagne, 2015).
CNTK	Visual Studio	Microsoft	- Derin sinir ağları, ESA gibi bilinen ağların kullanımını kolaylaştırmayı amaçlamaktadır (Yu ve ark., 2016).

Çizelge 2.3 (Devam) Derin öğrenme kütüphaneleri

DIGIT	C++	NVIDIA	- Çoklu GPU desteği vardır. - Görüntü sınıflandırma, nesne algılama gibi uygulamalarda kullanılmaktadır (NVIDIA Developer, 2022). - Gerçek zamanlı performans izleme imkânı tanımaktadır (Kutlu, 2019).
Pylearn2	Python	Montreal Üniversitesi LISA Lab	- Kullanıcı dostu bir makine öğrenmesi kütüphanesidir. - Amacı makine öğrenmesini için yapılan araştırmaları kolaylaştırmaktır (Goodfellow ve ark., 2013). - Derin öğrenmede de kullanılabilir.

Her geçen gün gelişme gösteren derin öğrenme sistemlerine şirketler, devletler, kurumlar tarafından ilgi ciddi oranda artmaktadır. Pek çok büyük bilişim şirketi bu konuda atılımlar yapmış yeni yaklaşımlar getirmiştir ve halen yapmaya devam etmektedir (Anonim, 2017). Hayatımızı daha kolay hale getirmek ve yaşamımızı sağlıklı bir şekilde sürdürmek için derin öğrenmeye hayatın birçok alanında yer vermeye başlanmış olup gelecekte hayatımızın her noktasında olacağı ve otomot bir dünyaya döneceğimize inanılmaktadır. Günümüzde sürücüsüz araç teknolojileri, daha güvenli yollar, hastalık tanı ve teşhisinin daha hızlı yapılması, doktorlar yerine ameliyatları robotların yapması, savunma sanayideki araçların derin öğrenme ile güçlendirilmesi gibi sistemler üzerine çalışmalar yapılmaktadır (Şeker ve ark., 2017).

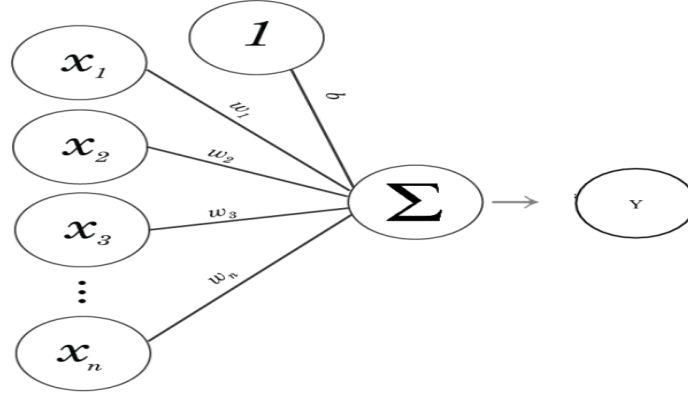
2.2.1 Tarihsel eğilimler

Derin öğrenme, yapay zekâ yaklaşımlarından olan makine öğrenmesinin alt dalıdır ve daha iyi anlaşılabilmesi, yapay zekânın tarihsel sürecinin incelenmesinden geçmektedir. Tarih boyunca insanlar makineleri bilinçli hale getirmeye çalışmışlardır (Kocaman, 2021). Yapay zekâ, düşünüp karar verebilen bir cihaz ya da yazılımı üretme işlemi olarak tanımlanabilir. Cumhurbaşkanlığı Dijital Dönüşüm Ofisi Başkanlığı ile Sanayi ve Teknoloji Bakanlığı işbirliğinde ve ilgili tüm kuruluşların etkin katılımıyla hazırlanan "Ulusal Yapay Zekâ Stratejisi 2021-2025"te yapay zekâ: "bir bilgisayarın veya bilgisayar kontrolündeki bir robotun çeşitli faaliyetleri zeki canlılara benzer şekilde yerine getirme kabiliyeti" olarak tanımlanmıştır (Anonim b, 2021).

Yapay zekânın ismen karşımıza çıkması çok uzak bir tarih olmamasına rağmen yapay zekânın tarihi 700’lü yıllara kadar dayanmaktadır. 700’lerde El-Harizmi’nin 0 rakamını keşfetmesi ile algoritmanın temelleri atılmıştır. 1100’lü yıllarda Sibernetiğin Babası olarak da bilinen El-Cezeri tarafından mekanik icatlar yapılmıştır (Demiraslan & Derin, 2019). El-Cezeri’nin en çok bilinen icadı Filli Su Saati’dir (Çalışkan, 2019). Filli Su saati su tankı görevi gören bir filin karın bölgesine yerleştirilmiş suyun içinde ortası delik bir kâsenin suyla dolması ve dolarken bağlı bulunduğu ipleri çekerek düzeneğin geri kalan unsurlarının çalıştırması mantığına dayanan bir otomattır. Bunun yanında El-Cezeri otomatik hizmetçi, müzik otomatı gibi farklı icatlara da imza atmıştır (Koştur, 2017). El-Cezeri’nin yaptığı icatlar ile Rönesans Döneminde yaşamış İtalyan asıllı Leonardo Da Vinci’e ilhâm olduğu söylenmektedir. Leonardo Da Vinci her ne kadar Mona Lisa (Wikipedia a, 2022) tablosu ile daha sık bilinse de yaşadığı dönemlerde insan anatomisine olan ilgisi nedeniyle insan gibi hareket edebilen mekanik bir robot üretmeye çalışmıştır. Ürettiği bu Şövalye Robot’un (Wikipedia b, 2022) çizim yapabildiği söylenmektedir.

1800’lere gelindiğinde ilk programcı olarak Ada Lovelace tarihteki yerini almıştır (Anonim b, 2021). Ada Lovelace, Cambridge Üniversitesi profesörlerinden Charles Babbage’nin programlanabilir mekanik makineler üzerine yazmış olduğu bir makaleye ilişkin, makaleden daha uzun açıklamalar yapmış ve açıklamalarda Babbage’in mekanik makinesi için Bernoulli sayı dizisini hesaplamaya yarayan bir yöntemden bahsetmiştir (İtik, 2020). Bu yöntem tarihte programa için ilk algoritma kabul edilirken Ada Lovelace’e de ilk programcı ünvanını getirmiştir. Makineler belirli işler için evrildikçe, makinelerin insanlar gibi düşünebileceği iddiaları daha sık ortaya atılmaya başlanmıştır. 1950 yılına gelindiğinde Alan Turing tarafından, bilgisayar ile gerçek insanı birbirinden ayırmaya yarayan Turing Testi çıkarılmıştır. Turing Testi, makinelerin düşünebilir olduğunun mantık çerçevesinde mümkün olduğunu kanıtlayabilmek için yapılmış yapay zekânın temelini oluşturan bir kavramdır (IBM, 2020). Bu dönemlerde Ülkemizin önemli profesörlerinden Ordinaryüs Profesör Cahit Arf, “Makineler Düşünebilir mi ve Nasıl düşünebilir?” adlı çalışmasını yayımlamıştır (Volaka, 2020). Yapay Zekâ ismini, John McCarthy öncülüğünde 1956 yılında yapılan Dartmouth Koleji Yapay Zekâ konferansında almıştır (IBM, 2020). Aynı yılda Frank Rosenblatt, insan beynine ait

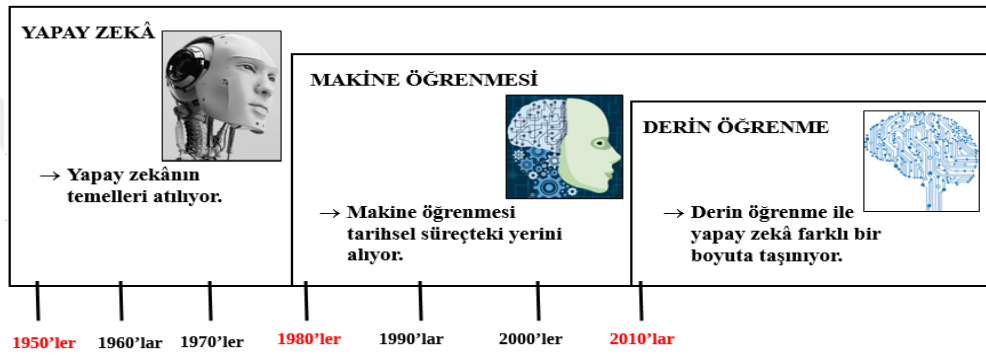
sinir hücrelerinin matematiksel modelini oluşturmuştur (Kızrak & Bolat, 2018). İnsan beyninin düşünme evresini canlandıran bu modelin ismi Perceptron'dur. Perceptron tek katmanlı bir yapay sinir ağı modelidir (Kın, 2019). İnsan beynindeki nöronların nasıl çalıştığının basitleştirilmiş bir matematiksel formülüdür (Sharma, 2017).



Şekil 2.4 Basit bir Perceptron yapısı (Saporito, 2019)

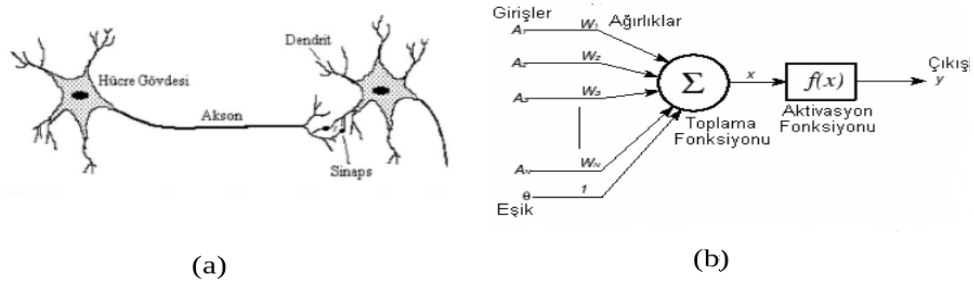
Şekil 2.4’de yer alan basit Perceptron’un yapısında da görüldüğü üzere x giriş değeri ve w ağırlık değerleri çarpılıp bias değeri ile eklenerek sonuç ortaya çıkmaktadır. Matematiksel olarak Perceptron, $f(x) = w \cdot x + b$ şeklinde ifade edilmektedir (Kara, 2019). Burada x giriş değerini, w ağırlık değerini, b bias değerini, $f(x)$ ise Şekil 2.4’deki y değerini, yani çıkışı ifade etmektedir. Ancak bu yöntem ile yalnızca AND, OR gibi tek katman ile çözülebilecek basit problemler halledilebilmektedir. XOR gibi karmaşık problemler tek katmanlı bir yapıda olan Perceptron modeli ile çözülememektedir (Doğan, 2018). Marvin Lee Minsky, bu durumu karmaşık problemlerin çözümü için birden fazla sinir ağına ve katmana ihtiyaç olduğunu savunarak ifade etmiştir. Yaşanan bu sorunlar 1960’larda donanımsal yetersizlikler ile birlikte “Yapay Zekâ Kışı” olarak isimlendirilen bir devre girilmesine sebep olmuştur ve yapay zekâ çalışmalarına ara verilmiştir (Kızrak & Bolat, 2018). Çalışmalarına devam eden bir grup insanın çok katmanlı yapay sinir ağları ile XOR problemini çözmesi 1980’li yıllarda yapay zekânın tekrardan gün yüzüne çıkmasını sağlamıştır (Azginoğlu, 2018). Yapay zekânın bu yeni döneminde artık devreye makine öğrenmesi de girmiştir. Makine öğrenmesi, yapay zekâ uygulamalarında kullanılan bir yöntemdir (Murat, 2021). Yapay zekâlı sistemler algoritmik hesaplarda kullanılmak için programlanan, hatalardan ders çıkaran sistemlerken, makine öğrenimi elindeki veriler ve parametreler ile benzetimler yaparak son derece iyi tespitlerde bulunan, kendi

kendini eğitilebilen sistemlerdir. Zamanla artan veri sayısına göre iyileşmeler gösteren algoritmalara dayanmaktadır. 1997 yılına gelindiğinde, IBM tarafından geliştirilmiş olan Deep Blue, çok büyük bir başarıya imza atarak dünyaca ünlü satranç şampiyonu Garry Kasparov’u yenmiştir (Anonim, 2020). 2000’li yıllar, makine öğrenmesinin eksikliklerinin giderilmesi için derin öğrenmenin ortaya çıktığı yıllar olmuştur (Özgür, 2021). Derin öğrenme yapay zekâ yöntemlerinden biri olan makine öğrenmesinin bir alt dalıdır. Şekil 2.5’te yapay zeka, makine öğrenmesi ve derin öğrenmenin tarihsel süreci ve birbirleri ile olan ilişkileri gösterilmektedir.



Şekil 2.5 Yapay zekâ, makine öğrenmesi ve derin öğrenme tarihsel süreci (Kayaalp & Süzen, 2018)

Derin öğrenme, yapay sinir ağlarının mantığına dayanmaktadır. İnsan beynindeki sinir hücrelerinden esinlenerek ortaya çıkan sinir ağları, derin öğrenmeyi destekleyen, görüntü tanıma ve robotik teknolojilerde önemli bir rol oynayan, çok katmanlı yapıya sahip ağlardır. Ağda ne kadar çok katman varsa ağ, o kadar derin demektir (Daşkın ve ark., 2018). İnsan beynindeki bir nöron diğer nörondan gelen sinyalleri alır ve bir sonraki nörona iletir yani nöronlar arasında bir haberleşme mevcuttur.



Şekil 2.6 (a) İnsan sinir hücresi (b) Yapay sinir hücresi (Murat, 2021)

Şekil 2.6 (a)'da görüldüğü üzere insan sinir hücrelerinde iletim, dentritlerden gelen sinyallerin belirli bir eşik değerinin üzerine çıkması sonucu aksonlar yardımıyla komşu hücrelere aktarılması ile gerçekleşmektedir. Şekil 2.6 (b)'de ise insan sinir hücrelerinin yapay sinir hücrelerine nasıl modellendiği gösterilmektedir. Bir yapay sinir ağında çıkış,

$$a = f(\sum W_i p_i + b) \quad (2.1)$$

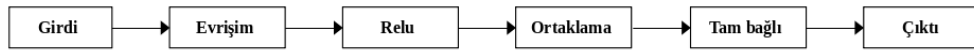
formülü ile hesaplanmaktadır. Formülde yer alan a çıkışı, b eşik değerini, $W_i p_i$ giriş sinyallerinin ağırlıkları ile çarpımlarını, f ise aktivasyon fonksiyonundan geçirme adımını ifade etmektedir. Yani yapılan işlem giriş sinyallerinin ağırlıkları ile çarpılması, önce çarpımların toplanması, sonra toplama bir eşik değerinin eklenerek aktivasyon fonksiyonundan geçirilmesidir (Kaynar & Taştan, 2009). Bir sinir ağında çıkış bu şekilde elde edilmektedir.

İnsan beynindeki nöronların çalışmasını baz alan hesaplama sistemleri derin öğrenmenin çatısını oluşturmaktadır. Derin öğrenmenin ilk temel ağı 0'dan başlayıp 9 rakamına kadarki el yazısı rakamların sınıflandırılması için kullanılan LeNet ağı olarak kabul edilmektedir (Doğan & Türkoğlu, 2019). İlk derin öğrenme ağı sayılan LeNet'in yapısı 1990 yılında sunulmuş olsa da derin öğrenmenin popülerleşmesi büyük ölçekli görsel tanıma yarışması ImageNet'te aldığı başarılar sonucunda olmuştur. Yarışmada yüksek başarı gösteren ve birincilik kazanan derin öğrenme ağları, derin öğrenmeye olan ilgiyi hızla artırmıştır. Yarışmada birçok görüntü tanıma algoritması ortaya atılmış ve bu algoritmalar başarı sağladıkça derin öğrenme mimarileri de ön plana çıkmıştır. Günümüzde birçok problemin çözümü için derin öğrenme mimarilerinin kullanımı yaygınlaşmıştır. Görüntü tanıma ve sınıflandırmada, tıpta ve finans sektöründe, gelecek tahmininde, ilaç ve savunma sanayi gibi farklı alanlarda derin öğrenme mimarilerinin kullanımı tercih edilir olmuştur (Özgür, 2021).

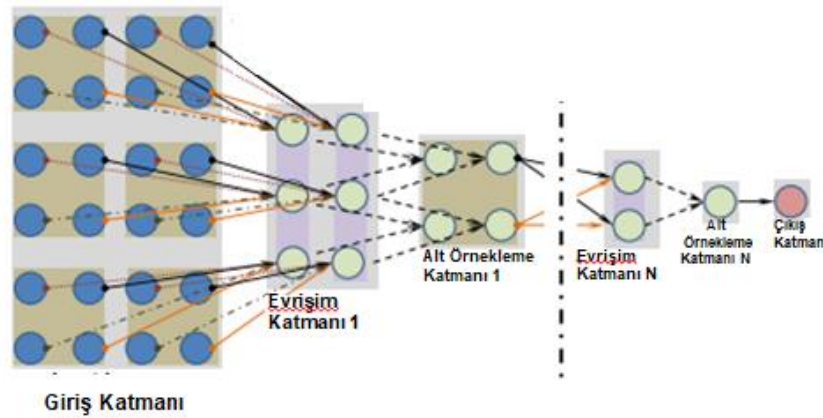
2.2.2 Evrimsel sinir ağları

Genellikle görsel bilginin analiz edilmesinde kullanılan derin öğrenme algoritması ESA, son yıllarda, çok çeşitli alanlarda farklı görüntü tanıma görevlerinde kullanılmakta ve büyük ilerleme kaydetmektedir. ESA'ların başlangıcı 1980 yıllarına dayanmasına rağmen, son birkaç yıl öncesinde sınıflandırma çalışmaları için sıklıkla tercih edilen bir

yöntem haline gelmiştir (Vrbančić & Podgorelec, 2022). İnsan beynindeki, göz retinasından gelen bilgiyi alıp yorumlayan yani, bir cisim gördüğümüzde onun hangi cisim olduğunu söyleyen görsel korteksin çalışma şekli baz alınarak geliştirilmiştir (Ravi ve ark., 2017). ESA'lar başka sinir ağlarına kıyasla az miktarda nöron bağlantısı içermektedir (Küçük & Arıcı, 2018). Bir görüntüyü girdi olarak kabul etmekte ve değişen evrişim ve alt örnekleme katmanları aracılığıyla, nesne kimliğine ilişkin tahminler sağlamak için kullanılmaktadır (Jain & Seung). Evrişim, ortaklama ve tam bağlı katman olmak üzere 3 temel katmanı bulunmaktadır (Murat, 2021). Evrişim katmanı, kenar bulma, piksel ekleme ve kaydırma adımı işlem adımları ile resimlerin özelliklerini algılamaktan sorumlu filtre uygulamaların gerçekleştiği bir katmandır. Ortaklama katmanı, ağıın parametre ve boyut sayısını azaltarak ağ katmanlarının hesaplama yükünü azaltan, maksimum değer alma veya ortalama alma yöntemi ile örnekleme uygulamalarının gerçekleştirildiği, ağıın ezberleme durumunun önüne geçmeyi sağlayan bir katmandır. Tam bağlı katman ise olasılıksal olarak sınıflandırma yapmak için kullanılan sınıf skorları gibi hedefleri elde etme uygulamalarının gerçekleştirildiği katmandır. Şekil 2.7'de ESA'nın evrişim, ortaklama ve tam bağlı katman ile ileri de bahsedilecek olan Relu adımını gösteren geleneksel bir şema yer almaktadır.



Şekil 2.7 Geleneksel bir ESA mimarisi (Fırıldak & Talu, 2019)



Şekil 2.8 Görüntüleri verimli bir şekilde işlemek için birkaç evrişim ve alt örnekleme katmanından oluşan ESA'nın temel mimarisi (Ravi ve ark., 2017)

Şekil 2.8’de de görüldüğü üzere ESA’ya ait süreç kısaca;

1. Giriş görüntüsü kullanılan bazı filtreler ile elden geçirilir,
2. Filtre uygulandıktan sonra elde edilen çıktı alt örneklenir,
3. Alt örneklemeden sonra elde edilen çıktı ise yeni girdi olarak kabul edilerek yüksek seviyeli özelliklerin çıkarılmasına kadar evrişim ve alt örnekleme işlemlerinin tekrar edilmesi sağlanır

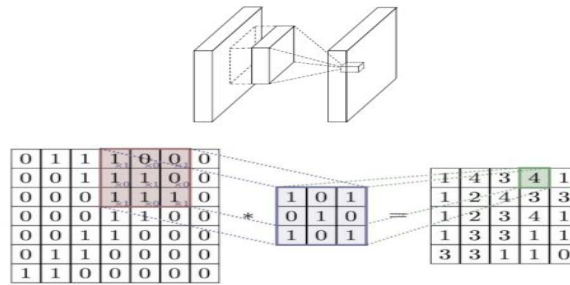
şeklinde özetlenebilir (Ravi ve ark., 2017).

Giriş Katmanı

Evrişim katmanının ilk katmanı olarak veri girişi yapılan katmandır. Giriş katmanında veriler ağa ham olarak verilmekte ve oluşturulan veri seti ağın tasarımına göre belirlenmektedir (Uysal ve ark., 2018). Oluşturulan veri setinin boyutuna göre ağ hızı, eğitim ile test süresi ve bellek ihtiyacı değişmektedir (Tan, 2019).

Evrişim Katmanı

ESA’nın ana katmanıdır. Görüntüler matris şeklinde olmaktadır. Özellik çıkarma, piksel ekleme ve kaydırma adımı işlemleri gibi evrişim işlemleri bu katmanda gerçekleştirilmektedir. Evrişim katmanı, evrişim işlemi adımları ile resmin özelliklerini algılamaktan sorumludur (Cengil & Çınar, 2019).

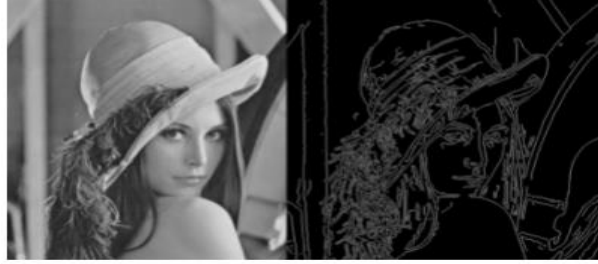


Şekil 2.9 Evrişim işlemi (Doğan & Türkoğlu, 2019)

Şekil 2.9’da örnek verilen evrişim işlemi genel yapısına göre; matrislerden oluşan görüntü, girdi olarak alındıktan sonra ilk olarak görüntü üzerine, görüntünün özelliklerinin çıkarılmasını sağlamak için filtre uygulanması gerekir, uygulanacak olan filtrenin önce x ve y eksenine göre simetrisi alınır, ikinci olarak görüntü matrisi ile filtre matrisinin birbirine denk gelen kısımlarındaki tüm değerler eleman eleman çarpılır, üçüncü olarak ise çarpılan değerlerin toplamı çıkış matrisinin ilgili elemanı olarak kaydedilir (Kızrak & Bolat, 2018).

$$y[m,n] = \sum_j \sum_i x[i,j].h[m-i,n-j] \quad (2.2)$$

formülü evrişim katmanındaki evrişim işleminin uygulanmasında kullanılan formüldür. Şekil 2.10'da örneği verilen evrişim işlemlerinden özellik çıkarma ile görüntüye ait kenar bilgisi gibi özelliklerin yani görüntü üzerindeki temel bilgilerin elde edilmesi sağlanmaktadır.



Şekil 2.10 Lena görüntüsü kenar bulma işlemi için özellik çıkarma örneği (Anonim a, 2021).

Giriş matrisinin üzerinde filtre gezdirilerek hesaplanan çıkış matrisi giriş matrisinden daha küçük boyutta olur. Çıkış matrisinin boyutunu hesaplamak için;

$$\ç = (n - f + 1) \times (n - f + 1) \quad (2.3)$$

formülü kullanılabilir (Kızrak & Bolat, 2018). Formülde yer alan $\ç$ çıkış matrisini, n giriş matrisini, f filtre matrisini ifade etmektedir. İlgili formül giriş matrisi üzerinde belirli bir piksel ekleme ve adım kaydırma oranı belirlemediğimizde çıkış matrisinin boyutunu bulmamıza yardımcı olan bir formüldür. Piksel ekleme ve adım kaydırma işlemlerini de ele aldığımızda çıkış matrisinin boyutunu bulduğumuz formül de farklılaşmaktadır. Giriş ve çıkış matrislerinin boyutlarının aynı olmasının istendiği durumlar söz konusu olabilmektedir. Giriş matrisi ile çıkış matrisinin boyut farkını yönetmek için dolgulama olarak da isimlendirilen piksel ekleme işlemi gerçekleştirilir (Kızrak, 2018). Piksel ekleme işlemi kenarlara sıfır ekleme yöntemi ile gerçekleştirilmektedir. Giriş matrisi ve çıkış matrisinin boyutunun eşit olmasının istendiği durumlarda, giriş matrisine eklenecek piksel değeri;

$$p = \frac{(f-1)}{2} \quad (2.4)$$

formülü ile belirlenmektedir. Formülde yer alan p eklenecek piksel değerini, f ise filtre matrisini ifade etmektedir. Bu durumda çıkış matrisinin boyutunun;

$$\zeta = (n + 2p - f + 1)x(n + 2p - f + 1) \quad (2.5)$$

formülü kullanılarak hesaplanması gerekmektedir (Kızrak, 2018). Formüldeki ζ çıkış matrisini, n giriş matrisini, f filtre matrisini, p ise eklenecek pikseli ifade etmektedir.

1	5	3	2	1	6
4	8	6	9	8	7
3	5	7	1	8	3
5	2	9	7	2	4
1	1	3	2	6	9
2	3	6	8	5	3

Giriş Matrisi

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	5	3	2	1	6	0	0
0	0	4	8	6	9	8	7	0	0
0	0	3	5	7	1	8	3	0	0
0	0	5	2	9	7	2	4	0	0
0	0	1	1	3	2	6	9	0	0
0	0	2	3	6	8	5	3	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Şekil 2.11 Kenarlara sıfır ekleme yöntemi ile piksel ekleme örneği (Setiawan ve ark., 2019)

Şekil 2.11 kenarlara sıfır ekleme yöntemine örnek teşkil eden bir giriş matrisini temsil etmektedir. Evrişim işlemlerinden kaydırma adımı, görüntü matrisi üzerinde filtre matrisini kaçır adım kaydırabileceğimizi planlamamıza yardımcı olmaktadır. Çıkış matrisinin boyutunu doğrudan etkileyen önemli bir parametredir (Kızrak, 2018). Kaydırma adımı 1'den farklı bir değer aldığında çıkış matrisinin boyutu;

$$\zeta = \left\lceil \frac{(n+2p-f)}{s} + 1 \right\rceil \times \left\lceil \frac{(n+2p-f)}{s} + 1 \right\rceil \quad (2.6)$$

formülü ile hesaplanmaktadır. Formüldeki ζ çıkış matrisini, n giriş matrisini, f filtre matrisini, p eklenecek pikseli, s ise kaydırma adımını ifade etmektedir. Kaydırma adımı değerinin artması çıkış matrisinin boyutunun küçülmesine neden olmaktadır. Bir görüntünün giriş matrisi ile çıkış matrisinin boyutunun eşit olması isteniyorsa piksel ekleme miktarı ile kaydırma adımı miktarının verilmesine dikkat edilmesi gerekmektedir (Kızrak & Bolat). Yani giriş matrisi ve çıkış matrisinin boyutunun eşit olması isteniyorsa, kaydırma adımı değerinin seçildiği büyüklüğe göre piksel ekleme değerinin de büyük seçilmesi gerekmektedir.

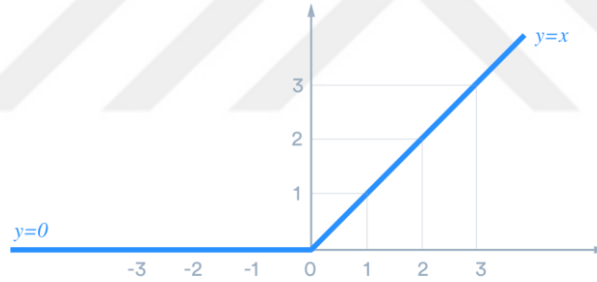
Relu Katmanı

Aktivasyon fonksiyonu, sinir ağına gerçek hayatta doğrusal olmayan görüntü, video, ses ve yazı gibi karmaşık dünya verilerini tanıtmak için kullanılmaktadır. Sinir ağında aktivasyon fonksiyonu kullanımı tercih edilmezse sinir ağı sınırlı bir öğrenme gücüne sahip olur (Kızrak, 2019).

Düzeltilmiş Doğrusal Birim (Relu), ESA’ da en yaygın, aslen ESA için varsayılan olarak kullanılan bir aktivasyon fonksiyonudur (Brownlee, 2019). Relu fonksiyonu ile yapılan aktivasyon işlemi sonucu negatif olarak elde edilen değerler 0 değerini alır (Liu, 2017). Negatif ekseninde 0 değerini alması çok miktarda işlem gerektiren büyük boyutlu sinir ağlarında hesaplamanın hızlı olmasına imkân tanımaktadır (Brownlee, 2019). Bu imkân ile ağda bulunan bazı nöronların 0 değerine çekildikleri için aktif olmayıp ve bazı nöronlar ise aktif şekilde seyreklik sağlayarak verimli bir hesaplama yükü oluşturmaktadır (Kızrak, 2019). Relu aktivasyon fonksiyonu;

$$f(x) = \max(0, x) \quad (2.7)$$

şeklinde ifade edilmektedir. Denklem (2.7)’de görüldüğü gibi aktivasyon fonksiyonuna verilen girdi değeri negatif bir değer ise çıktı değeri maksimum değer 0 olarak alınırken, pozitif bir değer ise maksimum değer olan kendi değerini alır (İbrahim, 2020). Şekil 2.12 Relu aktivasyon fonksiyonun grafiğini göstermektedir.

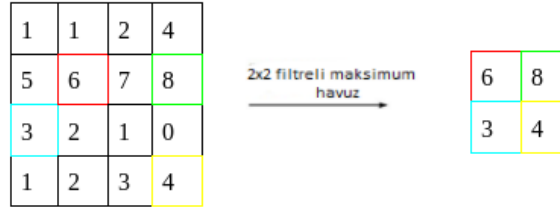


Şekil 2.12 Relu aktivasyon fonksiyonu grafiği (Liu, 2017)

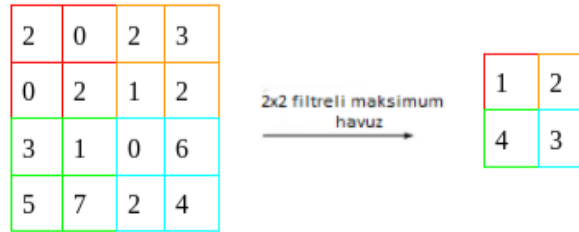
Ortaklama Katmanı

Ortaklama diğer adıyla havuzlama katmanının görevi ağdaki verileri indirgeyerek ağda dolaşacak veri miktarını azaltmak ve bu sayede ağın hem daha hızlı olmasını hem de ağdaki hesaplama miktarları ile kullanılacak bellek miktarının azalmasını sağlamaktır (Doğan & Türkoğlu, 2018). Genellikle Relu katmanından sonra kullanılmaktadır. Ortaklama katmanında bir filtre matrisi, Relu katmanından gelen ortaklama katmanına girdi olarak verilen matris üzerinde gezdirilir. Gezdirme sırasında elde edilecek çıkış değeri iki şekilde bulunmaktadır: Ortalama havuzlama veya en büyük değer havuzlama yöntemi. Ortalama havuzlama yönteminde Şekil 2.13’te gösterildiği gibi matris üzerinde gezdirilen filtre matrisinin içine denk gelen değerlerin hepsinin ortalaması

alınarak yeni matrisin değeri elde edilir. En büyük değer havuzlama yönteminde ise Şekil 2.14’de görüldüğü üzere filtre matrisinin denk geldiği değerlerden en büyük olan yeni matrise çıkış değeri olarak alınmaktadır.



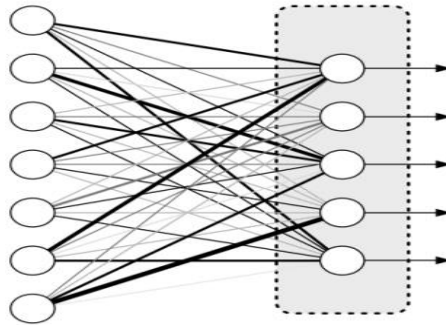
Şekil 2.13 En büyük değer havuzlama yöntemi örneği (Kın, 2019)



Şekil 2.14 Ortalama değer havuzlama yöntemi örneği (Kın, 2019)

Tam Bağlı Katman

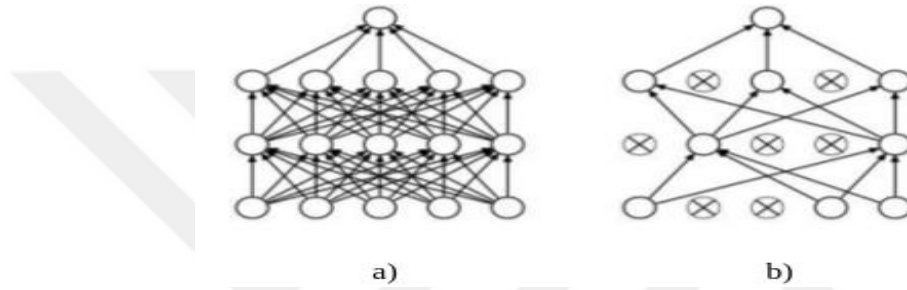
Her nöron kendinden sonra gelen nörona bağlantılı olduğundan tam bağlı olarak isimlendirilmektedir. Ortaklama katmanından sonra kullanılmaktadır (İbrahim, 2020). Kullanılan mimariye göre bu katmanın sayısı değişiklik gösterebilir. Tam bağlı katmanda bir önceki katmanda açığa çıkmış özellikler incelenerek nesnenin özelliklerini belirten ağırlıkların bulunduğu nöronlar tespit edilerek nesnenin hangi sınıfa ait olduğunun ortaya çıkması sağlanmış olur (Doğan & Türkoğlu, 2019). Şekil 2.15’te tam bağlı katman yapısı gösterilmiştir.



Şekil 2.15 Tam bağlı katman (Kın,2019)

Dropout Katmanı

ESA uygulamalarında ezberleme veya aşırı öğrenme diye isimlendirilen ağı öğrenmek yerine ezbere hareket ettiği durumlarla karşılaşılabilir. Büyük ağı eğitmek çok fazla hesaplama ve büyük miktarda eğitim verisi gerektirir. Bu verilerin farklı alt kümeleri üzerinde farklı ağı eğitmek için yeterli veri olmayabilir. Oluşturulan mimarilerde genellikle bu sorunların çözümünü sağlamak için dropout kullanılmaktadır (Srivastava ve ark., 2014). Dropout katmanındaki temel mantık ağdaki bazı düğümlerin kaldırılmasıdır (İnik & Ülker, 2017).



Şekil 2.16 a) Dropout olmayan standart ağ görünümü, b) Dropout kullanılan ağ görünümü (Uysal ve ark., 2018)

Şekil 2.16’da dropout kullanılan ve kullanılmayan ağlara örnek verilmiştir. ESA’da, dropout tekniğinin kullanımı tam bağlı katmanlarda gerçekleştirilirken, diğer katmanlarda kullanımı uygun değildir (Ergin, 2021). Tam bağlı bir katman, parametrelerin birçoğunu işgal ettiğinden eğitim sırasında her bir nöronun gücünü sınırlar ve eğitim verilerinin aşırı uymasına yol açar. Aşırı uymanın önüne geçmek için tam bağlı katmanlarda dropout kullanımı tercih edilmektedir (Budhiraja, 2021).

Sınıflandırma Katmanı

Sınıflandırma katmanındaki nesne sayısı ile çıkış katmanındaki nesne sayısı aynıdır (Uysal ve ark., 2018), yani sınıflandırma katmanında sınıflandırmak istenilen sayıda nesne bulunmaktadır. Sınıflandırma katmanında en sık tercih edilen sınıflandırıcı softmax sınıflandırıcısı (İnik & Ülker, 2017) olması sebebiyle katmanın adı softmax (yumuşatma) katmanı olarak da isimlendirilebilir. Softmax sınıflandırıcısı kendisinden önceki katmandan girdiyi alır ve ağdaki olasılıksal hesaplamaları gerçekleştirerek girdinin hangi sınıfa ait olduğunu çıktı olarak ortaya koyar (Setiawan ve ark., 2019).

Çıkış Katmanı

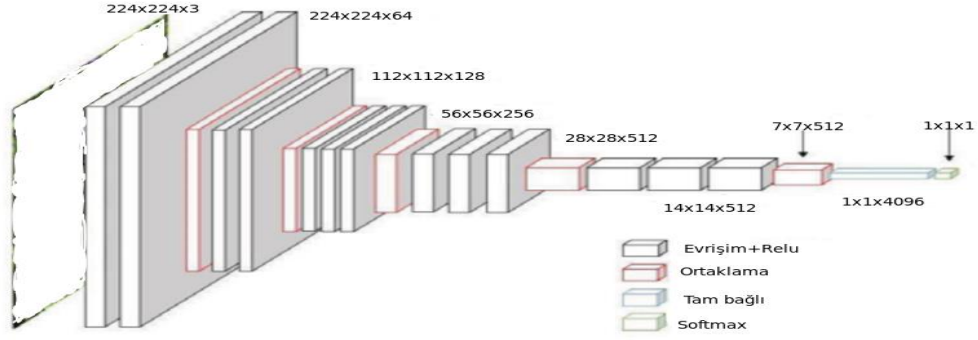
ESA'nın son katmanı olması ile birlikte ağın çıktısının verildiği katmandır. Sınıflandırma katmanındaki ile aynı sayıda eleman bulundurması nedeniyle aslında sınıflandırma katmanı da çıktı katmanı sayılabilmektedir.

2.2.3 VGGNet mimarisi

VGGNet mimarisi Simonyan ve Zisserman tarafından 2014 yılında geliştirilen ve ImageNet yarışmasında iyi bir başarı sergilemiş (%7.3 hata oranı) bir derin öğrenme modelidir (Simonyan & Zisserman, 2015). Çalışmada kullanımı tercih edilen VGGNET'in VGG16 modelinde her bir evrişim katmanının veya tam bağlı katmanın bir blok oluşturduğu modüler bir yapı mevcuttur. Modüllerdeki havuzlama katmanı özellik haritasının boyutunu küçültmek için kullanılmaktadır (Tao ve ark., 2021). Modelin asıl amacı, ağ karmaşıklığını artırmadan uygun katman derinliği ayarlarını dikkate alan bir model tasarlamaktır. Şekil 2.17'de VGGNet'e ait model yapısı, Şekil 2.18'de ise tam bir VGG16 mimarisi gösterilmektedir.

VGG Konfigürasyonları					
A	A-LRN	B	C	D	E
11 Ağırlık Katmanı	11 Ağırlık Katmanı	13 Ağırlık Katmanı	16 Ağırlık Katmanı	16 Ağırlık Katmanı	19 Ağırlık Katmanı
Girdi (224 x 224 RGB Resim)					
evriş3-64	evriş3-64 LRN	evriş3-64 evriş3-64	evriş3-64 evriş3-64	evriş3-64 evriş3-64	evriş3-64 evriş3-64
Havuzlama					
evriş3-128	evriş3-128	evriş3-128 evriş3-128	evriş3-128 evriş3-128	evriş3-128 evriş3-128	evriş3-128 evriş3-128
Havuzlama					
evriş3-256 evriş3-256	evriş3-256 evriş3-256	evriş3-256 evriş3-256	evriş3-256 evriş3-256 evriş1-256	evriş3-256 evriş3-256 evriş3-256	evriş3-256 evriş3-256 evriş3-256 evriş3-256
Havuzlama					
evriş3-512 evriş3-512	evriş3-512 evriş3-512	evriş3-512 evriş3-512	evriş3-512 evriş3-512 evriş1-512	evriş3-512 evriş3-512 evriş3-512	evriş3-512 evriş3-512 evriş3-512 evriş3-512
Havuzlama					
evriş3-512 evriş3-512	evriş3-512 evriş3-512	evriş3-512 evriş3-512	evriş3-512 evriş3-512 evriş1-512	evriş3-512 evriş3-512 evriş3-512	evriş3-512 evriş3-512 evriş3-512 evriş3-512
Havuzlama					
TB-4096					
TB-4096					
TB-1000					
softmax					

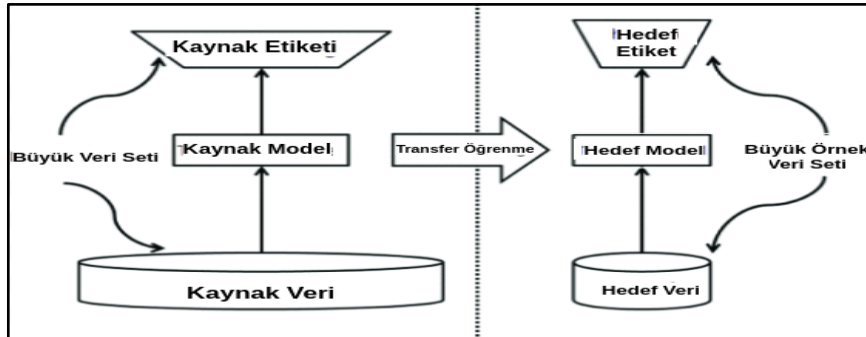
Şekil 2.17 VGGNet model yapısı (Simonyan & Zisserman, 2015)



Şekil 2.18 Tam bir VGG16 mimarisi (Hu, 2020)

2.2.4 Transfer öğrenme

Transfer öğrenme kavramı, elde edilen bilgilerin yeni şeylerin öğrenilmesine aktarılabilmesi yönüyle insan öğrenme davranışına benzemektedir (Fan, 2021). Bireyler bir problemle karşılaştıklarında o problemin çözümüne en hızlı ve uygun şekilde ulaşmayı hedeflerler. Bireylerin problemlere ürettikleri çözüm onların tecrübe kazanmasını ve benzer bir problemlerle karşılaştıklarında bu tecrübeyi kullanmalarını sağlar. Aynı mantıkla düşünüldüğünde transfer öğrenme geçmişte birbirine benzer problemlerin çözümü için hazırlanmış kalıpların yeni karşılaşılmış sorunlar için iyi performans ve hızlı sonuca ulaşmak amacıyla kullanıldığı bir yöntem olarak tanımlanabilir (Fırıldak & Talu, 2019). Şekil 2.19’da temel yapısı verilen transfer öğrenme, yapılacak olan tahmin işlevinin öğrenme performansını iyileştirmeyi amaçlamaktadır (Phan, 2020). Transfer öğrenme yönteminde, önceden eğitilmiş bir model, büyük ve karmaşık verileri eğitmek için yoğun katmanlar veya son katman dondurularak yeniden kullanılmaktadır (Rajeswari, 2021).



Şekil 2.19 Transfer öğrenmenin temelleri. (Li ve ark., 2021)

Mevcut sınırlı veriler ile başarılı bir ESA modeli kurmak kolay değildir. Bunun yerine, büyük bir veri seti üzerinde iyi sonuçlar veren bir model eğitmek ve daha sonra bu

eğitilen modeli sınırlı veriye sahip modelde uygulamak gerekir. Günümüzde birçok çalışmada kullanılan transfer öğrenme yöntemleri, sınırlı eğitim verisine sahip derin öğrenme alanındaki çalışmalarda önemli ilerlemeler kaydetmiştir. Transfer öğrenme yöntemi kullanılarak oluşturulan model, yeni bir kategoriye öğrenirken, daha önce öğrenilen kategorilere ait özelliklerini birleştirmekte ve yeni kategoriye ait az sayıda örneği kullanarak yeni grubun eğitim sürecini doğru bir şekilde tanımlayabilmektedir (Li ve ark., 2021).

2.2.5 Optimizasyon algoritmaları

Optimizasyon algoritmaları derin öğrenme mimarilerinde kullanılan önemli parametrelerden biridir. Optimizasyon algoritmasının performansı, modelin eğitim verimliliğini doğrudan etkilemektedir. Literatürde sıklıkla karşılaşılan optimizasyon algoritmaları RMSProp, Adam, Adagrad, Adadelta ve SGD optimizasyon algoritmalarıdır (Seyyarer ve ark., 2020). Çizelge 2.4’de optimizasyon algoritmalarına ilişkin açıklamalar yer almaktadır.

Çizelge 2.4 Çalışmada yer alan optimizasyon algoritmalarının açıklamaları ve formülleri

Optimizasyon Algoritması	Formül ¹ (Ser & Bati, 2019)	Açıklama
RMSProp	$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2$ $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$ $g_t = \nabla_{\theta_t} J(\theta_t)$	- Sabit bir şekilde artan öğrenme oranının etkisinden kurtulmak için geliştirilmiştir (Ruder, 2016). - Her parametre için öğrenme hızının hesaplanmasında kısmi türevlerin azalan ortalamasını kullanmaktadır. - Öğrenme hızı hesaplama açısından bakıldığında Adagrad’ın bir uzantısı olarak düşünülebilir. - Derin öğrenme ağları için hem etkili hem de pratik kabul edilen bir optimizasyon algoritmasıdır (Brownlee, 2021a).

Çizelge 2.4 (Devam) Çalışmada yer alan optimizasyon algoritmalarının açıklamaları ve formülleri

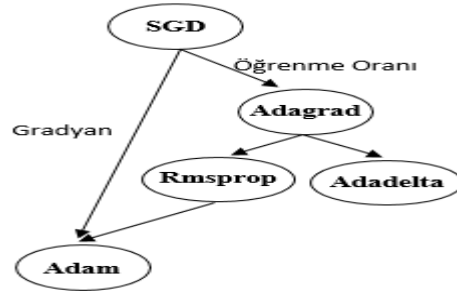
Adagrad	$g_{t,i} = \nabla_{\theta_t} J(\theta_{t,i})$ $\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} g_{t,i}$	- Sahip öğrenme oranı sorununu çözmek için geliştirilmiştir. Her bir adım için farklı bir öğrenme hızı kullanmaktadır (Ruder, 2016). - Öğrenme oranı giderek azalmakta ve belirli bir süre sonra öğrenme durmaktadır. Bu bir dezavantajdır. Çünkü uygun sonucun bulunmasından çok önce durma gerçekleşmiş olabilir (Çarkacı, 2018). - Arama sırasında önce o ana kadar görülen parametrenin kısmi türevlerini toplayarak, ardından ilk öğrenme oranı hiperparametresini, karesi alınan kısmi türevlerin toplamının kareköküne bölerek her parametre için adım boyutunu hesaplamaktadır (Brownlee, 2021b).
Adam	$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$ $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ $m'_t = \frac{m_t}{1 - \beta_1^t}, v'_t = \frac{v_t}{1 - \beta_2^t}$ $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v'_t + \epsilon}} m'_t$	- Uygulamalarda tercih edilmesinin temel sebepleri arasında; - Uygulanmasının kolay olması, - Bellek gereksiniminin küçük olması, - Sezgisel olması ile çok az ayarlama gerektirmesi, - Gürültülü veya seyrek eğilimli problemler için uygun olması - yer almaktadır (Brownlee, 2017). - Adagrad ve RMSProp optimizasyon algoritmalarının en iyi özellikleri birleştirilmeye çalışılmıştır (Kurt, 2018). - Her bir giriş parametresi için adım boyutu hesaplanmaktadır. Her bir adım boyutu, her bir değişken için karşılaşılan kısmi türevlere dayalı olarak arama süresi boyunca otomatik uyarlanmaktadır (Brownlee, 2021c).

Çizelge 2.4 (Devam) Çalışmada yer alan optimizasyon algoritmalarının açıklamaları ve formülleri

Adadelta	$E[g^2]_t = \rho E[g^2]_{t-1} + (1 - \rho)g_t^2$ $RMS[g]_t = \sqrt{E[g^2]_t + \epsilon}$ $\nabla x_t = \frac{\eta}{RMS[g]_t} g_t$	- Adagrad ve RMSProp'dan farklı olarak öğrenme oranı belirlemek zorunlu değildir. Manuel olarak belirlenmiş bir başlangıç öğrenme oranına ihtiyacı ortadan kaldırmak için tasarlanmıştır (Brownlee, 2021ç). - RMSProp optimizasyon algoritmasının yakınsamasını iyileştirmek için geliştirilmiştir.
SGD	$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} J(\theta_t; x^{(i)}; y^{(i)})$	- ESA'da varsayılan algoritmadır. - Bir eğitim veri setinde tahmine dayalı modelin kayıp fonksiyonunu en aza indirmeyi hedeflemektedir. - Diğer optimizasyon algoritmalarına kıyasla daha yavaş çalışmaktadır. Resim tanıma ile ilgili uygulamalarda iyi sonuç vermediği bilinmektedir (Çarkacı, 2018). - Optimizasyon sürecini iyileştirmek için tasarlanmıştır. RMSProp, AdaGrad, Adam ve AdaDelta'nın SGD'nın bir uzantısı olduğu söylenebilir.

¹ $\theta \in R^d$: Model parametreleri; η : öğrenme katsayısı; $\nabla_{\theta} J(\theta_t; x^{(i)}; y^{(i)})$: hedef fonksiyon eğimi; $G_{t,ii}$: t. iterasyona hesaplanmış eğim değerlerinin kareleri toplamı; ϵ : öğrenme katsayısının 0'a bölünmemesini sağlayan sabit değer, varsayılan 10^{-8} ; v_t : geçmiş eğimlerin karelerinin ağırlıklandırılmış ortalaması; m_t : Momentum değişiklikleri; β_1 : varsayılan 0.9; β_2 : varsayılan 0.999; ρ : bozunma sabiti; Δx_t : parametre güncellemesi; $RMS[g]_t$: değerlerin aritmetik ortalamasının karelerinin karekökü

Optimizasyon algoritmaları genel olarak hep bir öncekinde yaşanan sorunların önüne geçmek için çıkarılmıştır. Bazısı öğrenme oranında yaşanan belirli bir süre sonra durma problemlerini çözmek için bazısı ise gradyan yani eğim oranında yaşanan sorunları çözmek için geliştirmiştir (Ser & Bati, 2019). Şekil 2.20'de optimizasyon algoritmalarının evrimsel haritası verilmektedir.



Şekil 2.20 Optimizasyon algoritmalarının evrimsel haritası (Seyyarer ve ark., 2020)

2.2.6 Başarım ölçütleri

Bir modelin performansını değerlendirmek yapılan çalışmanın en önemli kısımlarından biridir. Model performansını ölçmek için kullanılan çeşitli başarım ölçütleri bulunmaktadır.

Karışıklık Matrisi

Gerçek değerler ve tahmin edilen değerlerin daha net görülebildiği matrislerdir. Matrisin sol tarafı gerçek değerleri verirken üst taraf tahmin edilen değerleri vermektedir. Her sınıf için hangi sınıfın gerçekte ne olması gerektiği fakat ne olarak tahmin edildiğine ilişkin bilgiler matris üzerinden alınabilmektedir. Çizelge 2.5’ de ikili sınıflandırma için gerçek ve tahmin değerlerini gösteren bir karışıklık matrisi yer almaktadır. Bu matristeki;

- Doğru Pozitifler(DP), sınıflandırıcı tarafından pozitif olarak sınıflandırılmış ve gerçek değerleri de pozitif olan değerleri,
- Yanlış Pozitifler(YP), sınıflandırıcı tarafından pozitif olarak sınıflandırılmamış ancak gerçek değerleri negatif olan değerleri,
- Doğru Negatifler(DN), sınıflandırıcı tarafından negatif olarak sınıflandırılmış ve gerçek değerleri de negatif olan değerleri,
- Yanlış Negatifler(YN), sınıflandırıcı tarafından negatif olarak sınıflandırılmamış ancak gerçek değerleri pozitif olan değerleri

ifade etmektedir. DP ve DN tahminin doğru yapıldığını, YP ve YN ise tahminin yanlış yapıldığını belirtmektedir (Kutlu,2019).

Çizelge 2.5 İkili sınıflandırma için karışıklık matrisi yapısı (Öğündür, 2019)

Gerçek	Tahmin	
	Doğru Pozitifler(DP)	Yanlış Negatifler(YN)
	Yanlış Pozitifler(YP)	Doğru Negatifler(DN)

Doğruluk

Tüm sınıflar için hesaplanan performans ölçüsüdür. Doğru tahmin edilen sınıfların toplam örnek sayısına oranı şeklinde hesaplanmaktadır (Öğündür, 2019). 0 ile 1 arasında değer almaktadır (Korkmaz, 2019).

$$Doğruluk = \frac{DP+DN}{DP+DN+YP+YN} \quad (2.8)$$

Kesinlik

Pozitif tahmin değeri olarak da bilinmektedir. Pozitif sınıflandırmaların ne kadarının gerçekten doğru olduğunu tahmin etmeye yarayan, gerçek pozitiflerin pozitif olarak sınıflandırılanlara bölünmesi ile elde edilen, 0 ile 1 arasında değer alan ölçüttür (Korkmaz, 2019; Öğündür, 2019).

$$Kesinlik = \frac{DP}{DP+YP} \quad (2.9)$$

Duyarlılık

Gerçek pozitiflerin ne kadarının doğru tahmin edildiğini bulmamıza yarar. Yani pozitif tahmin edilmesi gerekenlerin ne kadarının pozitif tahmin edildiğini gösteren, tüm pozitifler arasındaki gerçek pozitiflerin oranı veren (Aktaş, 2020), 0 ile 1 arasında değişen ölçüttür (Korkmaz, 2019).

$$Duyarlılık = \frac{DP}{DP+YN} \quad (2.10)$$

F1 Skor

Kesinlik ve duyarlılık değerlerinin harmonik ortalaması F1 skoru vermektedir (Aktaş, 2020). Eşit dağılım olmayan veri setlerinde daha doğru değerlendirmeler yapabilmek için doğruluk değerinin yerine kullanımı tavsiye edilmektedir (Öğündür, 2019). Hem

yanlış negatifleri hem de yanlış pozitifleri hesaba katan, 0 ile 1 arasında değişen bir ölçüttür (Korkmaz, 2019).

$$F1skor = 2x \frac{KesinlikxDuyarluluk}{Kesinlik+Duyarluluk} \quad (2.11)$$

Makro Ortalama

Makro ortalama, çok sınıflı sınıflandırmada kesinlik, duyarlılık ve F1 skor gibi puanlama ölçütlerini değerlendirmek için kullanılan ortalama alma yöntemidir. Sınıflandırıcının genel performansı değerlendirilmek istendiğinde kullanılmaktadır (Kumar, 2020).

Ağırlıklı Ortalama

Kesinlik, Duyarlılık ve F1 skor ile ilgili olarak çok sınıflı sınıflandırmada kullanılan ortalama alma yöntemlerinden biridir (M, 2019).

3. LİTERATÜR ÖZETLERİ

Görüntü sağlıktan astronomiye, coğrafi bilimlerden askeri uygulamalara, günlük yaşamdan uydu görüntülerine kadar hayatımızın ayrılmaz bir parçası haline gelmiştir. Görüntünün hayatımızdaki yerinin ve işlevinin artması ile birlikte ışık, sıcaklık gibi farklı etkenlerden dolayı görüntülerde oluşan gürültüler de literatürdeki çalışma alanlarından biri haline gelmiştir. Gürültüsüz görüntülerin önemi artmış, bu da beraberinde görüntülerindeki gürültülerin doğru şekilde tahmin edilmesi ve ortadan kaldırılması için farklı çalışmaların uygulanmasına neden olmuştur. Literatür incelendiğinde gürültü ile ilgili olan farklı çalışmalar aşağıda kısaca özetlenmiştir:

Immerkær (1996), görüntüdeki Gauss gürültüsünün varyansını tahmin etmek için sunduğu yöntemde hızlı ve basit bir şekilde görüntüdeki gürültü varyansı çok çeşitli gürültü varyans değerleri için performansını göstermiştir. Sunulan yöntem, 20'ye kadar ki varyans değeri için ideal performans gösterirken, 50 varyans değerinde bozulmaya başlar. Bunun yanı sıra yöntem, dokusu fazla olan görüntülerde veya bölgelerde, görüntüdeki ince çizgileri gürültü olarak algılamaktadır.

Hamza ve Krim (2001) sağlam tahmin teorisine dayanan doğrusal olmayan filtreleme tekniklerinden ortalama-serbest medyan, ortalama-LogCauchy ve serbest medyayı tanıtmaktadır. Parametrelerin önerilen filtreler üzerindeki etkileri analiz edilerek belirtilen teknikler ile gürültü düzeltme işleminde başarı elde edilmesi beklenmiştir. Çıkan sonuçlar değerlendirilmiş ve önerilen gürültü azaltma yöntemlerinin diğer filtreleme tekniklerine göre gürültü düzeltme de çok daha gelişmiş bir performans gösterdiğinden bahsedilmiştir.

Tai ve Yang (2008) görüntü gürültüsü tahmini için basit ve hızlı bir algoritma önermiştir. Giriş görüntüsünün Gauss gürültüsü tarafından bozulduğu varsayılır. Yapıları veya ayrıntıları gürültü varyans tahminine dahil etmemek için, ilk olarak birinci dereceden gradyanları kullanan basit bir kenar algılama algoritması uygulanır. Daha sonra bir Laplacian operatörü ve ardından tüm görüntünün ortalamasının alınması sağlanır. Bu yöntem ile çok doğru gürültü varyansı tahmini sağlanması

hedeflenmektedir. Simülasyon sonuçları, önerilen algoritmanın çok çeşitli gürültü varyanslarında farklı performans sergilediğini göstermektedir. Kullanılan Lena görüntüsü için tahmin hatası %50'den büyük, en kötü durum olan Mandrill görüntüsü için de %400'den fazla olduğu görülmüştür. Yöntem yüksek seviyeli gürültüler için iyi performans gösterirken, gürültü seviyesi düşük olduğunda iyileştirmelere gidilmesinin uygun olacağı değerlendirilmektedir.

Jain ve Seung (2008) yüz adet doğal görüntü içeren bir veri seti kullanarak, görüntülerdeki gürültüleri gidermek için özel olarak tasarlanmış evrişimli ağların bir yaklaşımı olan Markov rastgele filtre yöntemini (MRF) önermektedir. MRF yaklaşımının hesaplama zorluklarından kaçınması, yüksek derecede temsil gücüne sahip olması görüntü işleme mimarilerini öğrenmeyi mümkün kıldığını göstermektedir.

Zoran ve Weiss (2009) doğal görüntülerin ölçekler boyunca sabit bir basıklık değerine sahip olması gerektiğini belirtmektedir. Doğal görüntülerdeki sapmaların görüntünün doğasında bulunan gürültülerden kaynaklandığını ve görüntüdeki gürültülerin tespit edilmesi ile doğal görüntülerdeki ölçekler boyunca görülebilecek basıklık sorununun önüne geçilebileceğini varsaymaktadır. Varsayımdan yola çıkarak yeni model önerilmiş ve bozulmuş doğal görüntülerdeki gürültü standart sapmasını tahmin etmek için kullanılmıştır. Sonuçlar, bozulmuş varsayılan doğal görüntülerinin aslında gürültülü olduğunu ve temizlenebilir olduğunu göstermektedir.

Burger ve ark. (2012) görüntüleri gürültüsüzleştirme için çok katmanlı algılayıcı (MLP) kullanmayı önermektedir. MLP yönteminin en son görüntü gürültü giderme yöntemleri ile rekabet edebilecek derecede başarılı olacağını ileri sürmektedir. MLP birçok açıdan değerlendirilmeye çalışılarak incelenmiştir. İlk olarak eğitim örneklerinin sayısına bağlı olarak gauss gürültüsü içeren iki görüntü teste verilmiş ve eğitim örnek sayısı fazla olan çalışmanın daha iyi sonuç verdiği gözlemlenmiştir. İkinci olarak MLP'lerin blok eşleştirme ve 3D filtreleme (BM3D) yönteminden daha iyi olup olmadığı gözlemlenmiş ve kullanılan veri setinde yer alan 500 görüntüden 300'ünde MLP 'nin BM3D'den daha iyi performans gösterdiği sonucu ile karşılaştırılmıştır. Üçüncü olarak MLP ile BM3D'nin farklı gürültü seviyelerinde verdikleri tepkiler incelenmiş ve yine MLP'nin BM3D'den

iyi olduđu gözlemlenmiştir. Bunun yanında farklı gürültü türleri ve eğitimin farklı gürültü seviyelerindeki veriler ile yapıldığı modeller gibi konulara bağılı deneyler yapılmış ve her birinde MLP'nin BM3D'den daha iyi olduđu görölmüştür.

Lahmire ve ark. (2014) varyasyonel mod ayrışımına (VMD), deneysel mod ayrıştırmasına (EMD) ve ayrık dalgacık dönüşümüne (DWT) dayalı üç biyomedikal görüntü gürültü tahmini tekniğini bir beyin manyetik rezonans görüntüsü (MRI), bir prostat doku görüntüsü ve bir retina dijital görüntüsünden oluşan veri seti üzerinde karşılaştırmıştır. EMD' nin ortalama işlem süresi beyin MRI görüntüsü için 6841s, prostat dokusu görüntüsü için 3659s ve retina dijital görüntüsü için 1212s; VMD'nin ortalama işlem süresi sırasıyla 4068s, 2744s ve 912s olarak tespit edilmiştir. Buna göre, VMD yaklaşımı EMD'den daha hızlı olduđu görölmektedir. Öte yandan, DWT eşikleme yaklaşımı ikisinden çok daha hızlı sonuca ulaşmış ve ortalama işlem süresinin yaklaşık iki saniye sürdüğü tespit edilmiştir. Tepe sinyal-gürültü oranına (PSNR) dayanan sonuçlar ise, VMD ve EMD yaklaşımlarının geleneksel DWT tabanlı eşikleme yaklaşımından daha iyi performans gösterdiğini ve VMD' nin genel olarak en iyi performansta olduğunu göstermektedir.

Magud ve ark. (2016) ultrasonik tıbbi görüntülerde yaygın olan gürültü tiplerinden benek gürültüsünün, noktaların bir pikselden daha büyük olduđu durumlar da dahil olmak üzere, giderilmesi için geliştirilmiş bir medyan filtresi önerilmektedir. Önerilen yöntemin başarısını test etmek için Ortalama Kare Hatası (MSE), PSNR ve benzeri metrikler kullanılmıştır. Standart Benchmark ultrason görüntüleri kullanılarak yapılan testlerde karın görüntüleri için çıkan PSNR 17.56, MSE 43.44, boyun görüntüsü için çıkan PSNR 31.75, MSE 136.01 olarak bulunmuştur. Sonuçlar önerilen yöntemin, ultrason görüntülerinden benek gürültüsünü başarılı bir şekilde gidermek için kullanılabileceği göstermiştir.

Zhang ve ark. (2017) görüntüden gürültü giderme için ileri beslemeli evrişimsel sinir ağlarının (DnCNN'ler) inşasını araştırarak yeni bir model önermiştir. DnCNN modeli, genellikle belirli bir gürültü seviyesinde mevcut ayırt edici gürültü azaltma modellerinden farklı olarak, bilinmeyen gürültü seviyesiyle gauss gürültüsünü

kaldırabilir. Öncelikle model üç farklı varyans değeriyle ($\sigma = 15, 25$ ve 50) eğitilir. Daha sonra yapılan kapsamlı deneylerin sonucu, DnCNN modelinin yalnızca birkaç genel görüntü gürültü tahmini görevinde yüksek etkinlik sergilemekle kalmayıp, aynı zamanda GPU hesaplamasından yararlanılarak verimli bir şekilde uygulanabileceğini göstermiştir.

Zhang ve ark. (2018) hızlı çıkarım ve iyi performans elde etmek amacıyla, görüntü giderme alanında kullanılan bazı yöntemlerin çoğunlukla her bir gürültü seviyesi için belirli bir model öğrendiğini ve farklı gürültü seviyelerine sahip görüntüleri gürültüden kurtarmak için birden fazla model gerektirdiğini belirtmiştir. Giriş olarak ayarlanabilir bir gürültü seviyesi haritasına sahip hızlı ve esnek bir ESA, yani FFDNet kullanımını önerilmiştir. Amacın gerçekleştirilebilmesi için eğitim sırasında girdi olarak gürültü seviyesi haritaları kullanılmıştır. BM3D, DnCNN ve FFDNet için yapılan bir karşılaştırma ile her birinin farklı durumlarda daha hızlı sonuçlar verdiği tespit edilmiştir. BM3D'nin renkli görüntülerdeki gürültünün tespiti için gri tonlamalı görüntülerden daha fazla zaman harcadığı, FFDNet'in ise hem renkli görüntüler hem de gri tonlamalı görüntülerdeki gürültünün tespiti için yaklaşık olarak aynı zamanı harcadığı ve FFDNet'in CPU ile çalışırken BM3D'den ve DnCNN'den yaklaşık üç kat daha hızlı olduğu; GPU ile çalışırken DnCNN'den çok daha hızlı olduğu görülmüştür. Sonuç olarak, yapılan çalışma FFDNet'in verimli olduğunu ve pratik gürültü tahmini uygulamaları için oldukça etkili olduğunu göstermektedir.

Güraksın (2018) görüntülerde sıklıkla karşılaşılan tuz-biber gürültüsünün giderilmesi için k-ortalama algoritması kullanılarak tasarlanan bir filtre önermektedir. Önerilen filtrenin performans karşılaştırması için standart gürültü giderme filtrelerinden olan ortalama ve ortanca filtreleri ile uyarlamalı ortanca filtre kullanılmıştır. Sonuçlarda yalnızca görüntülerdeki gürültü oranı 0.1 iken önerilen yöntemin üzerinde PSNR değeri, ortanca filtre ile uyarlamalı ortanca filtrede görülmüştür. Bunun yanında gürültü seviyesindeki artışın k-ortalama algoritması ile tasarlanan filtreleme yönteminde diğer üç filtreleme yöntemine kıyasla başarıyı artırdığı gözlemlenmiştir.

Değirmenci ve ark. (2018) görüntülerde tespit edilen tuz-biber görüntüsünün giderilmesi için uyarlamalı medyan filtresi ve görüntülerde tuz-biber gürültüsü olmayan yerler içinde anahtarlamalı Gauss filtresi uygulamayı önermektedir. Amaç değişik gürültü türlerine değişik filtreler uygulayarak görüntünün kenarlarının bulanıklaşmasının önüne geçmek ve görüntüyü gürültüden arındırmaktır. Çalışmada kullanılan görüntülere 0.01-0.14 aralığında gauss gürültüsü ve üzerine 0.01-0.14 aralığındaki farklı yoğunluk değerlerinde tuz-biber gürültüsü eklenmiştir. Görüntülere önce uyarlamalı medyan filtresi uygulanarak tuz-biber gürültüsünün görüntüden arındırılması sağlanmıştır. Daha sonra anahtarlamalı gauss, görüntüyü gauss gürültüsünden arındırmak için kullanılmıştır. Deneysel sonuçlar, kullanılan yöntem ile kenar piksellerinin bulanıklaşma miktarının azaltılabileceğini ve görüntülerdeki gürültülerin azaltılması için geleneksel yöntemlere oranla daha iyi sonuçlar verdiğini göstermektedir.

Kaur ve Gandhi (2019) beyin görüntülerinin sınıflandırılması için VGG16 ve transfer öğrenme kullanımını önermiştir. Amaç, patolojik beyin görüntü sınıflandırması için transfer öğrenme ile önceden eğitilmiş bir DCNN VGG16 modelinin kapasitesini araştırmaktır. %100 tanıma oranı sağlayan bu modelin dezavantajı eğitim süresinin çok uzun olmasıdır.

Sil ve ark. (2019) evrişimsel sinir ağlarının, görüntüden gürültü gidermek için kullanılan diğer algoritmalara göre nasıl performans gösterdiğini kıyaslamak için VGG16 ve Inception-v3 algoritmalarını kullanmıştır. Sonuçlar Inception-v3 mimarisinin VGG16 mimarisinden gürültü sınıflandırma için daha iyi performans sergilediğini göstermektedir. 30 tur sonrasında Inception-v3 mimarisi, VGG16 mimarisinden daha iyi eğitim doğruluğu vermiş ve testte %1 oranında VGG16'dan daha fazla başarımlı sağlamıştır.

Mafi ve ark. (2019) görüntülerdeki dürtü- gauss gürültü kombinasyonlarının giderilmesi için kullanılan filtreler hakkında kapsamlı bir araştırma sunarak görüntü işleme ile ilgili ilerlemeleri içeren bilgiler vermiştir. Gürültü modellerini ve gürültü giderme filtrelerini

açıklayarak bunların türlerini ve uygulama alanlarına göre sınıflandırılmasını sunmuştur.

Tassano ve ark. (2019) evrişimsel sinir ağı mimarisine dayanan yeni görüntü gürültü giderme yöntemi olan FFDNet kullanmayı önermektedir. FFDNet için Adam optimizasyon algoritması kullanılmış ve eğitim 80 turda gerçekleştirilmiştir. Gri tonlu görüntüler ve renkli görüntülerin FFDNet gürültü gidericisi ile değerlendirilebilmesi için iki veri seti kullanılmıştır. Sonuçlara göre FFDNet'in her iki renk veri seti içinde benzer değerler verdiği gözlemlenmiştir. Yine artık öğrenme içerip içermemesine bağlı olarak FFDNet için iki ayrı mimari oluşturulmuş ve bu mimarilerden birinde artık öğrenme yapısı kullanılırken diğerinde düz ESA yapısı kullanılmıştır. Sonuçlar karşılaştırıldığında ise çok küçük başarı farklılıklarıyla artık öğrenme içeren FFDNet mimarisinin kullanılmasının uygun olduğuna karar verilmiştir. FFDNet'in diğer sinir ağı gürültü gidericilerine göre daha hızlı olduğunu gösterebilmek için bazı gürültü gidericiler kullanılmış ve bunlardan DnCNN diğerlerine kıyasla, FFDNet ile karşılaştırılabilir düzeyde sonuçlar vermiştir. Her bir modelin eğitim setini oluşturan eklerin miktarı DnCNN için 384000'den FFDNet için 1024000'e kadar çıkmakta ve turda artmaktadır. Bu parametreler, gürültünün güçlü olması halinde ve renklerdeki gürültüleri ele alırken FFDNet için performansın artmasına katkıda bulunmaktadır. FFDNet, DnCNN'den daha büyük bir alıcı alana sahip olması sebebiyle daha yüksek gürültü değerlerinin kaldırılmasını desteklemektedir. Sonuç olarak, FFDNet'in DnCNN'den daha hızlı ve daha etkili olduğu görülmüştür.

Uss ve ark. (2020) modele ve öğrenmeye dayalı iki tahmin ediciyi ele almaktadır. Modele dayalı, kesirli Brownian hareketi (fBm) alan parametrelerinin maksimum olabilirlik tahmin edicisini; öğrenmeye dayalı, gerçek yaşam görüntüleri üzerinde eğitilmiş NoiseNet evrişimli sinir ağını temsil eder. NoiseNet tahmincisi, fBm tahmincisine göre daha verimsiz sonuçlar göstermektedir.

Ponomarenko ve ark. (2020) görüntülerde yer alan gauss gürültüsünün görüntülerden arındırılarak orijinal gürültüsüz görüntünün kurtarılması için derin sinir ağları kullanılarak geliştirilen ESA modelleri ile bir çalışma gerçekleştirmektedir. Önerilen

yöntemlerin, BM3D filtresi ile birlikte kullanılan modelin mevcut çözümlerden daha verimli gürültü bastırma sağladığı görülürken, Genişletilmiş-Artık U-Net (DRUNet) gürültü giderici kullanılan modelin 0.2 saniyelik hesaplama süresi ile mevcut yöntemlerden daha iyi performans gösteren etkili ve hızlı gürültü bastırma sağladığı görülmektedir.

Küpeli ve Bulut (2020) Mean, Medyan ve Gauss geleneksel filtrelerini kullanarak literatürdeki “airplane.png, baboon.png, lena.png, peppers.png” görüntüleri ile gürültü giderme için test yapmıştır. Her filtrenin her görüntü için ve çalışmada kullanılan tuz-biber ve gauss gürültülerinin farklı gürültü oranları için PSNR değerlerini elde etmiştir. PSNR değerinin yüksekliği görüntünün kalitesini göstermektedir. Sonuçlara örnek olarak %1 oranında tuz-biber gürültülü Lena görüntüsü medyan filtresinde 33.7 PSNR değerindedir. Bu çalışma ile dijital fotoğraf ve videolardaki gürültülerin temizlenmesi hedeflenmektedir.

İbrahim (2020) ESA tabanlı bir gürültü giderme yöntemi önermiştir. Göğüs röntgen görüntüleri üzerinde test ettiği yöntemin PSNR değeri ortalama 28.100 elde edilmiştir.

Pimpalkhute ve ark. (2021) dijital görüntülerdeki aauss gürültüsünün DWT metodu ile ortadan kaldırılması ile oluşturulan gürültü tahmin modeli önermiştir. Sonuçlar, önerilen yöntemin, gürültü oranı arttıkça hata oranının 0.32’ye düşmesi ile diğer yöntemlere nazaran çok daha iyi bir değere sahip olduğunu göstermiştir.

Görüntüden gürültü arındırma işlemi yani filtre uygulaması gürültünün türüne göre gerçekleştirilmektedir. Gürültü türlerinin tespiti doğru ve pratik şekilde gerçekleştirilebilir ise görüntüden gürültü arındırma çalışmalarında kullanılan filtrelerin başarımlarını artabileceği düşünülmektedir. Bu nedenle görüntü çalışmalarına başarıları ile adını yazdırmış ESA ile oluşturulan yeni bir mimari aracılığıyla gürültü türünün doğru ve pratik bir şekilde hangi optimizasyon algoritmaları kullanılarak yapılabileceğine ilişkin bilgi verilmesi amaçlanmıştır.

4. MATERYAL ve YÖNTEM

Evrişimsel sinir ağları eğitilirken kullanılacak veri setinin amaca uygun verilerden oluşması gerekmektedir. Derin öğrenmede sınıflandırma işlemi yapılabilmesi için veri setleri oluşturulmakta ve bu veri setleri üzerinde test ve eğitim işlemleri gerçekleştirilmektedir. Veri seti uygun platformlar kullanılarak oluşturulduktan sonra seçilen derin öğrenme geliştirme ortamında amaca uygun oluşturulmuş mimariler ile eğitim ve test işlemleri yapılabilmektedir. Bu bölümde veri setinden ve yeni modelinin oluşturulmasından ayrıca kullanılan derin öğrenme geliştirme ortamından bahsedilecektir.

Çizelge 4.1 Çalışmada takip edilecek aşamalar

Adım 1:	Veri setinin oluşturulması
Adım 2:	Google Colab ve Google Drive entegrasyonunun sağlanması
Adım 3:	Eğitim ve test veri setlerinin Google Drive' a aktarılması
Adım 4:	Google Colab'ın Google Drive'dan eğitim ve test veri setlerine erişebilmesi için gerekli hazırlıkların yapılması
Adım 5:	Eğitim için ayrılmış veri setinin Google Colab'ta eğilmesi
Adım 6:	Test için ayrılmış veri setinin Google Colab'ta test edilmesi
Adım 7:	Herbir optimizasyon algoritması için eğitim ve test işlemlerinin gerçekleştirilmesi (4,5 ve 6 ncı adımlar)
Adım 8:	Çıkan sonuçların incelenerek karşılaştırılmasının yapılması

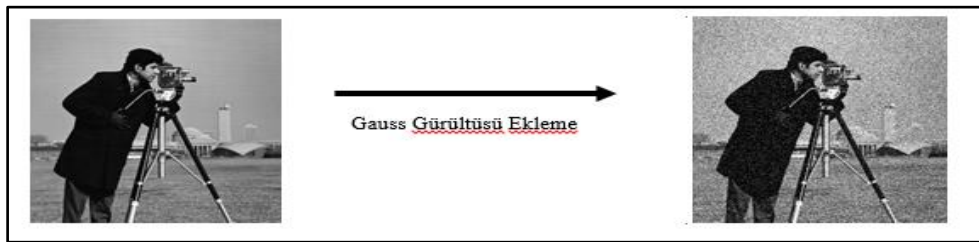
Çizelge 4.1'de çalışmaya ait izlenecek adımlar genel açıklamalar altında verilmiştir. Veri setinin oluşturulmasında görüntü işleme uygulamalarında sıkça kullanılan Matlab programlama dilinin kullanılması uygun görülmüştür. Çalışmada kullanılacak model, Oxford Üniversitesi'nde bir ekip tarafından tasarlanan VGG16 mimarisidir. Bu mimari önceden eğitilmiş bir model ve son 4 katmanı hariç diğerleri dondurularak üzerine 2 katman eklenmesi ile yeni bir model haline getirilmiştir. Eğitim ve test işlemleri için gerekli kodlar; Google'ın sunduğu, tarayıcı üzerinden Python programlama dili ile kod yazmamızı ve çalıştırmamızı sağlayan, ücretsiz GPU erişim imkanı sunan, özel bir

yapılandırma gerektirmeyen ve kolay paylaşım imkanı tanıyan Collaboratory arayüzü kullanılarak çalıştırılmıştır. Model 5 farklı optimizasyon algoritmasına ait doğruluk, F1 skor, kesinlik ve duyarlılık metriklerinin karşılaştırılması ile değerlendirilmektedir.

4.1 Veri Seti Oluşturulması

Çeşitli gürültü türlerine sahip görüntülerden oluşan veri setini elde edebilmek için kullanım kolaylığı ve pratikliği göz önüne alınarak Matlab (*Matrix Laboratory*) programlama dilinin kullanılmasına karar verilmiştir. Matlab; ilk olarak matris hesaplama dili olarak oluşturulmuş makine öğrenmesi, derin öğrenme ve veri bilimi gibi ileri düzey konularda da kullanılan dördüncü nesil bir programlama dilidir (Leblebicioğlu, 2019). Bu çalışmada Matlab, gürültülü görüntü elde etmek için kullanılacak olan bir platformdur.

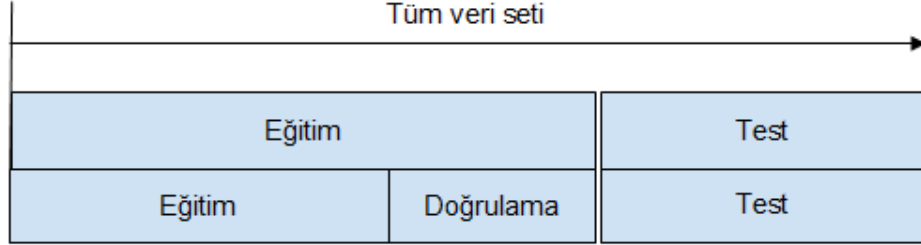
Orijinal görüntülere Matlab programlama dili aracılığıyla tuz-biber, gauss ve benek gürültüleri eklenerek gürültü türüne göre ayrı klasörde tutulan görüntülerden oluşan bir veri seti elde edilmiştir. Engin Akar, Sadık Kara ve arkadaşlarının görüntü filtreleme metotları ile ilgili yaptığı bir çalışmada Matlab programında hangi fonksiyonlar kullanılarak görüntüye gürültülerin eklendiğinden bahsedilmektedir (Akar ve ark., 2015). Şekil 4.1'de 0.01 gürültü seviyesinde gauss gürültüsünün Kameraman görüntüsüne eklenmesi ile oluşan gürültülü Kameraman görüntüsü, veri seti oluşturma aşamasına örnek olarak verilmiştir.



Şekil 4.1 Gürültü ekleme işlemine örnek

Veri setleri genel olarak eğitim seti, doğrulama seti ve test setinden oluşmaktadır. Eğitim seti, modelin öğrenmesi için kullanılacak seti, doğrulama seti modelin performansını iyileştirmeye yardımcı herhangi bir işlemin yapılıp yapılmayacağının tespitini sağlamak için kıyaslama amaçlı kullanılabilecek seti, test seti ise model

yeteneğini ölçmek için kullanılan seti ifade etmektedir. Doğrulama seti, eğitim veri seti içinden seçilmektedir. Şekil 4.2’de eğitim, doğrulama ve test veri seti arasındaki ilişki verilmiştir.

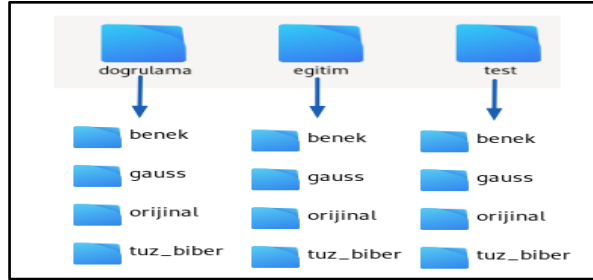


Şekil 4.2 Eğitim, doğrulama ve test seti arasındaki ilişki

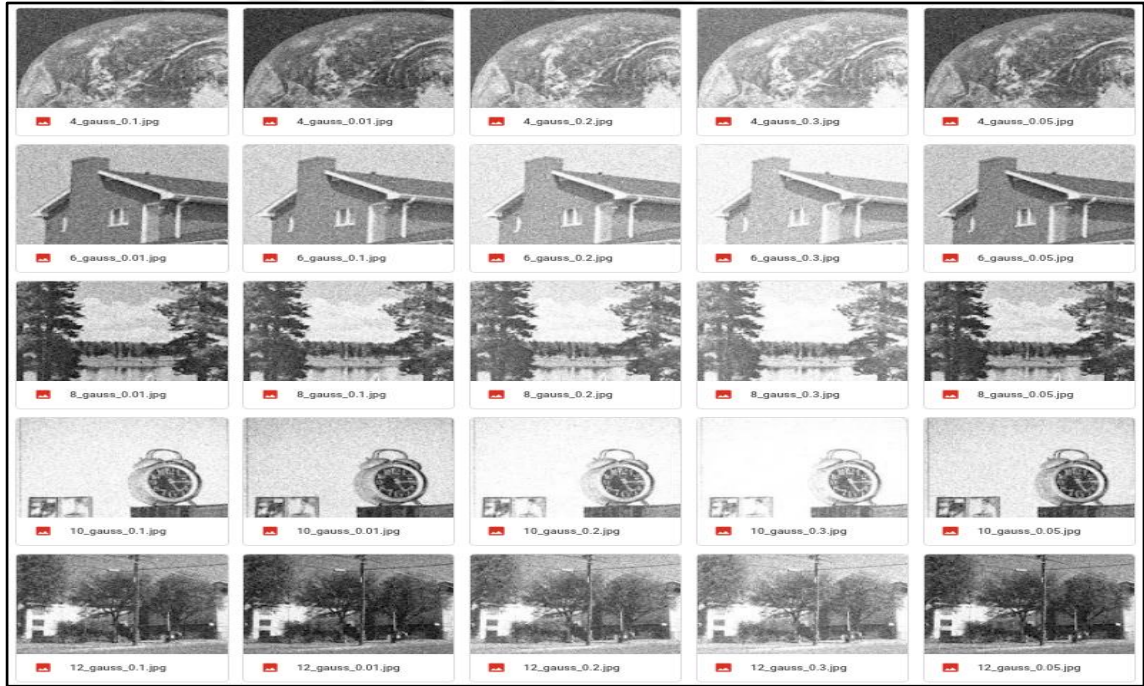
Veri seti eğitim ve test için toplamda 3840 görüntüden oluşmaktadır. Bu görüntülerden %70’lik kısım eğitim verisi iken geriye kalan % 30’luk kısım ise test verisidir. Eğitim için ayrılmış görüntülerin yer aldığı verilerde 180 orijinal, 900 gauss gürültülü, 900 tuz-biber gürültülü ve 900 benek gürültülü eğitim verisi bulunmaktadır. Test için ayrış görüntülerin yer aldığı verilerde ise 60 orijinal, 300 gauss gürültülü, 300 tuz-biber gürültülü ve 300 benek gürültülü test verisi yer almaktadır. Toplamda eğitim için 2880 görüntü, test için 960 görüntü veri setini oluşturmaktadır. Kullanılan veri setinde eğitim ve test verilerine ek doğrulama dediğimiz eğitim verisi için ayrılmış %70’lik kısmın içinden %25’lik kısım ile ön mini test amaçlı kullanılan doğrulama veri setide yer almaktadır. Bu veri setinin kullanım amacı ince ayar denilen uygulama için gerekli ayarların test aşamasına geçmeden daha doğru tespit edilmesidir. Bu durumda veri setinde yer alan 2880 eğitim verisinden 720 görüntü doğrulama setinin oluşturulması için de kullanılmıştır. Doğrulama veri setinde 225 gauss, 225 benek, 225 tuz-biber, 45 orijinal görüntü bulunmaktadır.

Veri seti hazırlanırken orijinal görüntülerin her birine görüntünün belirginliğini tamamen ortadan kaldırmayacak düzeyde seçilmiş belirli 5 farklı değerde (0.05-0.01-0.1-0.2-0.3) gauss, tuz-biber ve benek gürültüleri eklenerek eğitim ve test için gerekli gürültülü görüntüler elde edilmiştir. Her bir görüntü aynı forma indirgenmiş, siyah-beyaz, 256x256 boyutunda ve .jpg formu haline getirilmiştir. Şekil 4.3, veri setlerinin klasör yapısını göstermektedir. Şekil 4.4, Şekil 4.5, Şekil 4.6, Şekil 4.7, Şekil 4.8, Şekil 4.9, Şekil 4.10 ve Şekil 4.11 veri setinde eğitim ve test klasörleri içerisinde gauss, tuz biber, benek ve orijinal olarak ayrılmış görüntülerin birkaçına örnek teşkil etmektedir.

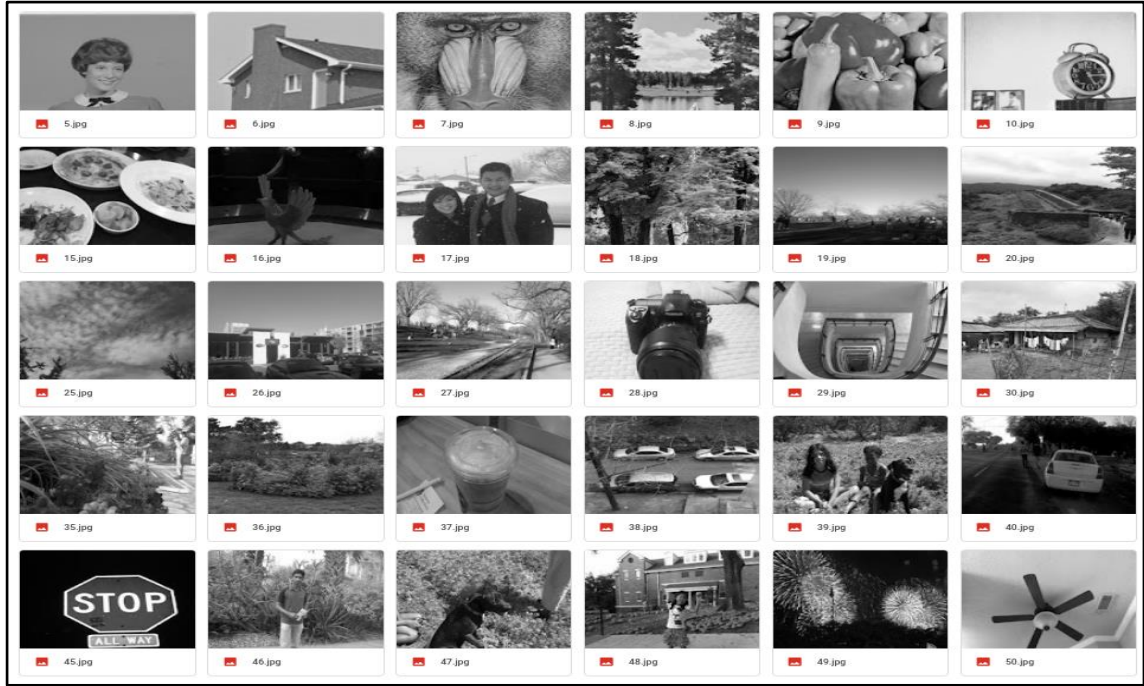
Doğrulama veri setinin, eğitim veri setinde yer alan görüntülerin %25'lik kısmını kapsayacak şekilde oluşturması sebebiyle örnek görüntülerinde doğrulama veri seti görüntülerine yer verilme ihtiyacı duyulmamıştır.



Şekil 4.3 Veri seti klasör yapısı



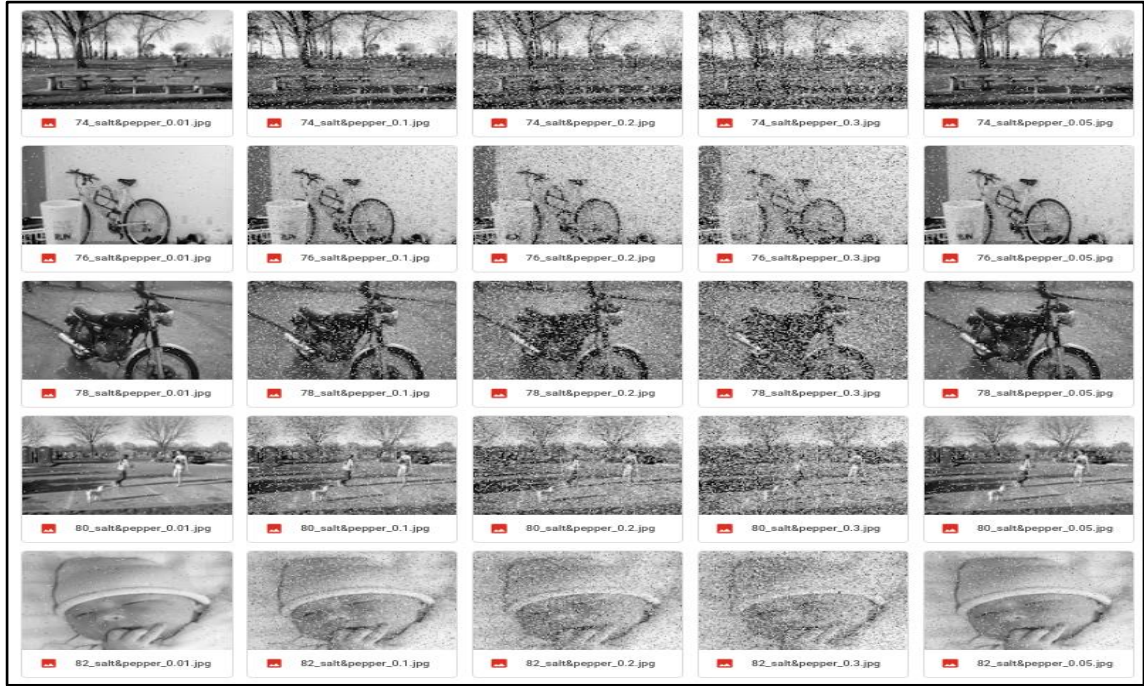
Şekil 4.4 Gauss gürültüsü eklenmiş eğitim veri seti örnekleri



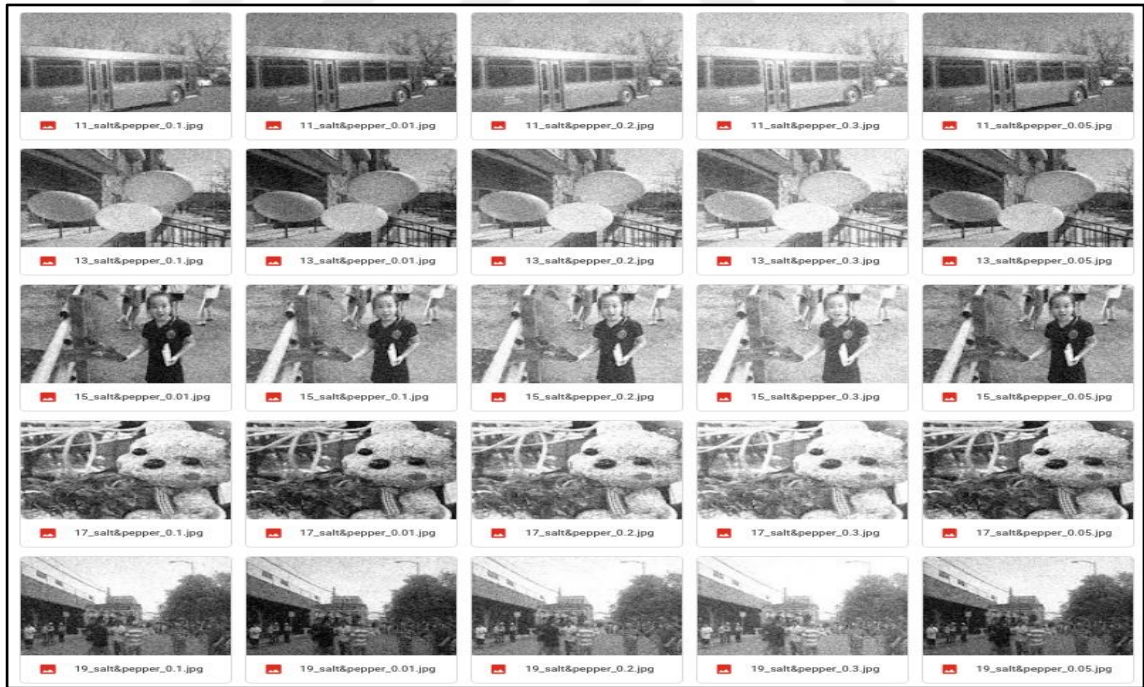
Şekil 4.5 Orjinal görüntüler eğitim veri seti örnekleri



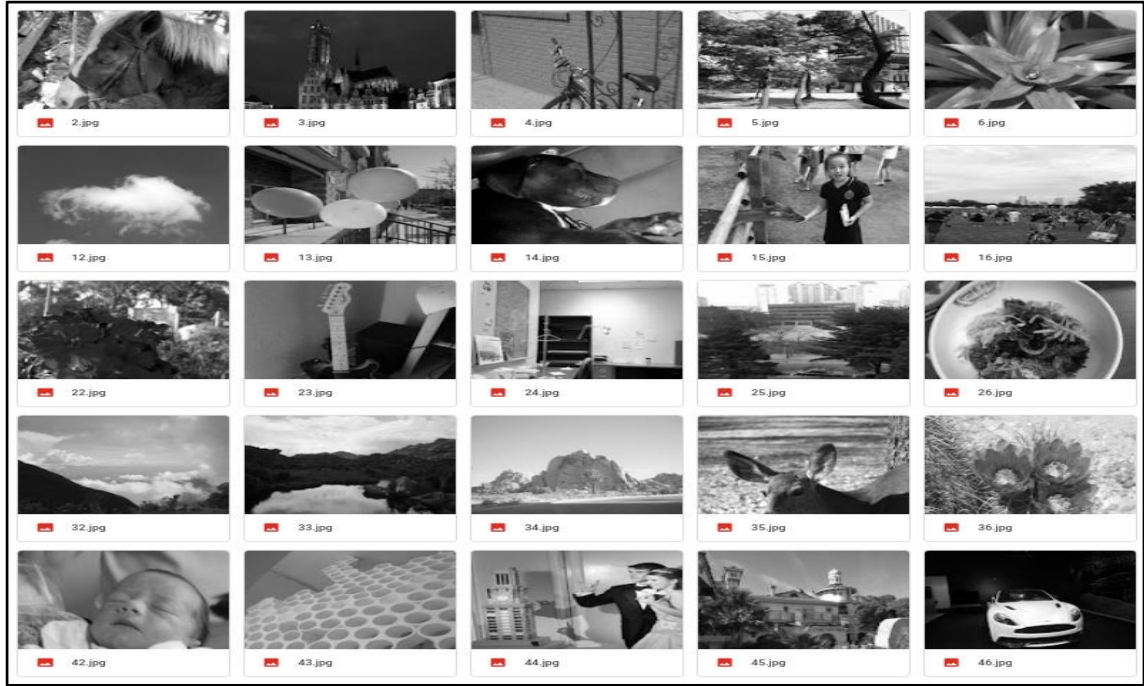
Şekil 4.6 Benek gürültüsü eklenmiş eğitim veri seti örnekleri



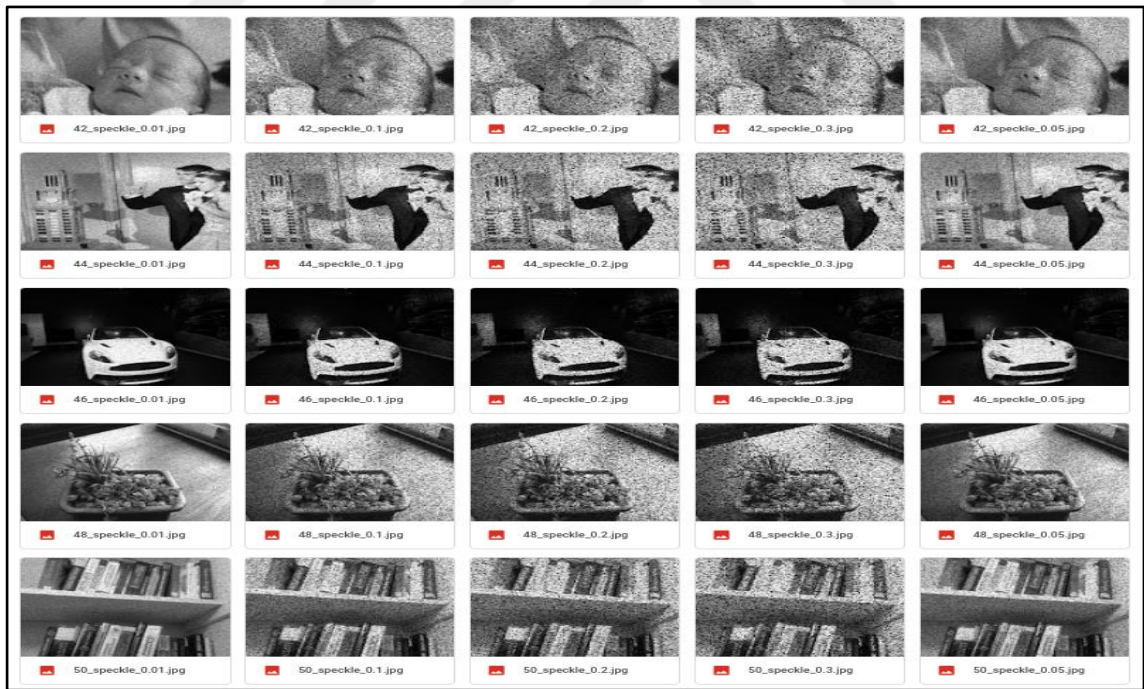
Şekil 4.7 Tuz-biber gürültüsü eklenmiş eğitim veri seti örnekleri



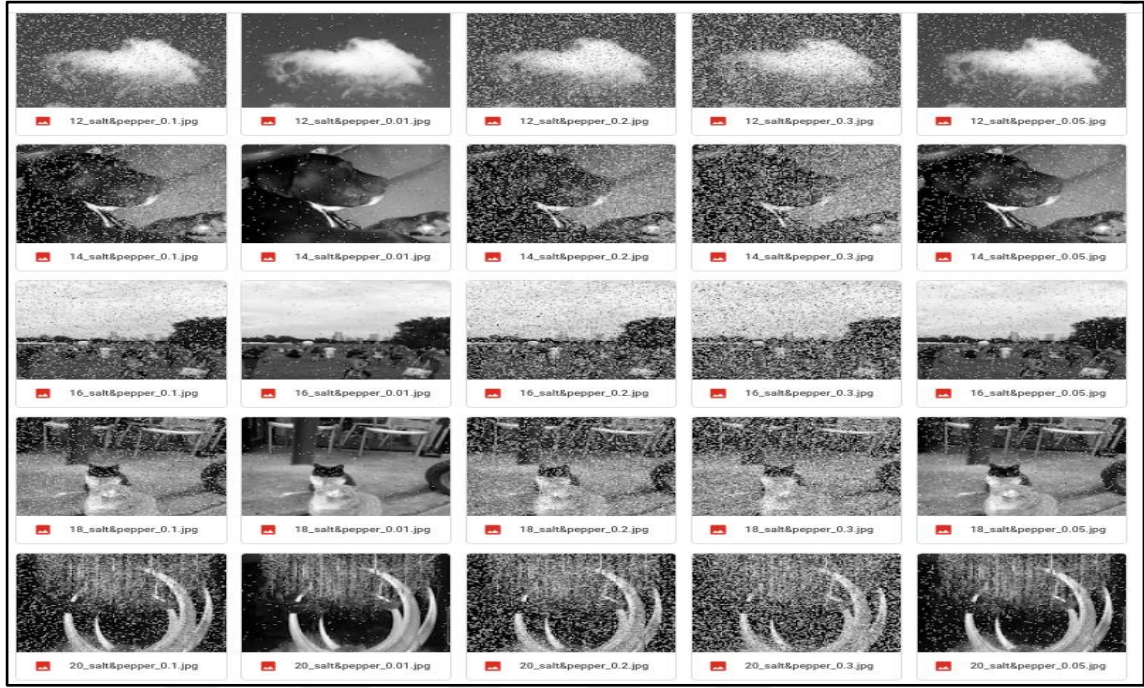
Şekil 4.8 Gauss gürültüsü eklenmiş test veri seti örnekleri



Şekil 4.9 Orijinal görüntü test veri seti örnekleri



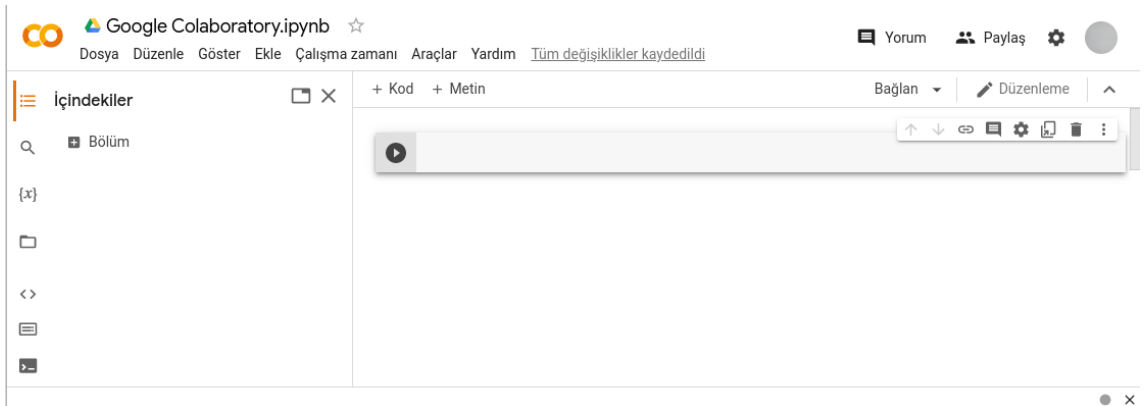
Şekil 4.10 Benek gürültüsü eklenmiş eğitim veri seti örnekleri



Şekil 4.11 Tuz-biber gürültüsü eklenmiş eğitim veri seti örnekleri

4.2 Derin Öğrenme Geliştirme Ortamı

Derin öğrenme alanında gerçekleştirilen çalışmalarda kullanılmakta olan birçok farklı geliştirme ortamı bulunmaktadır. Yapılan çalışma, Google Colaboratory platformunda gerçekleştirilmiştir. Python’da geliştirilmiş ve evrimsel sinir ağlarının bilgisayarlı görme uygulamalarında sıklıkla kullanımı tercih edilen kolay ve hızlı tasarlanabilen Keras kütüphanesi kullanılmıştır. Derin öğrenmede eğitimin gerçekleştirilmesi, geliştirme ortamlarında kendi yerel kaynaklarını kullanarak CPU desteği veya GPU desteği ile yapılabilmektedir. GPU desteği için ücretli ücretsiz birçok uygulama mevcuttur.



Şekil 4.12 Google Colaboratory platformu

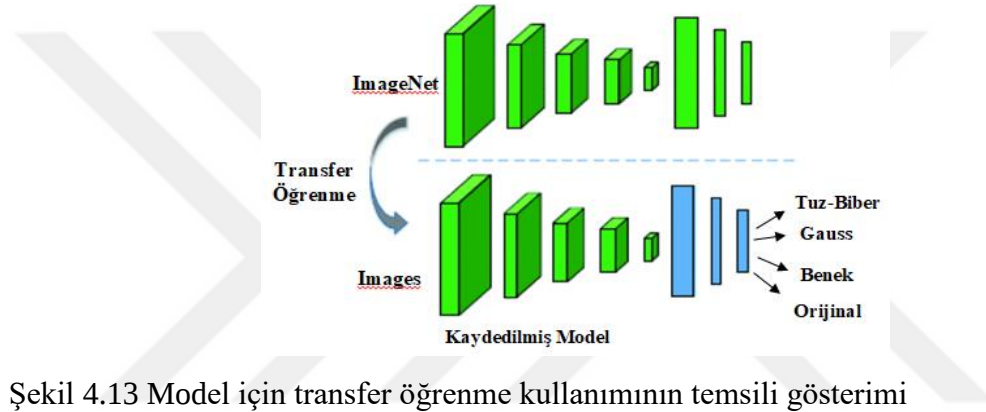
Ücretsiz GPU kullanım imkânı sunan kaynaklardan biri de Google Colaboratory platformudur (Ay Karakuş, 2018). Şekil 4.12’de Google Colaboratory platformu gösterilmektedir. Google Colaboratory platformu, derin öğrenme kütüphanelerinin hazır olarak sunulduğu ücretsiz bir ortamdır. Google Colaboratory herhangi bir yapılandırma gerektirmez. Yapılan uygulamalar ile ilgili açıklamalar yazmak istendiğinde metin alanı oluşturmaya izin verir. Uygulamaları başkaları ile kolaylıkla paylaşma imkânı sunar. Sağladığı GPU desteği ile de yüksek verim sağlayarak derin öğrenme çalışmalarına kolaylık sağlamaktadır (Carneiro ve ark., 2018).

4.3 Uygulama Ağının Oluşturulması

VGG16 literatürde en sık kullanılan algoritma olması sebebiyle oluşturulacak ESA mimarisinin tasarımında VGG16 kullanımı tercih edilmiş olup katman sayısında değişiklikler yapılarak uygulanmak istenen çalışma için en iyi sonucu vermesi hedeflenmiştir. Gündüz ve Cedimoğlu’nun Vikipedi görüntüleri ile yaptığı cinsiyet tahmini çalışmasında kendi yaptıkları ESA mimarisi ile VGG16 ve AlexNet mimarilerinin performansları kıyaslanmıştır. Kendi oluşturdukları ESA mimarisinin başarı oranı AlexNet’ten iyiyken, VGG16’dan kötü sonuç vermiştir (Gündüz & Cedimoğlu, 2019). VGG16 ‘ın bunun gibi çalışmalarda ve ImageNet yarışmasında elde ettiği başarılarından dolayı bu çalışmada ESA mimarisi tasarımında VGG16 kullanımı uygun görülmüştür. Kullanılan metot daha önce yapılmış birkaç çalışma incelenerek belirlenmiştir.

Görüntünün başlangıçtaki piksel sayısı modelin derinliğine doğrudan etki ettiği için çok düşük piksellerde görüntü kullanmak performans düşüklüğüne neden olmaktadır. VGG16’nın model olarak kullanıldığı birçok çalışmada 224x224 piksellik görüntüler ile çalışılmış olması nedeniyle ve performans düşüklüğü yaşanmaması için çalışmada kullanılacak görüntü boyutları 224x224 olarak belirlenmiştir. ESA mimarilerinden biri olan ve ImageNet yarışmasında yüksek oranda başarı gösteren VGG16 mimarisinin son 4 katmanı hariç diğer katmanları dondurulmuş ve üzerine yeni model eklenmiştir. Oluşturulan bu modelde yine VGG16’nın ImageNet yarışmasında elde edilmiş ağırlıkları kullanılarak transfer öğrenme gerçekleştirilmiştir. Transfer öğrenme bir

problemin çözümünde kullanılan sistemin eğitim aşamasında öğrenilen bilgisinin benzer problemlerin çözümünde de kullanılabilmesini ifade etmektedir. Bu çalışmada, transfer öğrenme yönteminin kullanılma amacı, ESA mimarisinin kısıtları arasında yer alan büyük eğitim veri seti gereksinimi ve yüksek hesaplama maliyeti gibi sınırlamaların üstesinden gelmektir. ImageNet yarışmasında elde edilen ağırlıklar kullanılarak transfer öğrenme ile yeni bir ESA mimarisi tasarlanmış ve bu mimarinin büyük eğitim veri setine ihtiyaç duymadan ve daha az hesaplama maliyeti ile daha hızlı, daha etkin sonuçlar elde etmek için kullanılması hedeflenmiştir. Şekil 4.13’de yeni oluşturulan modele ait transfer öğrenme kullanımının temsili gösterimi yer almaktadır.

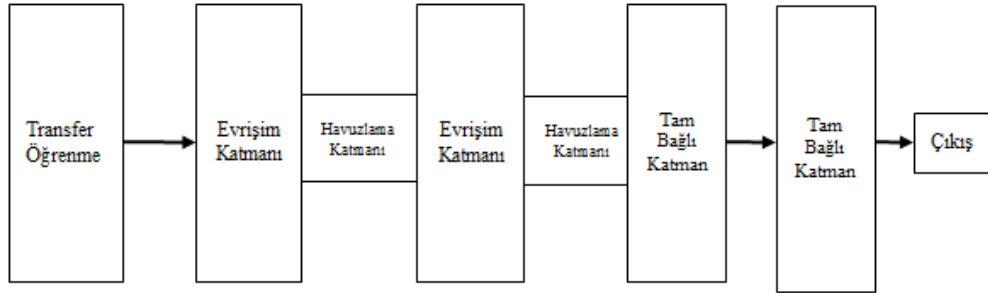


Şekil 4.13 Model için transfer öğrenme kullanımının temsili gösterimi

Dondurulan katmanlardan sonra iki blok eklenmiş, her blokta iki evrişim katmanı ve her evrişim işleminden sonra bir ortaklama katmanı eklenmiştir. Modelde kullanılan aktivasyon fonksiyonu Relu’dur. Aktivasyon fonksiyonları, evrişimin hızını kontrol etmek için kritik rol oynar. Relu aktivasyon fonksiyonu, hesaplamanın kolay olduğu ve hızlı yakınsama hızına ulaşan bir aktivasyon fonksiyonudur ve bu özelliği birçok çalışmada kanıtlanmıştır (Kirana ve ark., 2019). Daha önceki çalışmalarda yaygın olarak kullanılan Relu aktivasyon fonksiyonunun diğer aktivasyon fonksiyonlarına göre derin sinir ağlarının daha iyi eğitilmesini sağladığı ve bu nedenle derin sinir ağları için popüler bir aktivasyon fonksiyonu olduğu bilinmektedir (Shen & Yuan, 2019). Bu çalışmada da Relu aktivasyon fonksiyonunun kullanılması uygun bulunmuştur. Sonrasında iki adet tam bağlı katman eklenmiş ve tam bağlı katmanlarda belirli bir değerin altında kalan düğümlerde seyreltmeye gidilmesinin başarımı artırması sebebiyle ve aşırı öğrenmeyi önlemek için tam bağlı katmandan sonra 0.3 oranında dropout kullanılmıştır. Dropout, düğümlerin birbiri hakkında daha az bilgiye sahip olmasını ve bunun doğal sonucu olarak düğümlerin birbirlerinin ağırlık değişimlerinden

daha az etkilenmesini sağlayan genelde tam bağı katmanlardan sonra kullanılan bir tekniktir (Necmettin, 2021). Bunlara ilave çıkış katmanında eklenmesi ile yeni model kurulmuştur.

Derin öğrenme uygulamalarında hatayı minimize edebilmek için literatürde de sıkça rastlanan RMSProp, Adam, Adagrad, Adadelta ve SGD gibi optimizasyon algoritmaları kullanılmaktadır. Bu algoritmaların yardımı ile ağıın ürettiği çıkış değeri ve gerçek değer arasındaki fark küçük tutulmaya çalışılmaktadır. Çalışmada, RMSProp, Adam, Adagrad, Adadelta ve SGD optimizasyon algoritmalarına göre 70 turda eğitim yapılmış ve sonuçlar karşılaştırılmıştır. Şekil 4.14'de model mimarisine ait diyagram yer almaktadır.



Şekil 4.14 Model mimarisine ait diyagramı

Model: "sequential"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 7, 7, 512)	14714688
conv2d (Conv2D)	(None, 5, 5, 512)	2359808
zero_padding2d (ZeroPadding2D)	(None, 7, 7, 512)	0
conv2d_1 (Conv2D)	(None, 5, 5, 512)	2359808
max_pooling2d (MaxPooling2D)	(None, 2, 2, 512)	0
zero_padding2d_1 (ZeroPadding2D)	(None, 4, 4, 512)	0
conv2d_2 (Conv2D)	(None, 2, 2, 1024)	4719616
zero_padding2d_2 (ZeroPadding2D)	(None, 4, 4, 1024)	0
conv2d_3 (Conv2D)	(None, 2, 2, 1024)	9438208
max_pooling2d_1 (MaxPooling2D)	(None, 1, 1, 1024)	0
flatten (Flatten)	(None, 1024)	0
dense (Dense)	(None, 4096)	4198400
dense_1 (Dense)	(None, 4096)	16781312
dropout (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 4)	16388

Total params: 54,588,228
Trainable params: 46,952,964
Non-trainable params: 7,635,264

Şekil 4.15 Model mimarisine ait tablosu

Şekil 4.15’de yer alan model mimarisinde de görüldüğü üzere modelde toplam parametre sayısı yaklaşık 55M ‘dir. VGG16 mimarisinin dondurulan katmanları üzerine 2 bloktan oluşan yeni bir ekleme yapılarak model mimarisi tasarlanmıştır.

4.4 Değerlendirme

Matlab programı aracılığıyla oluşturulmuş olan veri seti için, derin öğrenme geliştirme ortamı Google Colaboratory’de, uygulama için tasarlanan ağ kullanılarak, Python programlama dili ile eğitim ve test işlemleri gerçekleştirmesinin ardından başarımlar ölçütleri yardımı ile optimizasyon algoritmaları değerlendirmiştir. Karışıklık matrisi, doğruluk, kesinlik, duyarlılık ve F1 skor değerleri ile optimizasyon algoritmalarının başarımlar düzeyleri karşılaştırılmıştır. Çizelge 4.2’de uygulamada kullanılan modele ilişkin karışıklık matrisinin nasıl görüneceğinin taslağı yer almaktadır.

Çizelge 4.2 Model karışıklık matrisi gösterimi

	TAHMİN ETİKET				
		Gauss	Orijinal	Benek	Tuz-biber
GERÇEK ETİKET	Gauss	g			
	Orijinal		o		
	Benek			b	
	Tuz-biber				t

5. BULGULAR ve TARTIŞMA

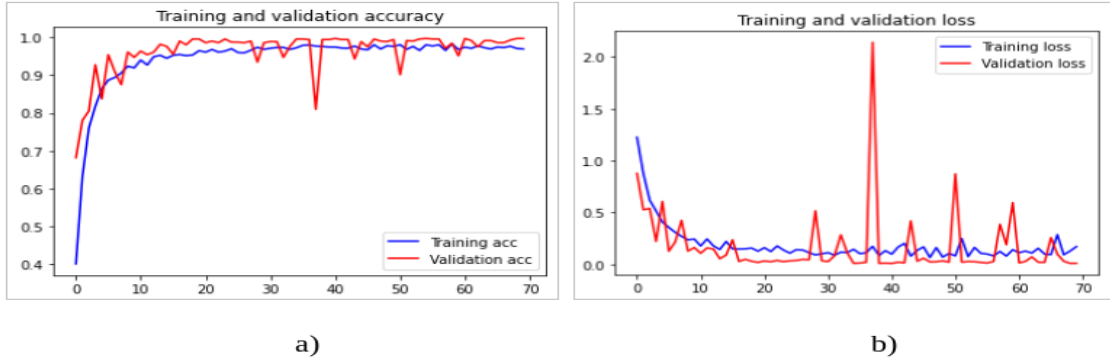
4 sınıflı, 2880 eğitim görüntüsü ve yine 4 sınıflı 720 doğrulama görüntüsünün, ImageNet yarışmasında kullanılan ağırlıklara sahip son 4 katmanı hariç diğerleri dondurularak tekrar yazılmış, literatürde sıklıkla kullanılan RMSProp, SGD, Adadelata, Adam ve Adagrad optimizasyon algoritmalarını içeren, yeni oluşturulan VGG16 mimarileri ile eğitilmesi sonucu açığa çıkan eğitim veri seti ve doğrulama veri seti doğruluk (accuracy) değerleri ve kayıp (loss) değerleri her optimizasyon algoritması için tespit edilerek test veri seti üzerindeki doğruluk, kesinlik, duyarlılık oranları ve F1 skorlarının performansları karşılaştırılmaktadır.

Doğrulama veri seti modelin performansını iyileştirmek için kullanılabilecek bir veri setidir. Bu veri seti yardımı ile modelin uygunluk değerlendirmesi gerçekleştirilebilmektedir. Her bir model için model eğitimi sonucu eğitim veri seti ve doğrulama veri seti için elde edilen doğruluk değerlerinin birbirlerine yakın değerler olması yeni oluşturulan VGG16 modellerinin oluşturulduğu halleriyle kullanılmasının uygun olduğu yönünde değerlendirilmesini sağlamaktadır.

5.1 Optimizasyon Algoritmalarının Karşılaştırması

Literatürde sıklıkla karşılaşılan optimizasyon algoritmalarından RMSProp, Adam, Adagrad, Adadelata ve SGD optimizasyon algoritması kullanılarak oluşturulan 5 farklı model aynı veri seti üzerinde aynı yöntemler ile eğitilerek test işlemleri gerçekleştirilmiş ve çalışma bitiminde çıkan sonuçlar değerlendirilmiştir.

5.1.1 RMSProp



Şekil 5.1 RMSProp optimizasyon algoritması için; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği

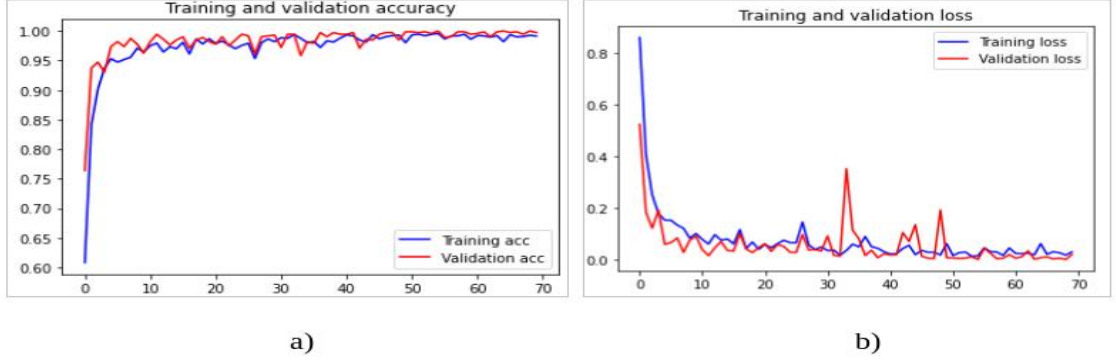
RMSProp optimizasyon algoritması kullanılarak oluşturulmuş mimari için eğitim veri seti ve doğrulama veri seti doğruluk değerlerine ve kayıp değerlerine Şekil 5.1’de yer verilmiştir. Şekil 5.1’de eğitim ve doğrulama görüntüleri için elde edilen doğruluk değerleri birbirine çok yakın değerlerde ve kayıp değerleri ise birkaç yerdeki farklılıklar dışında birbirine yakın düzeylerde seyretmektedir. Doğrulamadaki kayıp değerleri bazı turlarda artmış olmasına rağmen genel olarak incelendiğinde hem doğruluk hem de kayıp değerleri eğitim ve doğrulama veri seti için birbirine eşdeğer olarak hareket etmiştir. Tur sayısı 70 olarak belirlenen çalışmada eğitim veri seti için doğruluk değeri %97 iken doğrulama veri seti için doğrulama doğruluk değeri %100 olarak elde edilmiş; kayıp değerleri ise eğitim veri seti için 0.17 iken doğrulama veri seti için 0.01 olarak hesaplanmıştır. Eğitilen model ile 960 görüntüden oluşan test veri seti kullanılarak test yapılmıştır ve sonuçlar incelendiğinde 12 görüntünün hatalı bulunduğu, doğruluk değerinin ise %99 olarak hesaplandığı görülmüştür. Şekil 5.2’de test veri setinde yer alan ve yanlış etiketlenmiş görüntülerden birkaçı yer almaktadır.



Şekil 5.2 RMSProp optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntülere örnekler

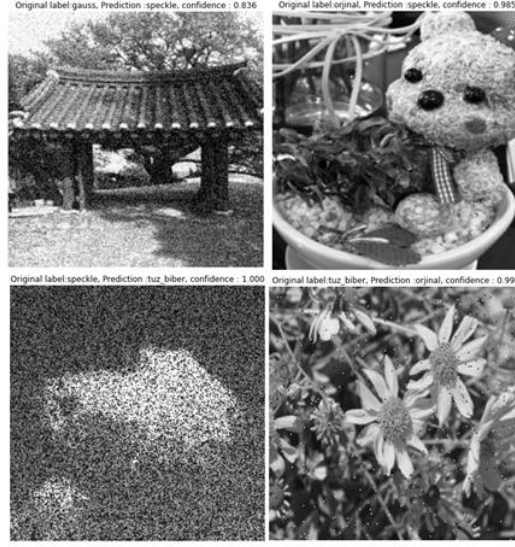
Test sonuçlarına göre, yanlış etiketlenen görüntülerin ağırlıklı olarak orijinal görüntüler olduğu ve bu görüntülerin benek gürültüsü eklenmiş görüntüler olarak tespit edildiği belirlenmiştir. Hatalı tespit edilen görüntülerin 7'sinin orijinal görüntüler olduğu ve bunlardan 6'sının benek gürültüsü eklenmiş görüntüler olarak, 1'inin ise gauss gürültüsü eklenmiş görüntü olarak etiketlendiği görülmüştür. Bunun yanında benek görüntülü görüntülerden 3'ü gauss gürültülü görüntüler olarak, tuz-biber gürültülü görüntülerden 1'i orijinal görüntü olarak, 1'i ise benek gürültülü görüntü olarak etiketlenmiştir. Gauss gürültüsü eklenmiş görüntülerinin tamamı gauss gürültüsü eklenmiş görüntüler olarak doğru tahmin edilmiştir. Ancak elde edilen sonuç gauss gürültüsünü tahmin etmede %100 başarı sağlandığı anlamına gelmemektedir. Bunun sebebi diğer türlerde yapılan yanlış tahmin hatalarından bazılarının gauss gürültülü görüntüler olarak etiketlenmiş olmasıdır.

5.1.2 Adam



Şekil 5.3 Adam optimizasyon algoritması için; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği

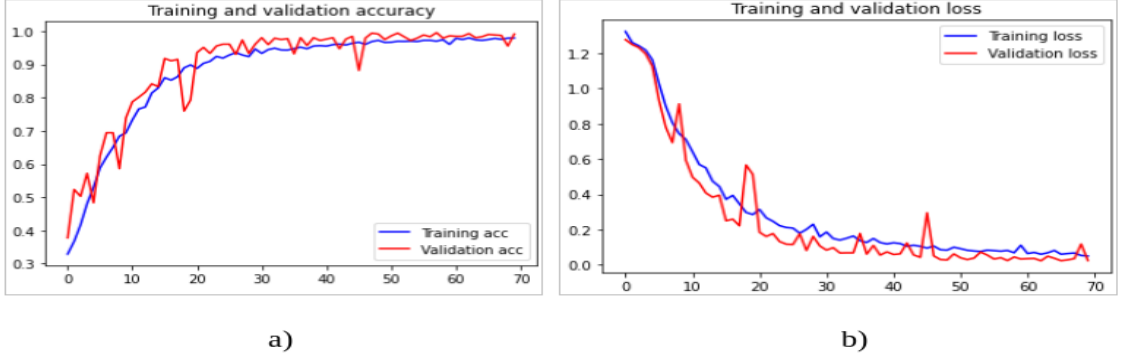
Adam optimizasyon algoritması kullanılarak oluşturulmuş mimari için eğitim veri seti ve doğrulama veri seti doğruluk değerlerine ve kayıp değerlerine, Şekil 5.3’de yer verilmiştir. Şekil 5.3’de eğitim ve doğrulama görüntüleri için elde edilen doğruluk ve kayıp değerlerinin genellikle birbirlerine yakın seyrettiği görülmektedir. Tur sayısı 70 olarak belirlenen çalışmada eğitim veri seti için doğruluk değeri ile doğrulama veri seti için doğrulama doğruluk değeri %99 olarak elde edilmiş; kayıp değerleri ise eğitim veri seti için 0.03 iken doğrulama veri seti için 0.02 olarak hesaplanmıştır. Eğitilen model ile 960 görüntüden oluşan test veri seti kullanılarak test yapılmıştır ve sonuçlar incelendiğinde 15 görüntünün hatalı bulunduğu, doğruluk değerinin ise %98 olarak hesaplandığı görülmüştür. Şekil 5.4’te test veri setinde yer alan ve yanlış tespit edilmiş görüntülerden birkaçına yer verilmiştir.



Şekil 5.4 Adam optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntülere örnekler

Test sonuçlarına göre, yanlış etiketlenen görüntülerin, ağırlıklı olarak gauss gürültüsü eklenmiş görüntüler ve orijinal görüntüler olduğu ve her iki türünde benek gürültüsü eklenmiş görüntüler olarak tespit edildiği belirlenmiştir. Hatalı görüntülerin 5'inin gauss gürültüsü eklenmiş görüntüler, 4'ünün orijinal görüntüler olduğu ve her iki tür içinde tahmin edilen sınıfın benek gürültülü görüntüler olduğu tespit edilmiştir. Bununla birlikte benek gürültüsü eklenmiş görüntülerden 3'ü tuz-biber gürültüsü eklenmiş görüntü olarak, 1'i ise gauss gürültüsü eklenmiş görüntü olarak etiketlenmiştir. Tuz-biber gürültüsü eklenmiş görüntülerden 1'i orijinal görüntü olarak, 1'i ise benek gürültüsü eklenmiş görüntü olarak belirlenmiştir. İncelemeler sonucunda Adam optimizasyon algoritması kullanılarak oluşturulan modelde yüksek değerde benek gürültüsüne sahip olan görüntülerin tuz-biber gürültüsü ile karıştırıldığı, belirli bir değerde olmasa da yoğun gauss gürültüsü içeren görüntülerin benek gürültüsü olarak etiketlendiği ve orijinal görüntülerden bazılarının yine benek gürültüsü olarak algılandığı gözlemlenmektedir.

5.1.3 Adagrad



Şekil 5.5 Adagrad optimizasyon algoritması için; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği

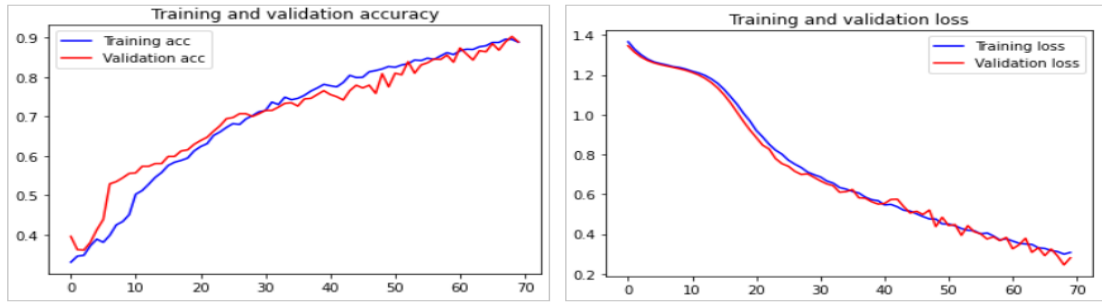
Adagrad optimizasyon algoritması kullanılarak oluşturulmuş mimari için eğitim veri seti ve doğrulama veri seti doğruluk değerlerine ve kayıp değerlerine Şekil 5.5’ de yer verilmiştir. Şekil 5.5’ de eğitim ve doğrulama görüntüleri için elde edilen doğruluk ve kayıp değerlerinin genellikle birbirlerine yakın seyrettiği görülmektedir. Tur sayısı 70 olarak belirlenen çalışmada eğitim veri seti için doğruluk değeri %98, doğrulama veri seti için doğrulama doğruluk değeri %99 olarak elde edilmiş; kayıp değerleri ise eğitim veri seti için 0.05 iken doğrulama veri seti için 0.03 olarak hesaplanmıştır. Eğitilen model ile 960 görüntüden oluşan test veri seti kullanılarak test yapıp sonuçlar incelendiğinde 26 görüntünün hatalı bulunduğu, doğruluk değerinin ise %97 olarak hesaplandığı görülmüştür. Şekil 5.6’ da test veri setinde yer alan ve yanlış tespit edilmiş görüntüler için bazı örnekler verilmiştir.



Şekil 5.6 Adagrad optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntülere örnekler

Test sonuçlarına göre, yanlış etiketlenen görüntülerin, ağırlıklı olarak orijinal görüntüler ve benek gürültüsü eklenmiş görüntüler olduğu tespit edilmiştir. Ağırlıklı olarak orijinal görüntülerin benek gürültüsü eklenmiş görüntüler; benek gürültüsü eklenmiş görüntülerin ise gauss gürültüsü eklenmiş görüntüler şeklinde etiketlendiği görülmüştür. Hatalı görüntülerin 13'ünün orijinal görüntüler olduğu ve bunlardan 9'unun benek gürültüsü eklenmiş görüntüler olarak tespit edildiği, 1'inin gauss gürültülü görüntü, 1'inin ise tuz-biber gürültülü görüntü olarak etiketlendiği görülmüştür. Bununla birlikte benek gürültüsü eklenmiş görüntülerden 11'inin hatalı etiketlendiği, bu görüntülerden 9'unun gauss gürültüsü eklenmiş görüntü, 2'sinin ise tuz-biber gürültüsü eklenmiş görüntü olarak etiketlendiği tespit edilmiştir. Tuz-biber gürültüsü eklenmiş görüntülerden ise yalnızca 2'si benek gürültüsü eklenmiş görüntü olarak etiketlenmiştir. Sonuç olarak, Adagrad optimizasyon algoritması kullanılarak oluşturulan modelde orijinal görüntülerin benek gürültüsü ile karıştırılmaya yatkın olduğu kanısına varılmış olup yüksek oranlarda benek gürültüsüne sahip görüntülerin ise gauss gürültüsü ile karıştırılabildiği gözlemlenmiştir.

5.1.4 Adadelata



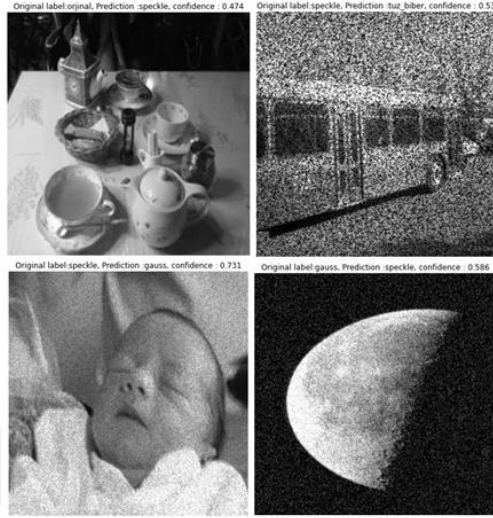
a)

b)

Şekil 5.7 Adadelata optimizasyon algoritması için; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği

Adadelata optimizasyon algoritması kullanılarak oluşturulmuş mimari için eğitim veri seti ve doğrulama veri seti doğruluk değerlerine ve kayıp değerlerine Şekil 5.7' de yer verilmiştir. Şekil 5.7' de eğitim ve doğrulama görüntüleri için elde edilen doğruluk değerlerinin ve kayıp değerlerinin birbirleri ile neredeyse aynı değerlerde seyrettiği görülmektedir. Tur sayısı 70 olarak belirlenen çalışmada eğitim veri seti için doğruluk değeri ile doğrulama veri seti için doğrulama doğruluk değeri %89 olarak elde edilmiş; kayıp değerleri ise eğitim veri seti için 0.31 ve doğrulama veri seti için 0.28 olarak

hesaplanmıştır. Eğitilen model ile 960 görüntüden oluşan test veri seti kullanılarak test yapılmıştır ve sonuçlar incelendiğinde 102 görüntünün hatalı bulunduğu, doğruluk değerinin ise %89 olarak hesaplandığı görülmüştür. Şekil 5.8’de test veri setinde yer alan ve yanlış tespit edilmiş görüntülere örnek verilmiştir.



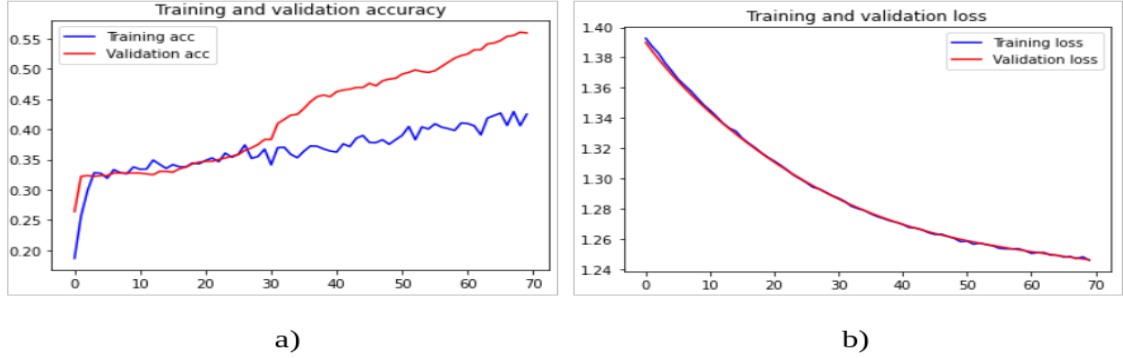
Şekil 5.8 Adadelata optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntülere örnekler

Test sonuçlarına göre, yanlış etiketlenen görüntülerin ağırlıklı olarak benek gürültüsü içeren görüntüler olduğu ve bunların gauss gürültüsü eklenmiş görüntüler ve tuz-biber gürültüsü eklenmiş görüntüler olarak etiketlendiği belirlenmiştir. Bununla birlikte gauss gürültüsü eklenmiş görüntülerin benek gürültüsü olarak etiketlendiği, orijinal görüntülerin benek gürültüsü ve az sayıda olmak üzere gauss gürültüsü ve tuz-biber gürültüsü olarak eklenmiş görüntüler olarak etiketlendiği gözlemlenmiştir. Hatalı tespit edilen görüntülerden orijinal görüntülerin 15’i benek gürültülü görüntüler, 6’sı gauss gürültülü görüntüler ve 4’ü tuz-biber gürültülü görüntüler olarak etiketlenmiştir. Benek gürültüsü eklenmiş görüntülerden 39’u tuz-biber gürültüsü eklenmiş görüntüler, 15’i gauss gürültüsü eklenmiş görüntüler, 1’i ise orijinal görüntü olarak tespit edilmiştir. Gauss gürültüsü eklenmiş görüntülerden ise 18’i benek gürültüsü eklenmiş görüntüler ve 1’i tuz-biber gürültüsü eklenmiş görüntü şeklinde etiketlenmiş olarak belirlenmiştir. Tuz-biber gürültülü görüntülerde ise yalnızca 1’er tane hatalı görüntü yer almaktadır.

Sonuçlara göre, gürültü oranı düşük olan benek gürültüsü eklenmiş görüntülerin gauss gürültüsü eklenmiş olan görüntüler ile karıştırıldığı, yüksek oranda benek gürültüsüne sahip görüntülerin ise tuz-biber gürültüsü eklenmiş görüntüler ile karıştırıldığı

gözlemlenmiştir. Orijinal görüntülerin, çoğunluğu benek gürültüsü olmak üzere gauss ve tuz-biber olarak etiketlendiği ve gauss gürültülü görüntülerin ise benek gürültüsü eklenmiş görüntüler ile karıştırıldığı gözlemlenmiştir.

5.1.5 SGD



Şekil 5.9 SGD optimizasyon algoritması için; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği

SGD optimizasyon algoritması kullanılarak oluşturulmuş mimari için eğitim veri seti ve doğrulama veri seti doğruluk değerlerine ve kayıp değerlerine Şekil 5.9'da yer verilmiştir. Şekil 5.9' da eğitim ve doğrulama görüntüleri için elde edilen doğruluk değerlerinin birbirinden çok farklı ilerlediği, doğrulama veri seti için doğruluk değerinin daha yüksek olduğu görülmektedir. Ancak kayıp değerlerinin birbirleri ile neredeyse aynı değerlerde seyrettiği görülmektedir. Tur sayısı 70 olarak belirlenen çalışmada eğitim veri seti için doğruluk değeri %42 iken, doğrulama veri seti için doğrulama doğruluk değeri %55 olarak elde edilmiş; kayıp değerleri ise eğitim veri seti ve doğrulama veri seti için 1.25 olarak hesaplanmıştır. Eğitilen model ile 960 görüntüden oluşan test veri seti kullanılarak test yapılmıştır ve sonuçlar incelendiğinde 406 görüntünün hatalı bulunduğu, doğruluk değerinin ise %58 olarak hesaplandığı görülmüştür. Şekil 5.10'da test veri setinde yer alan ve yanlış tespit edilmiş görüntülere örnek verilmiştir.



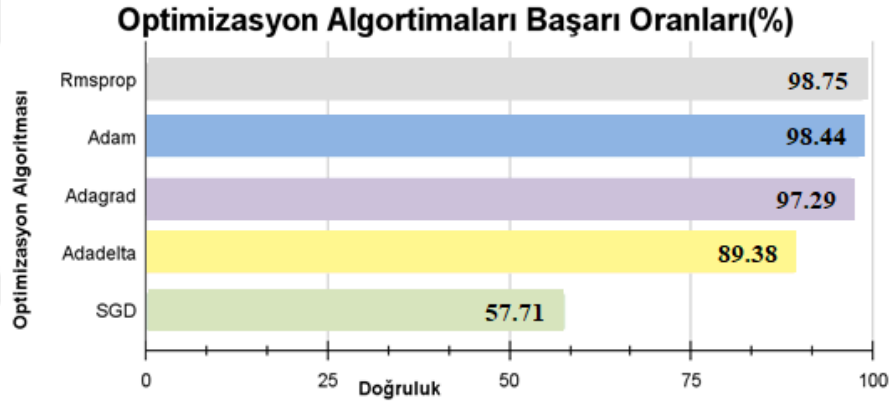
Şekil 5.10 SGD optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntülere örnekler

Test sonuçlarına göre, orijinal görüntülerin hiçbirinin doğru bulunmadığı, tüm orijinal görüntülerin farklı gürültü türlerindeki görüntü sınıfları ile etiketlendiği gözlemlenmiştir. SGD optimizasyon algoritmasının genellikle resim tanıma gibi uygulamalarda kullanılması tavsiye edilmemektedir. Sonuçlar göz önünde bulundurulduğunda tüm başarımların düşük olduğu, her bir sınıfın farklı diğer sınıflar ile orantısız olarak karıştırıldığı görülmüştür. Hatalı tespit edilen görüntülere göre, neredeyse tüm görüntü kombinasyonlarının birbirleri ile karıştırıldığı ve hiçbir orijinal görüntünün tespit edilemediği, SGD optimizasyon algoritması kullanılarak oluşturulan modelde, gauss gürültülü görüntülerin çoğunluğunun benek gürültülü görüntüler olmak üzere benek ve tuz-biber gürültülü görüntülerle karıştırdığı; benek gürültülü görüntülerin çoğunluğunun tuz-biber gürültülü görüntüler olmak üzere gauss ve tuz-biber gürültülü olarak etiketlediği; tuz- biber gürültülü görüntülerin ise neredeyse aynı oranda benek ve gauss gürültülü görüntüler olarak etiketlendiği gibi bir çok farklı kombinasyonlu hatalı tespit edilen görüntüler bulunduğu gözlemlenmiştir. Hatalı tespit edilen görüntülerden belirgin bir durum çıkarmak mümkün görünmemektedir. Kısaca, SGD optimizasyon algoritmanın tahmin edildiği gibi görüntü üzerinde iyi sonuçlar vermediği yapılan çalışma sonucunda da görülmüştür. Ek 1'de her bir optimizasyon algoritması için hatalı görüntülere yer verilmiştir.

Genel Değerlendirme

Çizelge 5.1 Çalışmada kullanılan optimizasyon algoritmalarının doğrulukları

Sıra Numarası	Optimizasyon Algoritması	Doğruluk
1	RMSProp	%98.75
2	Adam	%98.44
3	Adagrad	%97.29
4	Adadelata	%89.38
5	SGD	%57.71



Şekil 5.11 Çizelge 5.1'in grafik gösterimi

Çizelge 5.1 ve Şekil 5.11'de RMSProp, Adam, Adagrad, Adadelata ve SGD optimizasyon algoritmaları kullanılarak oluşturulan 5 farklı VGG16 modeline ait doğruluk değerlerine yer verilmiştir. Şekil 5.11 incelendiğinde RMSProp optimizasyon algoritması kullanılarak oluşturulan modelin %98.75 doğruluk oranı ile en iyi doğruluk oranına sahip olduğu görülmektedir. Diğer optimizasyon algoritmalarının ise doğruluk oranları sırasıyla Adam optimizasyon algoritması için %98.44, Adagrad optimizasyon algoritması için %97.29, Adadelata optimizasyon algoritması için %89.38 ve SGD optimizasyon algoritması için %57.71 şeklinde olduğu gözlemlenmiştir.

Çizelge 5.2 Optimizasyon algoritmalarının karışıklık matrisleri

			Tahmin Etiket			
			Gauss	Orijinal	Benek	Tuz-biber
Gerçek Etiket	RMSProp	Gauss	300	0	0	0
		Orijinal	1	53	6	0
		Benek	3	0	297	0
		Tuz-biber	0	1	1	298
		Toplam Hatalı Bulunan Sayısı: 12				
			Tahmin Etiket			
			Gauss	Orijinal	Benek	Tuz-biber
Gerçek Etiket	Adam	Gauss	295	0	5	0
		Orijinal	0	56	4	0
		Benek	1	0	296	3
		Tuz-biber	0	1	1	298
		Toplam Hatalı Bulunan Sayısı: 15				
			Tahmin Etiket			
			Gauss	Orijinal	Benek	Tuz-biber
Gerçek Etiket	Adagrad	Gauss	300	0	0	0
		Orijinal	1	47	11	1
		Benek	9	0	289	2
		Tuz-biber	0	0	2	298
		Toplam Hatalı Bulunan Sayısı: 26				

Çizelge 5.2 (Devam) Optimizasyon algoritmalarının karışıklık matrisleri

			Tahmin Etiket			
			Gauss	Orijinal	Benek	Tuz-biber
Gerçek Etiket	Adadelta	Gauss	281	0	18	1
		Orijinal	6	35	15	4
		Benek	15	1	245	39
		Tuz-biber	1	1	1	297
		Toplam Hatalı Bulunan Sayısı: 102				
Gerçek Etiket	SGD		Tahmin Etiket			
			Gauss	Orijinal	Benek	Tuz-biber
		Gauss	171	0	99	30
		Orijinal	38	0	19	3
		Benek	44	0	148	108
		Tuz-biber	33	0	32	235
		Toplam Hatalı Bulunan Sayısı: 406				

Çizelge 5.2’de optimizasyon algoritmalarının her biri için karışıklık matrisleri verilmiştir. Karışıklık matrisi, gerçek etiket değerleri ile tahmin edilen etiketlerin dağılımlarını daha iyi analiz etmemizi sağlamaktadır. Çizelge 5.2 incelendiğinde RMSProp optimizasyon algoritması ile Adam optimizasyon algoritmalarının toplam hata sayılarının birbirlerine yakın olduğu, Adagrad optimizasyon algoritmasının hata sayısının ise RMSProp ve Adam optimizasyon algoritmalarına göre daha fazla, Adadelta ve SGD optimizasyon algoritmalarının hata sayısından çok daha iyi olduğu görülmektedir. Ancak Adadelta ve SGD optimizasyon algoritmalarının hata sayıları fazla olup görüntüler üzerindeki hata oranı dağılımları da farklılık göstermektedir.

Her bir optimizasyon algoritmasında orijinal görüntülerin benek gürültüsü olan görüntüler ile karıştırıldığı örneklere rastlanmaktadır. RMSProp, Adam ve Adagrad optimizasyon algoritmaları için hatalı olarak etiketlenen bu orijinal görüntüler incelendiğinde, hatanın çoğunlukla aynı görseller üzerinde yapıldığı görülmektedir. Bu

durumun sebebinin veri setinde yer alan orijinal görüntülerin içerdikleri parlak ve koyu noktalar sebebiyle benek gürültüsü olarak algılanması olduğu düşünülmektedir. En kötü sonuçlara sahip olan Adadelta ve SGD optimizasyon algoritmaları için hata dağılımları daha farklıdır. İki optimizasyon algoritması da diğerlerine göre daha çok hata sayısına sahip olup SGD optimizasyon algoritmasının orijinal görüntülerin hiçbirini doğru tespit edemediği görülmektedir.

Yine RMSProp, Adam ve Adagrad optimizasyon algoritmaları için hata oranı dağılımları incelendiğinde gauss gürültüsü eklenmiş görüntüler ve tuz-biber gürültüsü eklenmiş görüntülerin çoğunlukla doğru tespit edildiği görülmektedir. Benek gürültüsü eklenmiş görüntüler ve orijinal görüntüler, gauss ve tuz-biber gürültüsü eklenmiş görüntülere göre daha çok hatalı tespit edilmiştir. Hatalı tahmin edilen benek gürültüsü eklenmiş görüntülerin çoğunun gauss gürültüsü eklenmiş görüntüler ile karıştırıldığı görülmüştür. Gauss gürültüsü, görüntü elde etme sırasında yüksek sıcaklık, aydınlatmanın az olması veya transferlerden kaynaklı olarak görüntüde oluşan bozulmalardır, benek gürültüsü ise görüntüde parlak ve koyu noktalar olarak görülen yapıcı ve yıkıcı girişim modellerinden kaynaklanan bozulmalardır. Benek gürültüsü eklenmiş görüntülerin gauss gürültüsü olarak belirlenmiş olmasının nedeni belirli oranlarda (0.05, 0.01, 0.1, 0.2, 0.3, 0.4) gürültü eklenerek oluşturulan gürültülü görüntülerde bazı oranların benek gürültülü görüntüleri, gauss gürültüsündeki gibi algılanmasına sebep olmaktan kaynaklandığı düşünülmektedir. Toplam hata sayıları incelendiğinde en iyi sonucun RMSProp optimizasyon algoritması kullanılarak oluşturulan mimaride olduğu gözlemlenmiş olup sırasıyla Adam, Adagrad, Adadelta ve SGD optimizasyon algoritmaları olarak hata sayıları artmaktadır.

Çizelge 5.3 Modelin performans ölçümleri

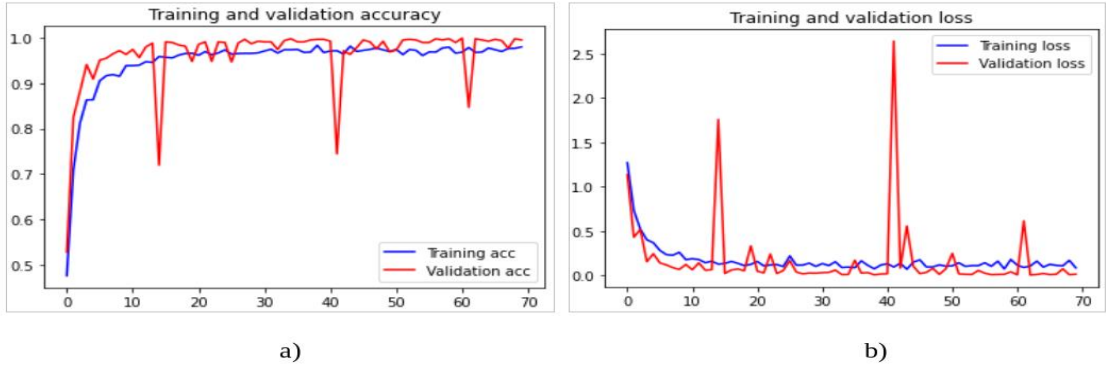
Optimizasyon Algoritmaları	Performans Ölçümleri	Kesinlik	Duyarlılık	F1 Skor
RMSProp	Makro Ort.	0.99	0.97	0.98
	Ağırlıklı Ort.	0.99	0.99	0.99
Adam	Makro Ort.	0.98	0.97	0.98
	Ağırlıklı Ort.	0.98	0.98	0.98
Adagrad	Makro Ort.	0.98	0.94	0.95
	Ağırlıklı Ort.	0.97	0.97	0.97
Adadelata	Makro Ort.	0.91	0.83	0.86
	Ağırlıklı Ort.	0.90	0.90	0.89
SGD	Makro Ort.	0.43	0.46	0.44
	Ağırlıklı Ort.	0.54	0.58	0.55

Çizelge 5.3’de modele ait performans ölçümleri görülmektedir. Optimizasyon algoritmalarına ait kesinlik, duyarlılık ve F1 skorun yer aldığı tablo incelendiğinde RMSProp, Adam ve Adagrad optimizasyon algoritmaları için makro ortalama ve ağırlıklı ortalama değerlerinin birbirlerine çok yakın olduğu, Adadelata ve SGD optimizasyon algoritmaları için ise daha düşük olduğu görülmektedir.

5.2 Katman Karşılaştırması

Bu bölümde, bir önceki bölümde farklı optimizasyon algoritmaları (RMSProp, Adam, Adagrad, Adadelata ve SGD) kullanılarak oluşturulan VGG16 mimarilerinden, en iyi performansı sağlamış olan RMSProp optimizasyon algoritması için VGG16 mimarisinin dondurulan katmanları yerine 1,2,3 ve 5 katman eklenerek oluşturulan yeni mimarilerin performansları karşılaştırılmaktadır.

5.2.1 Bir (1) katman eklenerek oluşturulan VGG16 mimarisi



Şekil 5.12 RMSProp optimizasyon algoritması için bir katmanlı mimari; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği

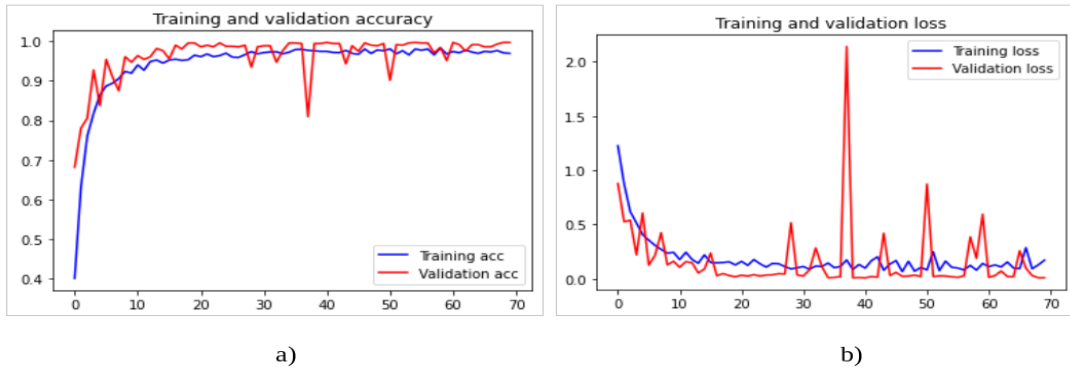
RMSProp optimizasyon algoritması ile bir (1) katman kullanılarak oluşturulmuş mimari için eğitim veri seti ve doğrulama veri seti doğruluk değerlerine ve kayıp değerlerine Şekil 5.12’de yer verilmiştir. Şekil 5.12 incelendiğinde 70 turda eğitim veri seti için doğruluk değeri %98 iken doğrulama veri seti için doğrulama doğruluk değeri %100 olarak elde edilmiş; kayıp değerleri ise eğitim veri seti için 0.09 iken doğrulama veri seti için 0.02 olarak hesaplanmıştır. Eğitilen model ile 960 görüntüden oluşan test veri seti kullanılarak test yapılmıştır ve sonuçlar incelendiğinde 19 görüntünün hatalı bulunduğu, doğruluk değerinin ise %98 olarak hesaplandığı görülmüştür. Şekil 5.13’te test veri setinde yer alan ve yanlış tespit edilmiş görüntülere örnek verilmiştir.



Şekil 5.13 RMSProp optimizasyon algoritması ile oluşturulan bir katmanlı modelin test sonucunda hatalı bulunan görüntülere örnekler

Test sonuçlarına göre, hatalı görüntülerin 8'inin benek gürültüsü eklenmiş görüntüler olduğu ve bunlardan 7'sinin gauss gürültüsü eklenmiş görüntüler olarak, 1'inin ise tuz-biber gürültüsü eklenmiş görüntü olarak etiketlendiği gözlemlenmiştir. Bunun yanında orijinal görüntülerin 5'i benek gürültülü görüntüler, 1'i ise gauss gürültüsü eklenmiş görüntü olarak, gauss gürültüsü eklenmiş görüntülerden 2'si benek gürültüsü eklenmiş görüntüler olarak, tuz-biber gürültülü görüntülerden 1'i orijinal, 2'si ise benek gürültülü olarak etiketlenmiştir. Yanlış etiketlenen görüntülerin ağırlıklı olarak benek gürültüsü eklenmiş görüntüler ve orijinal görüntüler olduğu görülmüştür. Benek gürültüsü eklenmiş görüntülerin gauss gürültüsü eklenmiş görüntüler ile orijinal görüntülerin ise benek gürültüsü eklenmiş görüntüler ile karıştırıldığı belirlenmiştir. Benek gürültülü görüntülerde gürültü oranı fazla olan görüntülerin genellikle gauss gürültülü görüntüler ile karıştırıldığı gözlemlenmektedir.

5.2.2 İki (2) katman eklenerek oluşturulan VGG16 mimarisi



Şekil 5.14 RMSProp optimizasyon algoritması için iki katmanlı mimari; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği

RMSProp optimizasyon algoritması kullanılarak oluşturulmuş mimari için eğitim veri seti ve doğrulama veri seti doğruluk değerlerine ve kayıp değerlerine Şekil 5.14'de yer verilmiştir. Şekil 5.14 incelendiğinde eğitim ve doğrulama görüntüleri için elde edilen doğruluk değerleri birbirine çok yakın değerlerde ve kayıp değerleri ise birkaç yerdeki farklılıklar dışında birbirine yakın düzeylerde seyretmektedir. Doğrulamadaki kayıp değerleri bazı turlarda artmış olmasına rağmen genel olarak incelendiğinde hem doğruluk hem de kayıp değerleri eğitim ve doğrulama veri seti için birbirine eşdeğer olarak hareket etmiştir. Tur sayısı 70 olarak belirlenen çalışmada eğitim veri seti için doğruluk değeri %97 iken doğrulama veri seti için doğrulama doğruluk değeri %100

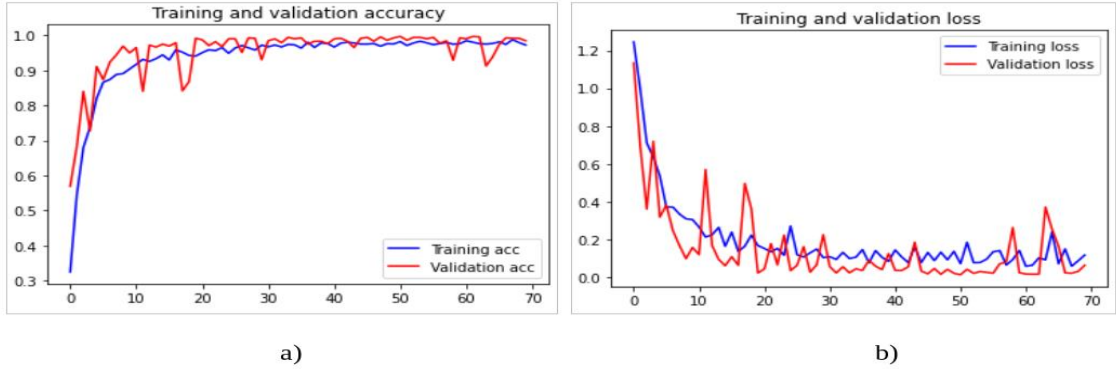
olarak elde edilmiş; kayıp değerleri ise eğitim veri seti için 0.17 iken doğrulama veri seti için 0.01 olarak hesaplanmıştır. Eğitilen model ile 960 görüntüden oluşan test veri seti kullanılarak test yapılmış ve sonuçlar incelendiğinde 12 görüntünün hatalı bulunduğu, doğruluk değerinin ise %99 olarak hesaplandığı görülmüştür. Şekil 5.15'te test veri setinde yer alan ve yanlış etiketlenmiş görüntüler yer almaktadır.



Şekil 5.15 RMSProp optimizasyon algoritması ile oluşturulan iki (2) katmanlı modelin test sonucunda hatalı bulunan görüntülere örnekler

Test sonuçlarına göre, yanlış etiketlenen görüntülerin ağırlıklı olarak orijinal görüntüler olduğu ve bunların benek gürültüsü eklenmiş görüntüler olarak tespit edildiği belirlenmiştir. Hatalı görüntülerin 7'si orijinal görüntüler olduğu ve bunlardan 6'sı benek gürültüsü eklenmiş görüntüler olarak, 1'i ise gauss gürültüsü eklenmiş görüntü olarak etiklendiği görülmektedir. Bunun yanı sıra, benek görüntülü görüntülerden 3'ü gauss gürültülü görüntüler olarak, tuz-biber gürültülü görüntülerden 1'i orijinal, 1'i ise benek gürültülü olarak etiketlenmiştir. Sonuçta gauss gürültüsü eklenmiş görüntülerinin tamamının doğru tahmin edildiği, ancak diğer bazı türlerinde gauss gürültüsü olarak etiketlenmesi nedeni ile bu doğru tahmin başarısına %100 gauss gürültüsü tahmini gerçekleştirilebiliyor demenin mümkün olmadığı görülmüştür. Eğer diğer türlerden hiçbiri gauss olarak etiketlenmemiş olsaydı, o zaman bu başarı tam olarak sağlanmış olacaktı.

5.2.3 Üç (3) katman eklenerek oluşturulan VGG16 mimarisi



Şekil 5.16 RMSProp optimizasyon algoritması için üç katmanlı mimari; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği

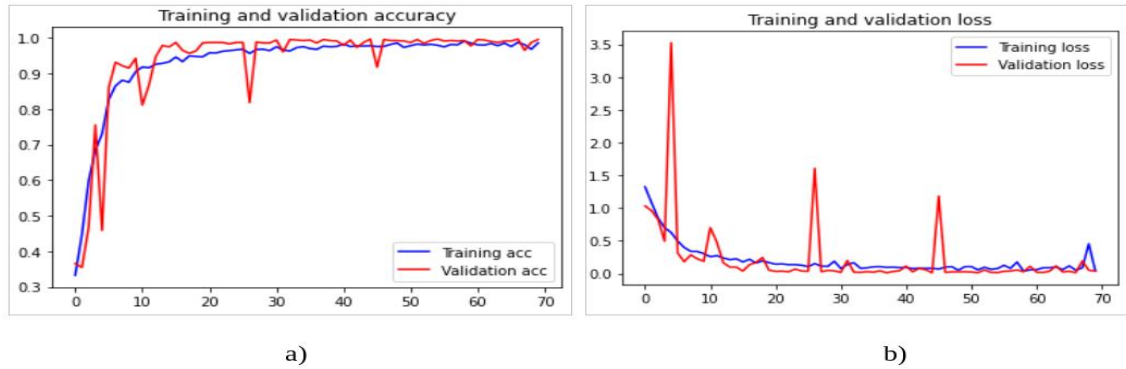
RMSProp optimizasyon algoritması ile üç (3) katman kullanılarak oluşturulmuş mimari için eğitim veri seti ve doğrulama veri seti doğruluk değerlerine ve kayıp değerlerine Şekil 5.16'da yer verilmiştir. Şekil 5.16 incelendiğinde 70 turda eğitim veri seti için doğruluk değeri %98 iken doğrulama veri seti için doğrulama doğruluk değeri %99 olarak elde edilmiş; kayıp değerleri ise eğitim veri seti için 0.12 doğrulama veri seti için ise 0.06 olarak hesaplanmıştır. Eğitilen model ile 960 görüntüden oluşan test veri seti kullanılarak test yapılmıştır ve sonuçlar incelendiğinde 41 görüntünün hatalı bulunduğu, doğruluk değerinin ise %96 olarak hesaplandığı görülmüştür. Şekil 5.17'de test veri setinde yer alan ve yanlış tespit edilmiş görüntülere örnek verilmiştir.



Şekil 5.17 RMSProp optimizasyon algoritması ile oluşturulan üç (3) katmanlı modelin test sonucunda hatalı bulunan görüntülere örnekler

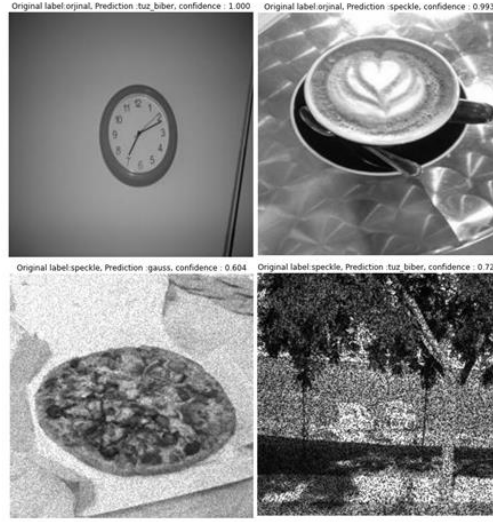
Test sonuçlarına göre, toplamda bulunan 41 hatalı görüntüden 33'ünün benek gürültüsü eklenmiş görüntü olarak etiklendiği gözlemlenmiştir. Benek gürültüsü eklenmiş görüntü olarak hatalı etiketlenen görüntülerden 14'ü gauss gürültüsü eklenmiş görüntü, 12'si tuz-biber gürültüsü eklenmiş görüntü ve diğer 12'si ise orijinal görüntüdür. Bunun yanında 2 gauss gürültüsü eklenmiş görüntü tuz-biber gürültüsü eklenmiş görüntü olarak, 3 tuz-biber gürültüsü eklenmiş görüntü orijinal görüntü olarak ve 1 benek gürültüsü eklenmiş görüntü gauss gürültüsü eklenmiş görüntü olarak hatalı şekilde etiketlenmiştir. Her bir gürültü türünde bazı görüntülerin çoğunlukla benek gürültüsü eklenmiş görüntüler ile karıştırılmış olduğu yani modelin doğru etiketlemeyi tespit edemediğinde benek gürültüsü eklenmiş olarak etiketleme yaptığı gözlemlenmiştir.

5.2.4 Beş (5) katman eklenerek oluşturulan VGG16 mimarisi



Şekil 5.18 RMSProp optimizasyon algoritması için beş katmanlı mimari; a) Eğitim ve doğrulama veri seti doğruluk değeri grafiği, b) Eğitim ve doğrulama veri seti kayıp değeri grafiği

RMSProp optimizasyon algoritması ile beş (5) katman kullanılarak oluşturulmuş mimari için eğitim veri seti ve doğrulama veri seti doğruluk değerlerine ve kayıp değerlerine Şekil 5.18'de yer verilmiştir. Şekil 5.18 incelendiğinde 70 turda eğitim veri seti için doğruluk değeri %99 ve doğrulama veri seti için doğrulama doğruluk değeri %100 olarak elde edilmiş; kayıp değerleri ise eğitim veri seti ve doğrulama veri seti için 0.04 olarak hesaplanmıştır. Eğitilen model ile 960 görüntüden oluşan test veri seti kullanılarak test yapılmıştır ve sonuçlar incelendiğinde 23 görüntünün hatalı bulunduğu, doğruluk değerinin ise %98 olarak hesaplandığı görülmüştür. Şekil 5.19'da test veri setinde yer alan ve yanlış tespit edilmiş görüntüler verilmektedir.



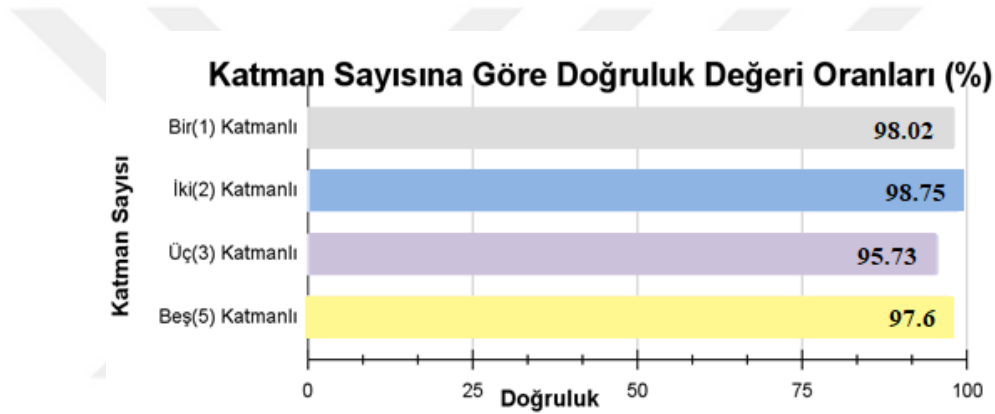
Şekil 5.19 RMSProp optimizasyon algoritması ile oluşturulan beş (5) katmanlı modelin test sonucunda hatalı bulunan görüntülere örnekler

Test sonuçlarına göre, 23 hatalı görüntüden 12'sinin benek gürültüsü eklenmiş görüntüler, 6'sının orijinal görüntüler, 4'ünün gauss gürültüsü eklenmiş görüntüler ve 1'inin de tuz-biber gürültüsü eklenmiş görüntü olduğu gözlemlenmiştir. Bu hatalı etiketlenen görüntülerden, benek gürültüsü eklenmiş görüntülerin 8'i tuz-biber gürültüsü eklenmiş görüntü, 4'ü gauss gürültüsü eklenmiş görüntü olarak; orijinal görüntülerin 4'ü benek gürültüsü eklenmiş görüntü, 2'si tuz-biber gürültüsü eklenmiş görüntü olarak; gauss gürültüsü eklenmiş görüntülerin 4'ü benek gürültüsü eklenmiş görüntü olarak ve tuz-biber gürültüsü eklenmiş 1 görüntünün ise orijinal görüntü olarak etiketlendiği görülmüştür. İncelemelere göre yüksek oranda benek gürültüsüne sahip görüntülerin tuz-biber gürültüsü eklenmiş görüntüler olarak etiketlendiği, orta düzeyde benek gürültüsüne sahip görüntülerin ise gauss gürültüsü olarak etiketlendiği gözlemlenmiştir. Bazı gauss gürültülü görüntülerin benek gürültülü görüntüler ile, bazı orijinal görüntülerin ise benek gürültülü ve tuz-biber gürültülü görüntüler ile karıştırıldığı tespit edilmiştir. Ek 2'de RMSProp optimizasyon algoritması için katman karşılaştırmalarına ait hatalı görüntülere yer verilmiştir.

Genel Değerlendirme

Çizelge 5.4 Kullanılan katman sayısına göre RMSProp optimizasyon algoritmasının doğrulukları

Sıra Numarası	Optimizasyon Algoritması	Doğruluk
1	Bir(1) Katmanlı	%98.02
2	İki(2) Katmanlı	%98.75
3	Üç(3) Katmanlı	%95.73
4	Beş(5) Katmanlı	%97.6



Şekil 5.20 Çizelge 5.4'ün grafik gösterimi

Çizelge 5.4'de ve Şekil 5.20'de RMSProp optimizasyon algoritması kullanılarak oluşturulan VGG16 modelinin kullanılan katman sayılarına ait doğruluk değerlerine yer verilmektedir. Çizelge 5.4 ve Şekil 5.20 incelendiğinde iki katman kullanılarak oluşturulan mimarinin diğerlerine göre daha iyi doğruluk sağladığı görülmektedir. Doğruluk oranları bir katman kullanılan mimari için %98.02, iki katman kullanılan mimari için %98.75, üç katman kullanılan mimari için %95.73 ve son olarak beş katman kullanılan mimari için %97.6'dır.

Çizelge 5.5 Katman sayısına bağlı modellerin karışıklık matrisleri

Gerçek Etiket			Tahmin Etiket			
			Gauss	Orijinal	Benek	Tuz-biber
Gerçek Etiket	Bir katman	Gauss	298	0	2	0
		Orijinal	1	54	5	0
		Benek	7	0	292	1
		Tuz-biber	0	1	2	297
		Toplam Hatalı Bulunan Sayısı: 19				
Gerçek Etiket	İki katman	Gauss	300	0	0	0
		Orijinal	1	53	6	0
		Benek	3	0	297	0
		Tuz-biber	0	1	1	298
		Toplam Hatalı Bulunan Sayısı: 12				
Gerçek Etiket	Üç katman	Gauss	284	0	14	2
		Orijinal	0	51	9	0
		Benek	1	0	299	0
		Tuz-biber	0	3	12	285
		Toplam Hatalı Bulunan Sayısı: 41				
Gerçek Etiket	Beş katman	Gauss	296	0	4	0
		Orijinal	0	54	4	2
		Benek	4	0	288	8
		Tuz-biber	0	1	0	299
		Toplam Hatalı Bulunan Sayısı: 23				

Çizelge 5.5’de yer alan katman sayılarına bağlı modellerin karışıklık matrisleri incelendiğinde her birinde farklı gürültü türlerine sahip görüntülerin farklı oranlarda yanlış tahmin edilmiş olduğu görülmektedir. Bir (1) katman eklenerek oluşturulan mimaride orijinal görüntüler ve benek gürültüsü eklenmiş görüntüler ağırlıklı olarak yanlış etiketlenmişken iki (2) katmanlı da toplam hata sayısının diğerlerinden az olması ile birlikte çoğunluklu hata sayısı orijinal görüntülere aittir. Bununla birlikte üç (3) katmanlı mimaride orijinal, gauss ve tuz-biber gürültülü görüntüler ağırlıklı olarak hatalı tespit edilirken hata sayısındaki artış fazladır. Katman sayısının üç (3) katmandan beş (5) katmana çıkarılması ile ise hata sayısında tekrardan bir azalma gözlemlenmektedir. Denenen dört farklı katman sayısı ile oluşturulan mimarilerden en iyisinin iki (2) katman eklenerek oluşturulan mimari olduğu test sonuçlarının değerlendirilmesi ile açığa çıkmaktadır.

Çizelge 5.5’deki karışıklık matrisinin incelenmesi sonucu elde edilen bilgiler arasında bir (1) katman eklenerek oluşturulan mimaride çoğunlukla orijinal görüntülerin benek gürültüsü eklenmiş görüntüler ve benek gürültüsü eklenmiş görüntülerin gauss gürültüsü eklenmiş görüntüler olarak tespit edildiği; iki (2) katman eklenerek oluşturulan mimaride çoğunlukla orijinal görüntülerin benek gürültüsü eklenmiş görüntüler olarak tespit edildiği; üç (3) katman eklenerek oluşturulan mimarilerde çoğunlukla gauss gürültüsü eklenmiş görüntülerin benek gürültüsü eklenmiş görüntüler, tuz-biber gürültüsü eklenmiş görüntülerin benek gürültüsü eklenmiş görüntüler ve orijinal görüntülerin benek gürültüsü eklenmiş görüntüler olarak hatalı tespit edildiği; beş (5) katman eklenerek oluşturulan mimarilerde çoğunlukla gauss gürültüsü eklenmiş görüntülerin benek gürültüsü eklenmiş görüntüler, benek gürültüsü eklenmiş görüntülerin gauss gürültüsü eklenmiş görüntüler ve orijinal görüntülerin benek gürültüsü eklenmiş görüntüler olarak hatalı tespit edildiği yer almaktadır.

Çizelge 5.6 Katman sayısına göre modellerin performans ölçümleri

Optimizasyon Algoritmaları	Performans Ölçümleri	Kesinlik	Duyarlılık	F1 Skor
Bir Katman	Makro Ort.	0.98	0.96	0.97
	Ağırlıklı Ort.	0.98	0.98	0.98
İki Katman	Makro Ort.	0.99	0.97	0.98
	Ağırlıklı Ort.	0.99	0.99	0.99
Üç Katman	Makro Ort.	0.96	0.94	0.95
	Ağırlıklı Ort.	0.96	0.96	0.96
Beş Katman	Makro Ort.	0.98	0.96	0.97
	Ağırlıklı Ort.	0.98	0.98	0.98

Çizelge 5.6’da modele ait performans ölçümleri görülmektedir. Katman sayılarına bağlı oluşturulan mimarilere ait kesinlik, duyarlılık ve F1 skorun yer aldığı tablo incelendiğinde makro ortalama ve ağırlıklı ortalama değerlerinin birbirlerine yakın olduğu görülmektedir.

Çizelge 5.7 Literatürde yer alan bazı örneklerin yapılan çalışma ile karşılaştırması

Yazar	Yıl	Method	Veri seti	Amaç	Doğruluk	PSNR*	MSE*	Ort. Hata
Burger ve ark.	2012	MLP	LamelMe + BSDS	Gürültü giderme	-	Farklı gürültü oranlarına göre ort. 30.18	-	-
Magud ve ark.	2016	Değiştirilmiş medyan filtresi	Standart Benchmark ultrason görüntüleri (karın - boyun)	Gürültü giderme	-	Karın görüntüsü: 17.56, Boyun görüntüsü: 31.75	Karın görüntüsü: 43.44, Boyun görüntüsü: 136.01	-

Çizelge 5.7 (Devam) Literatürde yer alan bazı örneklerin yapılan çalışma ile karşılaştırması

Değirmenci ve ark.	2018	Gradyan anahtarlama Gauss görüntü filtresi	Kameraman ve Lena görüntüleri	Gürültü giderme	-	Kameraman ort. 19.45, Lena ort. 21.13	Kameraman: ort. 804.58, Lena ort. 575.89	-
Güraksın	2018	k-ortalamlar algoritması	-	Gürültü giderme	-	Farklı gürültü oranlarına göre ortalama 94.85	-	-
Kaur ve Gandhi	2019	VGG16+ Transfer öğrenme	20 normal + 140 anormal beyin görüntüsü	Görüntü sınıflandırma	%100	-	-	-
Sil ve ark.	2019	VGG16, Inception-v3	BSDS300'den rastgele alınmış 14000 görüntü	Gürültü sınıflandırma	%67.99, %77.08	-	-	-
Tassano ve ark.	2019	FFDNet, DnCNN	CBSD68 - Kodak24 renkli görüntüler	Gürültü giderme	-	FFDNet: ort. 29.52, DnCNN: ort. 30.50	-	-
Küpeli ve Bulut	2020	Geleneksel filtreler (Mean, Medyan ve Gauss)	4 temel bilindik görüntü (airplane.png, baboon.png, lena.png, peppers.png)	Gürültü giderme	-	Her filtrenin her görüntü için ve her gürültü türü için ayrı değerleri mevcuttur.	-	-
İbrahim	2020	ESA tabanlı gürültü giderme yöntemi	Göğüs röntgen görüntüleri	Gürültü giderme	-	Ortalama 28.100	-	-

Çizelge 5.7 (Devam) Literatürde yer alan bazı örneklerin yapılan çalışma ile karşılaştırması

Pimpalkhute ve ark.	2021	DWT	Canlı veri kümesi (günlük hayattaki çeşitli nesne ve yapılar)	Gürültü giderme	-	-	-	0.6488
Çalışma*	2022	VGG16(yeni tasarım)+ Transfer öğrenme	Toplam 3840 adet günlük hayattan çeşitli nesne, yapı ve doğa görüntüleri	Gürültü tahmini	%98.75	-	-	-

*Literatürde yer alan çalışmalarda elde edilen PSNR ve MSE değerleri, bazı çalışmalarda farklı gürültü oranları veya farklı görüntüler için birden fazla değer verilmiş olması sebebiyle tüm sonuçların ortalamaları alınarak hesaplanmış ve bu çalışmaya o şekilde yerleştirilmiştir. Fazla miktarda sonuç değerine sahip olan çalışmalar için genel bir değerlendirme cümlesi yazılmıştır. Mevcut çalışmanın sonuçları ise en iyi sonucu veren algoritma üzerinden değerlendirilmiştir.

Geleneksel filtre yöntemlerinden yeni tasarlanan pek çok filtre yönteminin kullanılmasına kadar birçok gürültü giderme çalışması mevcuttur. Ancak gürültü türünün doğru şekilde tespit edilmesine ilişkin çalışma sayısı az bulunmaktadır. Çizelge 5.7’de literatürde yer alan bazı örneklerin yapılan çalışma ile karşılaştırmalarına yer verilmiştir. İnceleme sonucunda gürültü giderme çalışmalarının literatürde önemli bir yer kapladığı ve gürültü tahmini çalışmalarının da gürültü giderme çalışmalarına katkı sağlamayı hedeflediği gözlemlenmiştir. Çizelge 5.7’ye göre çalışmaya en yakın literatür örneği Sil ve ark. (2019) tarafından gerçekleştirilmiş olan gürültü sınıflandırma çalışmasıdır. Yalnızca doğruluk değerleri baz alınarak yapılan çalışma da VGG16 %67.99 doğruluk göstermiştir. Yapılan çalışmada ise VGG16’daki değişiklikler ile tasarlanan ESA modeli %98.75 doğruluk oranına sahiptir. Başarım ölçütü olarak doğruluk değerine ek kesinlik, duyarlılık ve F1 skorun maksimum ortalama ve ağırlıklı ortalama değerleri de ele alınmıştır. Çalışmadaki kesinlik değerine ait maksimum ve ağırlıklı ortalama %99; duyarlılık değerinde ait maksimum ortalama %97, ağırlıklı ortalama %99; F1 skora ait maksimum ortalama %98, ağırlıklı ortalama %99 olarak elde edilmiştir. Yapılan çalışma ile tasarlanan modelin literatürdeki örnekten başarılı olduğu gözlemlenmiştir.

6. SONUÇ ve ÖNERİLER

Görüntüleri gürültülerden arındırmak için kullanılacak olan filtrelerin, gürültü türüne göre oluşturulması gerekmektedir. Bunun yanında gürültü türünün doğru şekilde tespit edilememesi görüntünün gürültüden arındırılması için kullanılan yöntemlerin de başarısını olumsuz etkilemektedir. Eğer gürültü türünün tahmini doğru bir şekilde gerçekleştirilebilir ise görüntüye uygulanacak arındırma yöntemi gürültünün türüne göre belirleneceği için başarı oranını da artıracaktır. Bu sebeple gürültü türünün doğru bir şekilde tahmin edilmesi, uygun filtre kullanımını sağlamak ve başarı oranını artırmak için önem arz etmektedir.

Artık sıklıkla görüntü işleme çalışmaları için kullanılan ESA mimarilerinin gürültü tahmini için hangi optimizasyon algoritması ile daha verimli sonuçlar verebileceğine örnek teşkil eden bu çalışma, yeni bir görüntü tanıma çalışması yapılacağı durumlarda optimizasyon algoritması seçimine ilişkin yön gösterici olacaktır. Optimizasyon algoritmalarının başarı oranına ilişkin yapılmış bir çalışmanın ilerideki çalışmalar için uygun optimizasyon algoritmasının tercih edilebilmesinde yardımcı olacağı düşünülmektedir.

Transfer öğrenme ve VGG16 ile oluşturulan ESA mimarisi için literatürdeki 5 farklı optimizasyon algoritmasını kullanarak her biri için ayrı ayrı eğitim ve test işlemleri gerçekleştirilmiştir. Daha sonra kullanılan RMSProp, Adam, Adagrad, Adadelat ve SGD optimizasyon algoritmaları için eğitim sonucunda elde edilen kayıp ve doğruluk değerlerinin ve test sonucunda çıkan başarı oranlarının karşılaştırılması gerçekleştirilmiştir.

Yapılan test sonuçlarına göre RMSProp, Adam, Adagrad, Adadelat ve SGD optimizasyon algoritmaları kullanılarak oluşturulan 5 farklı VGG16 modeline ait doğruluk değerleri, sırasıyla RMSProp optimizasyon algoritması için %98.75, Adam optimizasyon algoritması için %98.44, Adagrad optimizasyon algoritması için %97.29, Adadelat optimizasyon algoritması için %89.38 ve SGD optimizasyon algoritması için %57.71 şeklinde olduğu gözlemlenmiştir. RMSProp optimizasyon algoritması en iyi

doğruluk oranına sahip model olmuştur. Toplam hatalı görüntü sayıları incelendiğinde en iyi sonucun 12 hatalı görüntü ile RMSProp optimizasyon algoritması kullanılarak oluşturulan mimaride olduğu gözlemlenmiş olup sırasıyla 15 hata ile Adam, 26 hata ile Adagrad, 102 hata ile Adadelta ve 406 hata ile SGD optimizasyon algoritmaları olarak hata sayıları artmaktadır.

Bununla birlikte optimizasyon algoritmalarında en yüksek değeri veren RMSProp optimizasyon algoritması için ESA mimarisindeki katman sayısının etkisini ölçmek için bir, iki, üç ve beş katman ile oluşturulmuş mimariler başarımları ve hatalı görüntü sayıları ile karşılaştırılmıştır. Sonuçlar katman sayısının da model başarısında etkili olduğunu göstermiştir.

RMSProp, Adam, Adagrad, Adadelta ve SGD optimizasyon algoritmaları için yapılan karşılaştırmalar ve RMSProp optimizasyon algoritması kullanılarak oluşturulan farklı katman sayılarındaki mimariler için yapılan karşılaştırmalara dair tüm sonuçlar incelendiğinde aslında görüntülere eklenen gürültülerin oranlarının da optimizasyon algoritmalarının başarımları üzerinde farklı etkileri olabileceği gözlemlenmiştir. Bazen gürültü oranı az olan bir görüntü orijinal olarak etiketlenebilirken, bazen de gürültü oranı fazla olan bir görüntü diğer gürültü türleri ile karıştırılabilmektedir. Katman sayılarındaki farklılık ise toplam hata sayısı oranını etkileyebilmektedir.

Yapılan çalışma ile gürültü tahmini için ilerleyen çalışmalarda hangi optimizasyon algoritmalarının kullanımının daha etkili olacağına dair fikir edinilebilmesine katkı sağlanmıştır. İleri de aynı çalışma renkli görüntüler üzerinde de denenebilir ve aynı başarımlarının elde edilip edilemeyeceği değerlendirilebilir. Bunun yanında gürültü tahminine ek olarak görüntülerdeki gürültü oranlarının önemi de anlaşılmış olup görüntülerdeki gürültü seviyesi tahminlerinin doğru ve pratik şekilde yapılması gerektiği düşünülmüştür. Bu işlemler ile görüntülerin gürültülerden arındırılması işlemlerine bir katkı daha sağlanabilir.

KAYNAKÇA

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... , Ghemawat, S., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.
- Akar, E., Kara S., Akdemir H., Kırış A., 2015. A MATLAB tool for an easy application and comparison of image denoising methods," 2015 Medical Technologies National Conference (TIPTEKNO), Bodrum, Turkey.
- Aktaş, A., 2020. Derin öğrenme yöntemleri ile görüntü işleme uygulamaları. (Y.Lisans Tezi), Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- Alom, M. Z., Taha, M. T., Yakopcic, C., Westberg, S., Sidike, P., Nasrin M. S., Essen, B. C., Awwal, A. A. S., Asari, V. K., 2018. The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. Cornell University Library, ArXiv, abs/1803.01164.
- Anonim, 2015. Lasagne Docs, Lasagne. <https://lasagne.readthedocs.io/en/latest/> (17.03.2022).
- Anonim, 2017. Derin öğrenme teknolojinin geleceğini nasıl şekillendiriyor? <https://www.exastax.com.tr/derin-ogrenme/derin-ogrenme-teknolojinin-gelecegini-nasil-sekillendiriyor/> (6.4.2022).
- Anonim, 2020. Deep blue defeated Garry Kasparov in chess. <https://societyofai.medium.com/deep-blue-defeated-garry-kasparov-in-chess-b8a3f33cf519> (7.4.2022).
- Anonim a, 2021. CS231n Convolutional Neural Networks for Visual Recognition. Tutorial 1: Image Filtering. <https://ai.stanford.edu/~syueung/cvweb/tutorial1.html> (06.03.2022).
- Anonim b, 2021. Ulusal Yapay Zekâ Stratejisi 2021-2025. Ankara.
- Arora, D., Garg, M., Gupta, M., 2020. Diving deep in Deep Convolutional Neural Network. 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Noida, India.
- Ay Karakuş, B., 2018. Derin öğrenme ve büyük veri yaklaşımları ile metin analizi. (Doktora Tezi), Fırat Üniversitesi, Elazığ.
- Aydın, G., 2020. Evrimsel sinir ağları (Convolutional Neural Networks). <https://tr.linkedin.com/pulse/evri%C5%9Fimli-sinir-a%C4%9Flar%C4%B1-convolutional-neural-networks-gizem-ayd%C4%B1n> (7.4.2022).
- Azgınoğlu, N., 2018. Çok katmanlı algılayıcılar(Multi-Layer Perceptron). <http://www.nuhazginoglu.com/2018/05/15/cok-katmanli-algilayicilar-multi-layer-perceptron/> (6.4.2022).
- Bayrakdar, B., Doğan, E., 2021. Adaftif Sinyalize kavşaklar için derin öğrenme yaklaşımı ile gecikme modelleri geliştirilmesi. Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi, 13(2), 390-405.
- Brownlee,2017. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/> (17.04.2022).
- Brownlee, J., 2019. A Gentle Introduction to the Rectified Linear Unit(Relu). <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/> (08.03.2022).

- Brownlee, J., 2021a. Gradient descent with RMSProp from Scratch. <https://machinelearningmastery.com/gradient-descent-with-rmsprop-from-scratch/> (15.4.2022).
- Brownlee, J., 2021b. Gradient Descent With AdaGrad From Scratch. <https://machinelearningmastery.com/gradient-descent-with-adagrad-from-scratch/> (17.4.2022).
- Brownlee, J., 2021c. Code Adam Optimization Algorithm From Scratch. <https://machinelearningmastery.com/adam-optimization-from-scratch/> (17.4.2021).
- Brownlee, J., 2021ç. Gradient Descent With Adadelta from Scratch. <https://machinelearningmastery.com/gradient-descent-with-adadelta-from-scratch/> (17.4.2022).
- Budhiraja A., 2016. Dropout in (Deep) Machine learning. Medium. <https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5> (17.12.2021).
- Burger, H. C., Schuler, C. J. Harmeling, S., 2012. Image denoising: Can plain neural networks compete with BM3D?. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Providence, USA.
- Çalışkan, 2019. Cezeri'nin Su Saatleri. <https://medium.com/@malicaliskan/cezerinin-su-saatleri-a8e56c84b80a> (8.4.2022).
- Carneiro, T., Medeiros Da Nóbrega, R.V., Nepomuceno, T., Bian, G., de Albuquerque, V.H., & Filho, P. (2018). Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. IEEE Access, 6, 61677-61685.
- Cengil, E. ve Çınar, A., 2019. Classification of Human Driving Behaviour Images Using Convolutional Neural Network Architecture, International Conference on Hybrid Artificial Intelligence Systems, 264-274.
- Collobert, R., Bengio, S., & Mariéthoz, J., 2002. Torch: a modular machine learning software library.
- Çarkacı, N., 2018. Derin öğrenme uygulamalarında en sık kullanılan hiper-parametreler. <https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4> (4.5.2022).
- Çınar, A. C., 2017. MATLAB ile Derin Öğrenmeye Giriş. <https://www.ahmetcevahircinar.com.tr/2017/08/09/matlab-ile-derin-ogrenmeye-giris-introducing-deep-learning-with-matlab/> (7.3.2022).
- Das, S., 2017. CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more. <https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5> (5.4.2022).
- Daşkın, Y. Ç., Balık Şanlı F., Özkan, T., Kartal, H., 2018. Derin öğrenme yöntemleri ile yüz görüntülerinden yaş kestirimi. VII. Uzaktan Algılama- CSB Sempozyumu(UZAL-CBS 2018), Eskişehir.
- Değirmenci, A., Çankaya İ., Demirci R., 2018. Gradyan Anahtarlmalı Gauss Görüntü Filtresi. Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 6 (1), 196-215.
- Demiraslan, V., & Derin, M., 2019. Toplum 5.0'da Robotik ve Dünya Siberetik Bilimi Kurucusu El-Cezeri. 11. International Balkan and Near Eastern Social Sciences Congress Series..

- Doğan, F., Türkoğlu, İ., 2018. Derin Öğrenme Algoritmalarının Yaprak Sınıflandırma Başarımlarının Karşılaştırılması, Sakarya University Journal of Computer and Information Sciences, 1.
- Doğan, T., Yapay sinir ağları ve tek katmanlı ağlarda öğrenme. LinkedIn. <https://tr.linkedin.com/pulse/yapay-sinir-a%C4%9Flar%C4%B1-ve-tek-katmanl%C4%B1-a%C4%9Flarda-%C3%B6%C4%9Frenme-tanju-do%C4%9Fan> (6.4.2022).
- Doğan, F., Türkoğlu, İ., 2019. Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme, DÜMF Mühendislik Dergisi, 10 (2) ,409-445.
- Ergin T., 2018. Evrişimsel Sinir Ağında Niçin Dropout Kullanmamalısınız. Medium. <https://medium.com/@tuncerergin/evrisimsel-sinir-aginda-nicin-dropout-kullanmamalısınız-7e31941f8bb0> (17.12.2021).
- Fan, J., Lee, J. H., Lee, Y., 2021. Application of Transfer Learning for Image Classification on Dataset with Not Mutually Exclusive Classes. 2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Jeju, Korea (South).
- Fırıldak, K., Talu, M. F., 2019. Evrişimsel Sinir Ağlarında Kullanılan Transfer Öğrenme Yaklaşımlarının İncelenmesi. Anatolian Journal of Computer Science, 4(2), 88-95.
- Géron A., 2019. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc., 510, Amerika Birleşik Devletleri.
- Goodfellow, I. J., Warde-Farley, D., Lamblin, P., Dumoulin, V., Mirza, M., Pascanu, R., and Bengio, Y., 2013. Pylearn2: a Machine Learning Research Library, arXiv preprint arXiv:1308.4214.
- Gülal Ö., 2018. Yapay zeka, derin öğrenme ve makine öğrenmesi arasındaki farklılıklar. <http://ozhangulal.com/hayat/yapay-zeka-derin-ogrenme-ve-makine-ogrenimi-arasindaki-farkliliklar/> (7.4.2022).
- Gündüz, G., Cedimoğlu, İ. H., 2019. Derin Öğrenme Algoritmalarını Kullanarak Görüntüden Cinsiyet Tahmini. Sakarya University Journal of Computer and Information Sciences, 2(1), 9-17.
- Güraksın G. E., 2018. Tuz Biber Gürültülerinin Giderilmesi için k-Ortalama Algoritması Tabanlı Filtre Tasarımı. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 22 (2), 972-978.
- Hamza, A. B., Krim H., 2001. Image denoising: A nonlinear robust statistical approach. IEEE Transactions on Signal Processing 49(12), 3045–3054.
- Hu, L., 2020. Image Noise Estimation with Using Pre-trained Conventional Neural Network. 2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE).
- Heller, M., 2020. Deeplearning4j: Deep learning and ETL for the JVM. <https://www.infoworld.com/article/3567055/deeplearning4j-deep-learning-and-etl-for-the-jvm.html#:~:text=Eclipse%20Deeplearning4j%20is%20an%20open,serve%20as%20the%20Python%20API> (5.4.2022).
- Hoomod, H. K., Dawood, S. H., 2017. Fast images denoising based on modify CNN and noise estimation. 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), Baghdad, Iraq.

- Hu, L., 2020. Image noise estimation with using pre-trained conventional neural network. 2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), Shenyang, China.
- Immerkær, J., 1996. Fast noise variance estimation. *Computer Vision Image Understanding*. 64 (2), 300–302.
- İbrahim, M. S., Evrişimsel sinir ağları tabanlı gürültü giderici otomatik kodlayıcı kullanarak görüntüyü gürültüden arındırma. (Y.Lisans Tezi), Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.
- İnik, Ö., Ülker, E., 2017. Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri. *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6(3), 85-104.
- İtik, B. M., 2020. Tarihin ilk programcısı: Ada Lovelace. <https://medium.com/t%C3%BCrkiye/tarihin-i%C3%87lk-bilgisayar-programc%C4%B1s%C4%B1-ada-lovelace-3b2e3a058a69#:~:text=Augusta%20Ada%20King%20ya%20da,bir%20Londra%20g%C3%BCn%C3%BCnde%20d%C3%BCnyaya%20gelmi%C5%9Ftir> (24.4.2022).
- Jain, V., Seung, H. S., 2008. Natural Image Denoising with Convolutional Networks. *NIPS*.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R.B., Guadarrama, S., Darrell, T., 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *Proceedings of the 22nd ACM international conference on Multimedia*.
- Kara, B., 2019. Perceptron Nedir? <https://medium.com/@buse.kara7998/perceptron-nedi%C3%87r-a15a7c8e5276> (6.4.2022).
- Karakuş, S., 2018. Derin öğrenme yöntemleri kullanılarak dijital deliller üzerinde adli bilişim incelemesi. (Y. Lisans Tezi), Fırat Üniversitesi, Elazığ.
- Kaur, S., 2015. Noise types and various removal techniques. *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, 4 (2), 226-230.
- Kaur, T., Gandhi, T. K., 2019. Automated Brain Image Classification Based on VGG16 and Transfer Learning. 2019 International Conference on Information Technology (ICIT), Bhubaneswar, India.
- Kayaalp, K., Süzen, A. A., 2018. Derin Öğrenme ve Türkiye’deki Uygulamaları. Iksad Yayınevi, ISBN 978-605-7510-53-2.
- Kın, Z. B., 2019. Türk işaret dili alfabesinin derin öğrenme yöntemi ile sınıflandırılması. (Y.Lisans Tezi), Başkent Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Kirana, K. C., Wibawanto, S., Hidayah, N., Cahyono, G. P., Asfani, K., 2019. Improved Neural Network using Integral-RELU based Prevention Activation for Face Detection. 2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE), Denpasar, Indonesia.
- Kızrak, A., 2018. Derine Daha Derine: Evrişimli Sinir Ağları. <https://ayyucekizrak.medium.com/deri%C3%87ne-daha-derine-evrişimli-sinir-ağları-2813a2c8b2a9> (08.03.2022).
- Kızrak A., 2019. Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması. <https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n->

- [kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-
cee17fd1d9cd](#) (09.03.2022).
- Kızrak, M. A., Bolat, B., 2018. Derin öğrenme ile kalabalık analizi üzerine detaylı bir araştırma. Bilişim Teknolojileri Dergisi, 11(3), 263-286.
- Kocaman, Ş., 2021. Grafik tasarım endüstrisinde yapay zekâ. Journal of Social and Humanities Sciences Research, 8(77), 3000-3016.
- Korkmaz, S., 2019. Small Drug Molecule Classification Using Deep Neural Networks. Türkiye Klinikleri Journal of Biostatistics, 11(2), 93-101.
- Kumar, A., 2020. Micro-average & Macro-average Scoring Metrics – Python. <https://vitalflux.com/micro-average-macro-average-scoring-metrics-multi-class-classification-python/> (18.4.2022).
- Kutlu, Ö., 2019. İnsansız hava aracı ile elde edilen görüntülerin derin öğrenme yöntemleri ile analizi. (Y.Lisans Tezi), Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- Kurt, F., 2018. Evrimsel sinir ağlarında hiper parametrelerin etkisinin incelenmesi. (Y.Lisans Tezi), Hacettepe Üniversitesi, Ankara.
- Küçük, D., Arıcı, N., 2018. Doğal dil işlemede derin öğrenme uygulamaları üzerine bir literatür çalışması. Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi, 2(2), 76-86.
- Küpeli, C., Bulut, F., 2020. Görüntüdeki tuz-biber ve gauss gürültülerine karşı filtrelerin performans analizleri. Haliç Üniversitesi Fen Bilimleri Dergisi, 3(2), 211-239.
- Lahmiri, S., Boukadoum, M., 2014. Biomedical image denoising using variational mode decomposition.in Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS) Proceedings, Lausanne, Switzerland.
- Leblebicioğlu, E., 2019. Matlab Nedir? Nerelerde Kullanılır? <https://muhendistan.com/matlab-nedir-nerelerde-kullanilir/> (8.5.2022).
- Li, G., Zhen, H., Jiao, F., Hao, T., Wang, D., Ni, K., 2021. Research on tobacco leaf grading algorithm based on transfer learning. 2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China.
- Liu D., 2017. A Practical Guide to ReLu. <https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7> (08.03.2022).
- M, R., 2019. Churning the Confusion out of the Confusion Matrix. <https://blog.clairvoyantsoft.com/churning-the-confusion-out-of-the-confusion-matrix-b74fb806e66> (18.4.2022).
- Madhavan, S., Ahmed, S., Rao, V., John, A., 2021. Compare deep learning frameworks. <https://developer.ibm.com/articles/compare-deep-learning-frameworks/> (5.4.2022).
- Mafi, M., Martin, H., Cabrerizo, M., Andrian, J., Barreto, A., Adjouadi, M., 2019. A comprehensive survey on impulse and Gaussian denoising filters for digital images. Signal Processing, 157, 236–260.
- Magud, O., Tuba, E., Bacanin, N., 2016. An algorithm for medical ultrasound image enhancement by speckle noise reduction. International Journal of Signal Processing, 1, 146–151.
- Murat, S., 2021. İnsansız Hava Aracı Görüntülerinden Derin Öğrenme Yöntemleriyle Nesne Tanıma. (Y.Lisans Tezi), Maltepe Üniversitesi. Lisansüstü Eğitim Enstitüsü, İstanbul.

- NVIDIA, 2017. NVcaffe Overview. https://docs.nvidia.com/deeplearning/frameworks/caffe-release-notes/overview_relnot.html#overview_relnot (3.4.2022).
- Öğündür, G., 2019. Doğruluk (Accuracy) , Kesinlik(Precision) , Duyarlılık(Recall) ya da F1 Score? <https://medium.com/@gulcanogundur/do%C4%9Fruluk-accuracy-kesinlik-precision-duyarl%C4%B1%C4%B1k-recall-ya-da-f1-score-300c925feb38> (15.4.2022).
- Özgür, S. B., 2021. Algoritmalar, yapay zeka, makine öğrenmesi, derin öğrenme ve uygulamaları: beşeri fayda üretiminin yazılımlar tarafından karşılanması. Ekonomi ve Yönetim Araştırmaları Dergisi, 10(1).
- NVIDIA Developer, 2022. NVIDIA DIGITS- Interactive Deep Learning GPU Training System. <https://developer.nvidia.com/digits> (5.4.2022).
- Phan, T. V., Sultana, S., Nguyen, T. G., Bauschert, T., 2020. Q - TRANSFER: A Novel Framework for Efficient Deep Transfer Learning in Networking. 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), Fukuoka, Japan.
- Pimpalkhute, V. A., Page, R., Kothari, A., Bhurchandi, K. M., Kamble, V. M., 2021. Digital Image Noise Estimation Using DWT Coefficients. IEEE Transactions on Image Processing, 30, 1962 - 1972.
- Ponomarenko, M., Miroshnichenko O., Lukin, V., Egiazarian, K., 2021. Blind Estimation and Suppression of Additive Spatially Correlated Gaussian Noise in Images. 2021 9th European Workshop on Visual Information Processing (EUVIP), Paris, Fransa.
- Rajeswari, S. S., Nair, M., 2021. A Transfer Learning Approach for Predicting Alzheimer's Disease. 2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), NaviMumbai, India.
- Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., Yang, G. Z., 2017. Deep Learning for Health Informatics. IEEE journal of Biomedical and Health Informatics, 21(1), 4-21.
- Ruder, S., 2016. An overview of gradient descent optimization algorithms. ArXiv, abs/1609.04747.
- Saporito, G., 2019. What is a perceptron?. <https://towardsdatascience.com/what-is-a-perceptron-210a50190c3b> (25.4.2022).
- Savaş, S., 2020. Derin Öğrenme. <https://medium.com/yapay-zeka-makine-%C3%B6%C4%9Frrenmesi-derin-%C3%B6%C4%9Frrenme/derin-%C3%B6%C4%9Frrenme-78e3f5d2765> (7.4.2022).
- Ser, G., Bati, C. T., 2019. Derin Sinir Ağları ile En İyi Modelin Belirlenmesi: Mantar Verileri Üzerine Keras Uygulaması. Yüzüncü Yıl Üniversitesi Tarım Bilimleri Dergisi, 29(3).
- Setiawan, W., Imam U. M., Rulaningtyas, R., 2019. Classification of neovascularization using convolutional neural network model. Telecommunication Computing Electronics and Control (TELKOMNIKA), 17(1), 463-472.
- Seyyarer, E., Ayata, F., Uçkan, T., Karcı, A., 2020. Derin Öğrenmede Kullanılan Optimizasyon Algoritmalarının Uygulanması Ve Kıyaslanması. Anatolian Journal of Computer Sciences, 5(2), 90-98.
- Sharma S., What the Hell is Perceptron? <https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53> (6.4.2022).

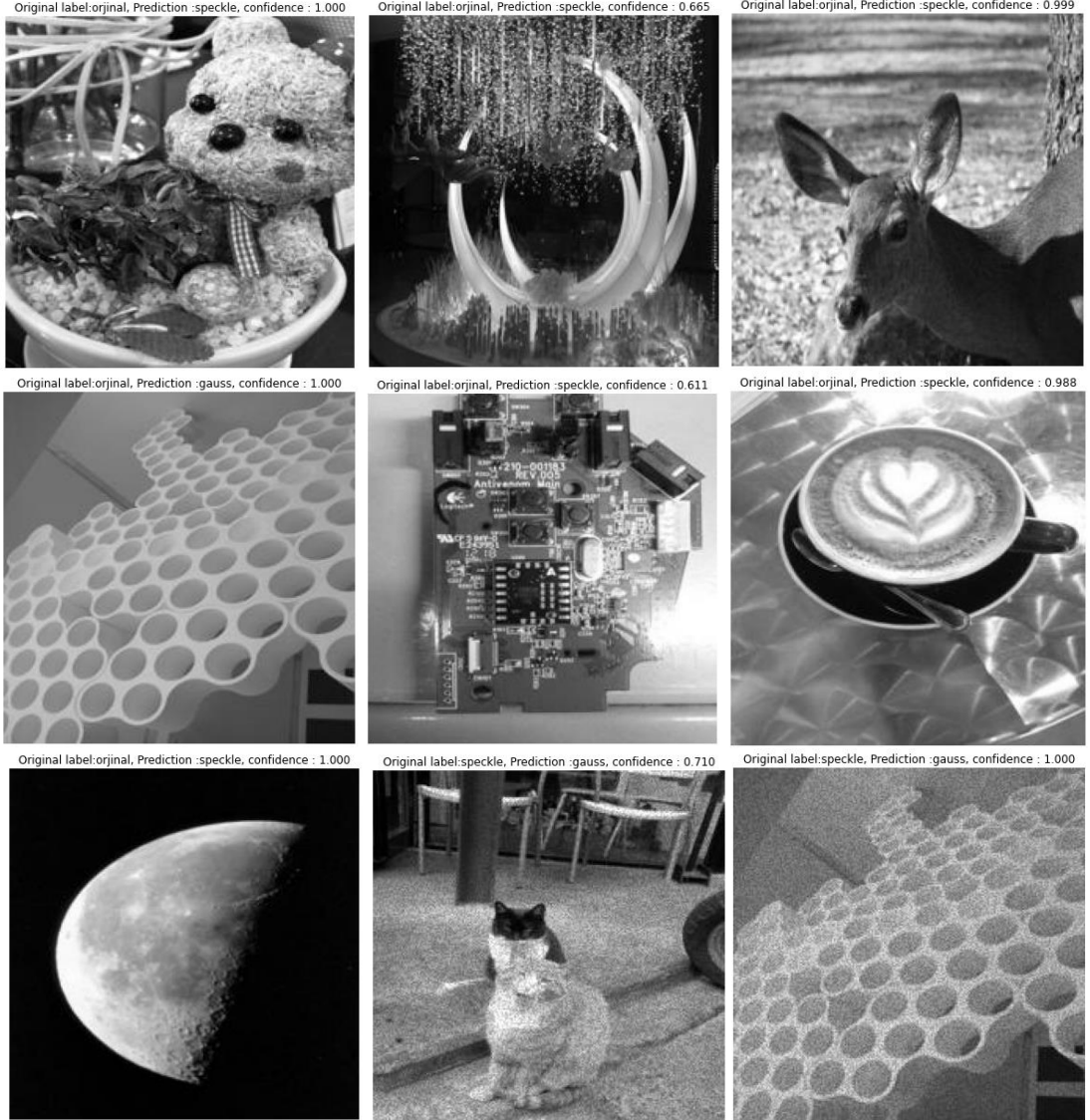
- Shen, G., Yuan, Y., 2019. On Theoretical Analysis of Single Hidden Layer Feedforward Neural Networks with Relu Activations. 2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Jinzhou, China.
- Sil, D., Dutta, A. Chandra, A., 2019. Convolutional Neural Networks for Noise Classification and Denoising of Images. TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India.
- Simonyan, K., Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. Visual Geometry Group, Department of Engineering Science, University of Oxford. arXiv:1409.1556v6.
- Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15 (2014) 1929-1958.
- Szegedy, C., Liu, W., Jia, Y. Q., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going Deeper with Convolutions. 2015 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr): 1-9.
- Şeker, A., Diri, B., Balık, H. H., 2017. Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. Gazi mühendislik Bilimleri Dergisi, 3(3), 47-64.
- Tai, S. C., Yang, S. M., 2008. A fast method for image noise estimation using laplacian operator and adaptive edge detection. 2008 3rd International Symposium on Communications, Control, and Signal Processing, St. Julians, Malta.
- Tan, Z., 2019. Derin öğrenme yardımıyla araç sınıflandırma. (Y.Lisans Tezi), Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.
- Tao, J., Gu, Y., Sun, J., Bie, Y., Wang, H., 2021. Research on vgg16 convolutional neural network feature classification algorithm based on Transfer Learning. 2021 2nd China International SAR Symposium (CISS), Shanghai, China.
- Tassano M., Delon, J., Veit, T., 2019. An analysis and implementation of the FFDNet image denoising method. Image Processing On Line, 9, 1–25.
- Uçar, A., Bingöl, M. S., 2018. Derin öğrenmenin Caffe kullanılarak grafik işleme kartlarında değerlendirilmesi. DÜMF Mühendislik Dergisi, 9(1), 39-49.
- Uss, M. L., Lukin, V. V., Vozel, B., Chehdi, K., 2020. Accuracy of Model-based and Learning-based Approaches for Image Noise Variance Estimation. 2020 IEEE Ukrainian Microwave Week (UkrMW), Kharkiv, Ukraine.
- Uysal, F., Hardalaç, F., Koç, M., 2018. Classification of T1 and T2 Weighted Magnetic Resonance Prostate Images Using Convolutional Neural Networks. 2018 Medical Technologies National Congress (TIPTEKNO), Magusa, Cyprus.
- Verma, R., Ali, J., 2013. A Comparative Study of Various Types of Image Noise and Efficient Noise Removal Techniques. International Journal of Advanced Research in Computer Science and Software Engineering, 10(3).
- Volaka, H. C., 2020. Cahit Arf- Makine Düşünebilir mi ve Nasıl Düşünebilir? <https://futurizm.org/cahit-arf-makine-dusunebilir-mi-ve-nasil-dusunebilir-1958-59/> (21.4.2022).
- Vrbančić, G., Podgorelec, V., 2022. Efficient ensemble for image-based identification of Pneumonia utilizing deep CNN and SGD with warm restarts. Expert Systems with Applications, 187.
- Wikipedia a, 2022. Mona Lisa. https://en.wikipedia.org/wiki/Mona_Lisa (8.4.2022).
- Wikipedia b, 2022. Leonardo's robot. https://en.wikipedia.org/wiki/Leonardo's_robot (8.4.2022).

- Yapıcı, M. M., Tekerek, A., Topaloğlu N., 2019. Derin öğrenme araştırma alanlarının literatür taraması. *Gazi Mühendislik Bilimleri Dergisi*, 5(3), 188-215.
- Yu, D., Eversole, A., Seltzer, M., Yao, K., Huang, Z., Guenter, B., ... , Droppo, J., 2016. An Introduction to Computational Networks and The Computational Network Toolkit. Microsoft Technical Report MSR-TR-2014-112.
- Zhang, Y., 2016. Theano: A Python Framework for Fast Computation of Mathematical Expressions, The Theano Development Team, arXiv preprint arXiv:1605.02688v1 [cs.SC].
- Zhang, K., Zuo, W., Chen, Y., Meng D., Zhang, L., 2017. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Process*, 26 (7), 3142–3155.
- Zhang, K., Zuo, W., Zhang L., 2018. FFDNet: Toward a fast and flexible solution for CNN-Based image denoising. *IEEE Transactions on Image Processing*, 27(9).
- Zoran, D., Weiss, Y., 2009. Scale invariance and noise in natural images. *Proceedings of the IEEE International Conference on Computer Vision*, Kyoto, Japan.

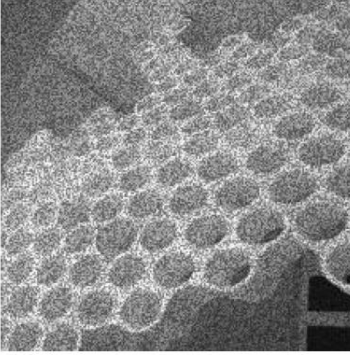
EKLER

Ek 1. Optimizasyon Algoritmaları için Hatalı Bulunan Görüntüler

RMSProp optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntüler:



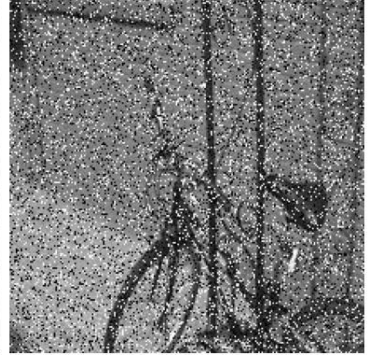
Original label: speckle, Prediction : gauss, confidence : 1.000



Original label: tuz biber, Prediction : orjinal, confidence : 0.923



Original label: tuz biber, Prediction : speckle, confidence : 0.684



Adam optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntüler:

Original label: gauss, Prediction : speckle, confidence : 0.836



Original label: gauss, Prediction : speckle, confidence : 0.702



Original label: gauss, Prediction : speckle, confidence : 1.000



Original label: gauss, Prediction : speckle, confidence : 0.845



Original label: gauss, Prediction : speckle, confidence : 0.995



Original label: orjinal, Prediction : speckle, confidence : 0.985



Original label: orjinal, Prediction : speckle, confidence : 0.919

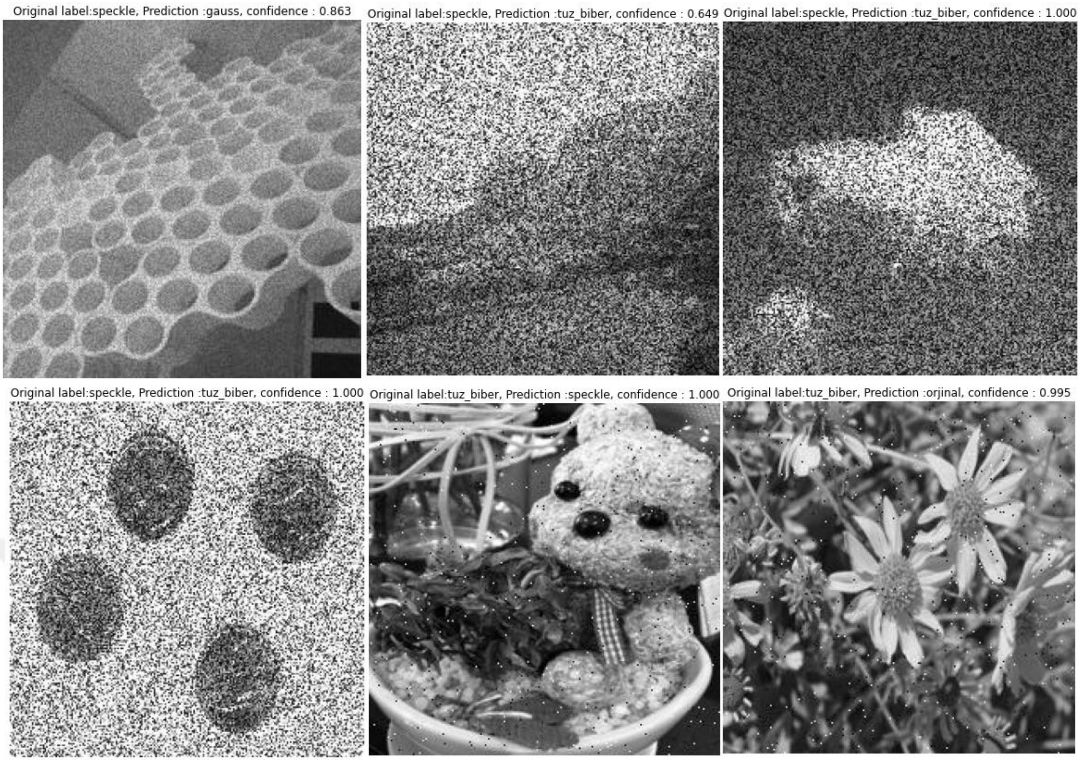


Original label: orjinal, Prediction : speckle, confidence : 0.861



Original label: orjinal, Prediction : speckle, confidence : 0.997





Adagrad optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntüler:



Original label:original, Prediction :gauss, confidence : 0.612



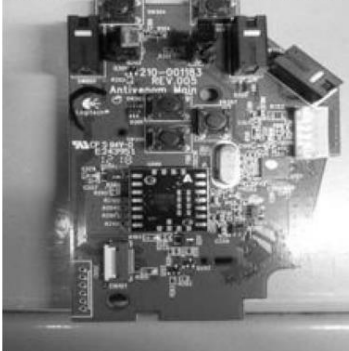
Original label:original, Prediction :speckle, confidence : 0.552



Original label:original, Prediction :speckle, confidence : 0.972



Original label:original, Prediction :speckle, confidence : 0.925



Original label:original, Prediction :speckle, confidence : 0.871



Original label:original, Prediction :speckle, confidence : 0.765



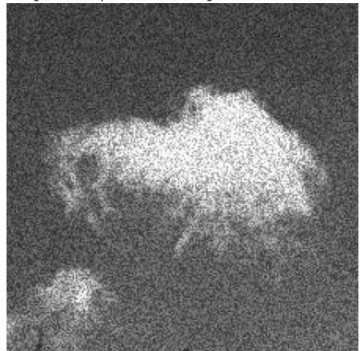
Original label:original, Prediction :speckle, confidence : 0.988



Original label:speckle, Prediction :gauss, confidence : 0.946



Original label:speckle, Prediction :gauss, confidence : 0.953



Original label:speckle, Prediction :tuz_biber, confidence : 0.800

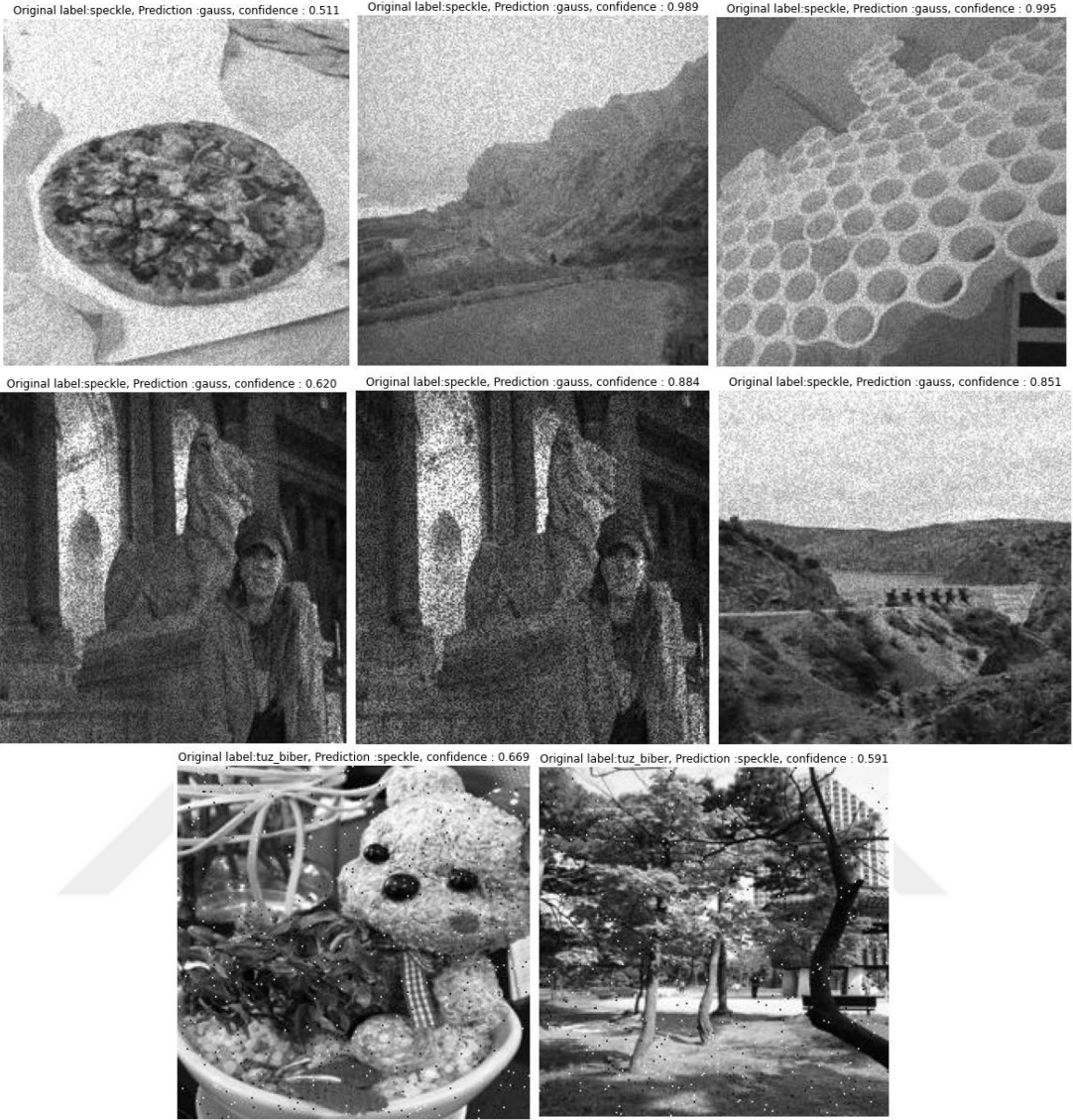


Original label:speckle, Prediction :tuz_biber, confidence : 0.986



Original label:speckle, Prediction :gauss, confidence : 0.912





Adadelta optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntülere örnekler:



Original label:original, Prediction :speckle, confidence : 0.723



Original label:original, Prediction :speckle, confidence : 0.874



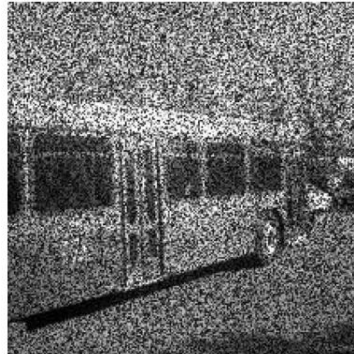
Original label:original, Prediction :speckle, confidence : 0.474



Original label:speckle, Prediction :tuz_biber, confidence : 0.540



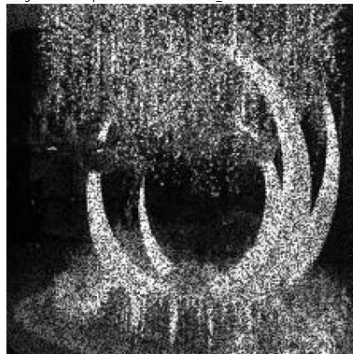
Original label:speckle, Prediction :tuz_biber, confidence : 0.537



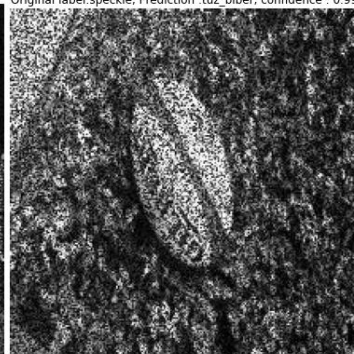
Original label:speckle, Prediction :tuz_biber, confidence : 0.645



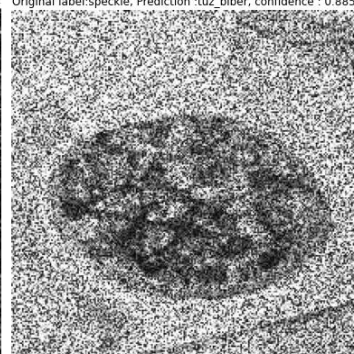
Original label:speckle, Prediction :tuz_biber, confidence : 0.551



Original label:speckle, Prediction :tuz_biber, confidence : 0.992



Original label:speckle, Prediction :tuz_biber, confidence : 0.885



Original label:speckle, Prediction :gauss, confidence : 0.967

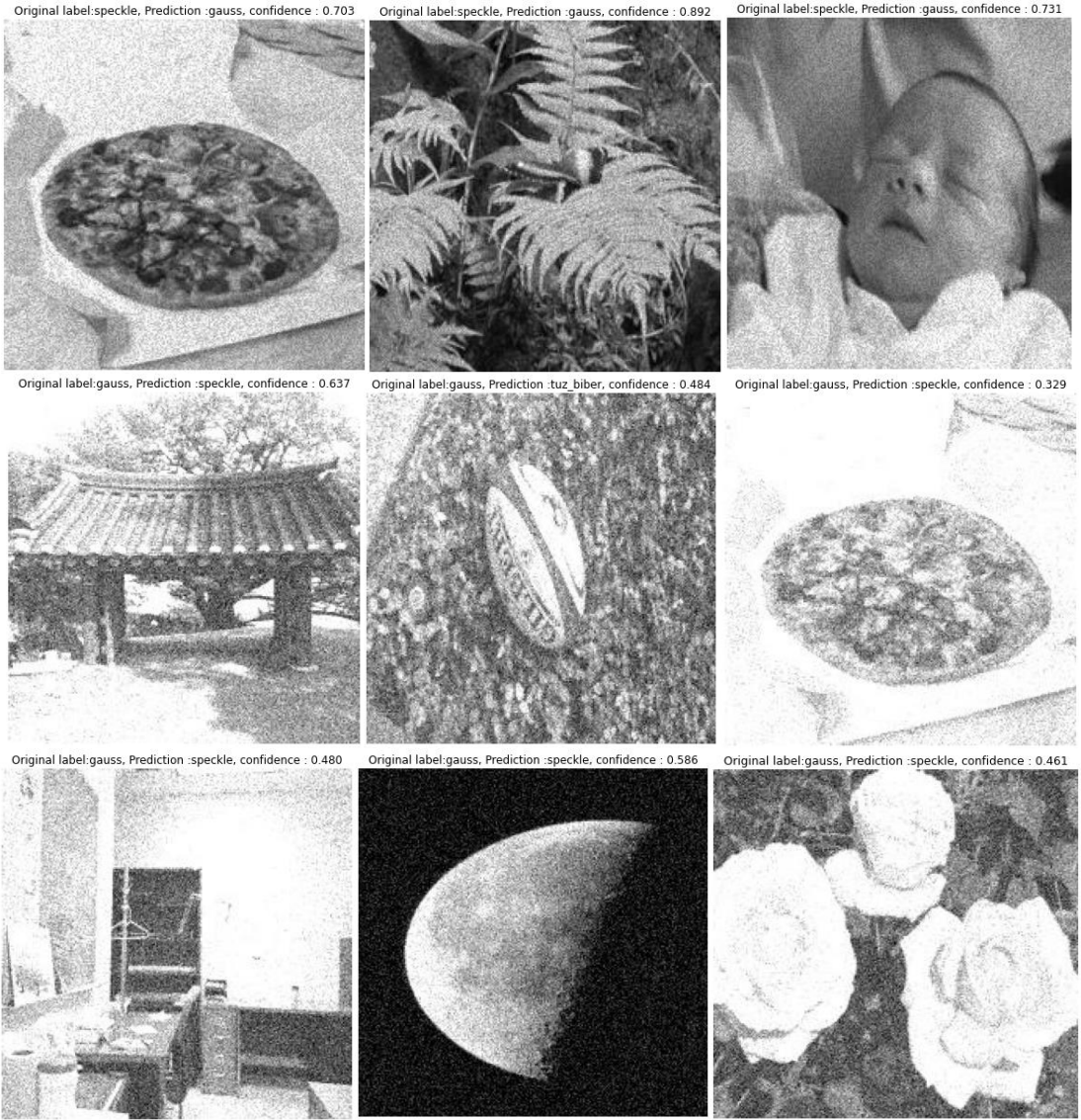


Original label:speckle, Prediction :gauss, confidence : 0.522

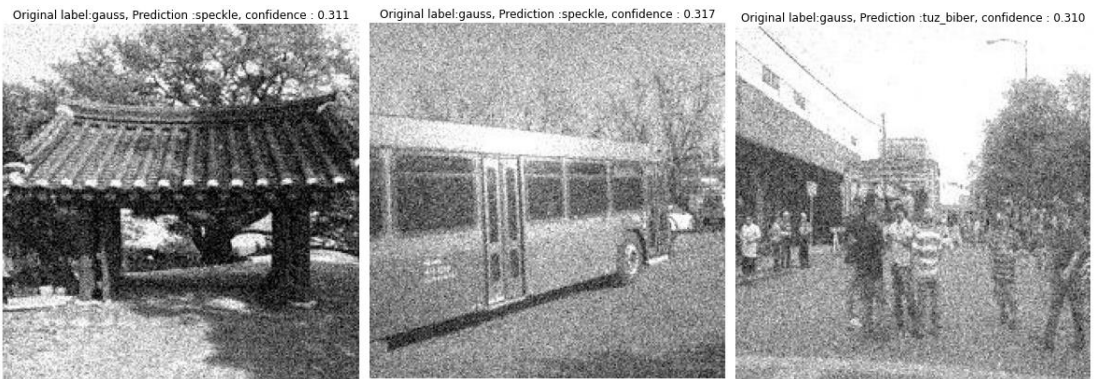


Original label:speckle, Prediction :gauss, confidence : 0.808

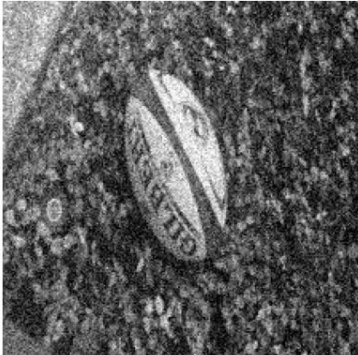




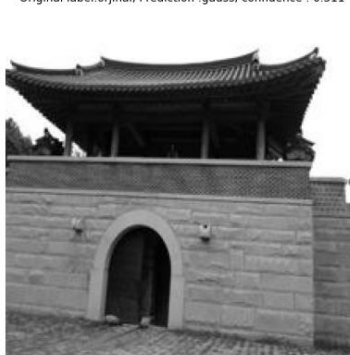
SGD optimizasyon algoritması ile oluşturulan modelin test sonucunda hatalı bulunan görüntülere örnekler:



Original label:gauss, Prediction :tuz_biber, confidence : 0.312



Original label:original, Prediction :gauss, confidence : 0.311



Original label:original, Prediction :gauss, confidence : 0.319



Original label:original, Prediction :speckle, confidence : 0.316



Original label:original, Prediction :tuz_biber, confidence : 0.315



Original label:original, Prediction :tuz_biber, confidence : 0.304



Original label:speckle, Prediction :tuz_biber, confidence : 0.318



Original label:speckle, Prediction :tuz_biber, confidence : 0.309



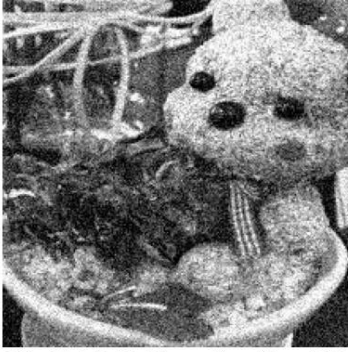
Original label:speckle, Prediction :gauss, confidence : 0.314



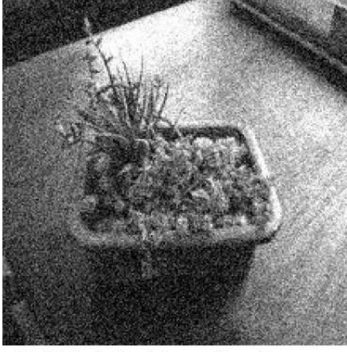
Ek 2. RMSProp Optimizasyon Algoritması için Katman Farklılıklarında Hatalı Bulunan Görüntüler

RMSProp optimizasyon algoritması ile oluşturulan bir(1) katmanlı modelin test sonucunda hatalı bulunan görüntüler:

Original label:gauss, Prediction :speckle, confidence : 1.000



Original label:gauss, Prediction :speckle, confidence : 1.000



Original label:orjinal, Prediction :speckle, confidence : 0.854



Original label:orjinal, Prediction :speckle, confidence : 1.000



Original label:orjinal, Prediction :speckle, confidence : 0.990



Original label:orjinal, Prediction :speckle, confidence : 0.985



Original label:orjinal, Prediction :gauss, confidence : 1.000



Original label:orjinal, Prediction :speckle, confidence : 0.994



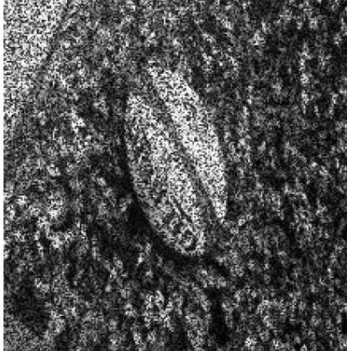
Original label:speckle, Prediction :gauss, confidence : 0.662



Original label:speckle, Prediction :gauss, confidence : 0.905



Original label:speckle, Prediction :tuz_biber, confidence : 0.759



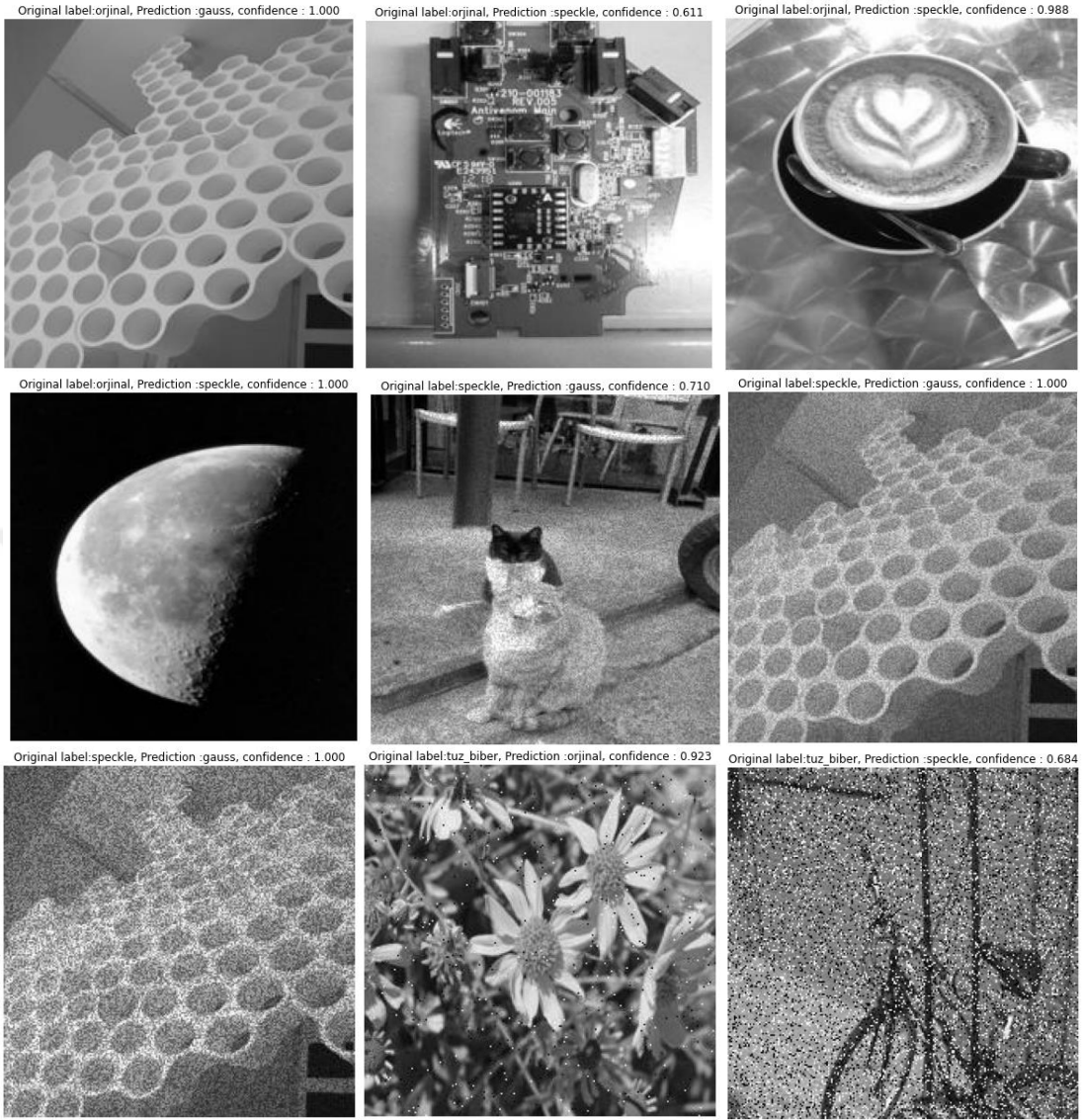
Original label:speckle, Prediction :gauss, confidence : 0.986





RMSProp optimizasyon algoritması ile oluşturulan iki(2) katmanlı modelin test sonucunda hatalı bulunan görüntüler:





RMSProp optimizasyon algoritması ile oluşturulan üç(3) katmanlı modelin test sonucunda hatalı bulunan görüntüler:



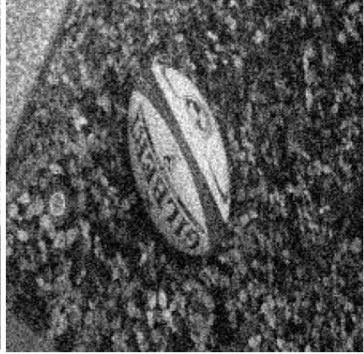
Original label:gauss, Prediction :speckle, confidence : 1.000



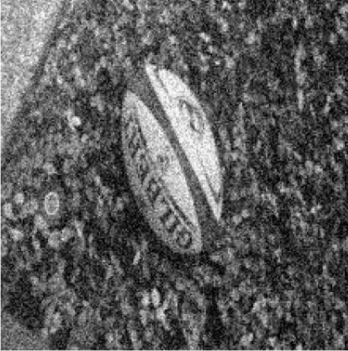
Original label:gauss, Prediction :speckle, confidence : 0.542



Original label:gauss, Prediction :speckle, confidence : 0.938



Original label:gauss, Prediction :speckle, confidence : 0.952



Original label:gauss, Prediction :tuz biber, confidence : 0.834



Original label:gauss, Prediction :speckle, confidence : 0.827



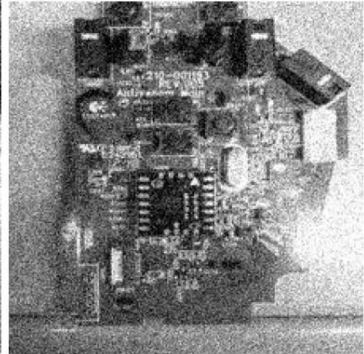
Original label:gauss, Prediction :tuz biber, confidence : 0.299



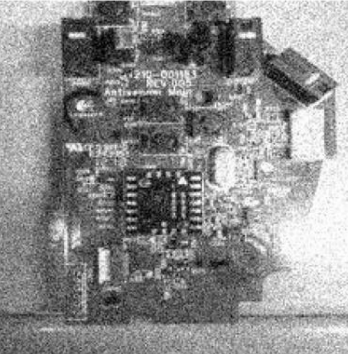
Original label:gauss, Prediction :speckle, confidence : 0.993



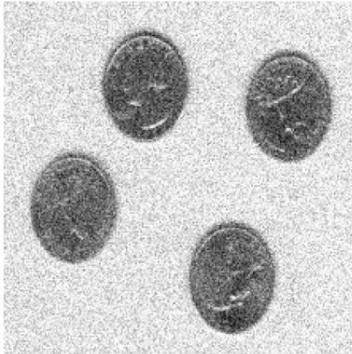
Original label:gauss, Prediction :speckle, confidence : 0.963



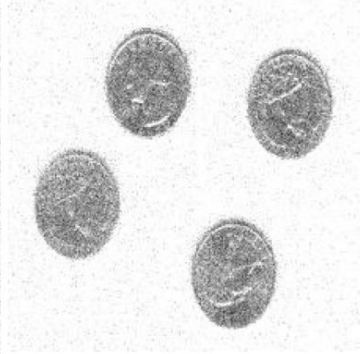
Original label:gauss, Prediction :speckle, confidence : 0.932



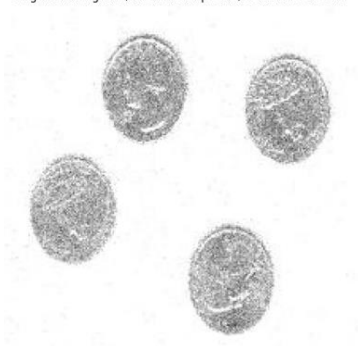
Original label:gauss, Prediction :speckle, confidence : 0.998



Original label:gauss, Prediction :speckle, confidence : 0.934



Original label:gauss, Prediction :speckle, confidence : 0.973



Original label:original, Prediction :speckle, confidence : 0.504



Original label:original, Prediction :speckle, confidence : 0.524



Original label:original, Prediction :speckle, confidence : 0.787



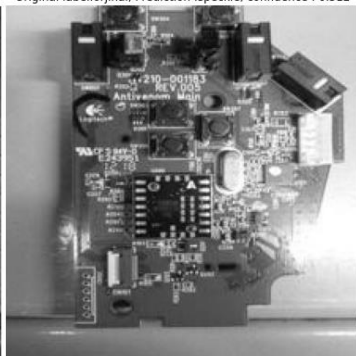
Original label:original, Prediction :speckle, confidence : 0.718



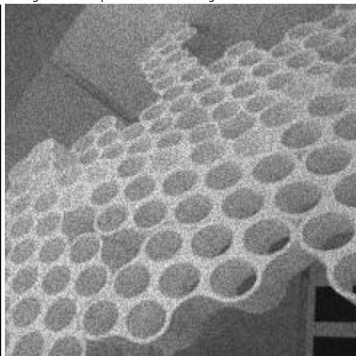
Original label:original, Prediction :speckle, confidence : 0.553



Original label:original, Prediction :speckle, confidence : 0.512



Original label:speckle, Prediction :gauss, confidence : 0.583



Original label:original, Prediction :speckle, confidence : 0.745



Original label:original, Prediction :speckle, confidence : 0.726



Original label:original, Prediction :speckle, confidence : 0.512



Original label:tuz_biber, Prediction :speckle, confidence : 0.688



Original label:tuz_biber, Prediction :speckle, confidence : 0.610 Original label:tuz_biber, Prediction :speckle, confidence : 0.464 Original label:tuz_biber, Prediction :speckle, confidence : 0.979



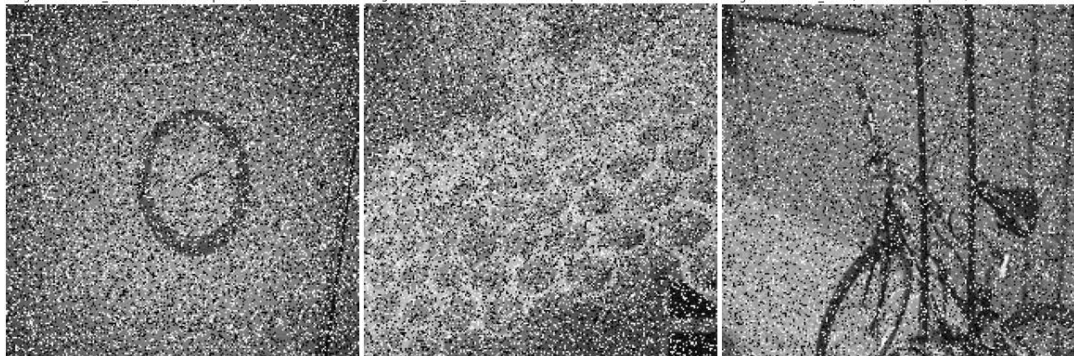
Original label:tuz_biber, Prediction :original, confidence : 0.559 Original label:tuz_biber, Prediction :original, confidence : 0.955 Original label:tuz_biber, Prediction :speckle, confidence : 0.468

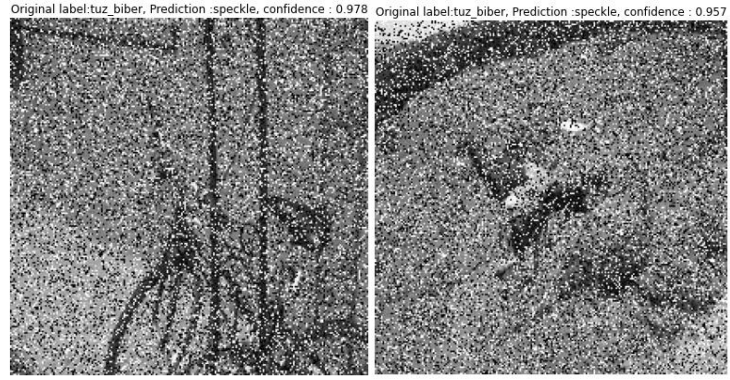


Original label:tuz_biber, Prediction :speckle, confidence : 0.795 Original label:tuz_biber, Prediction :speckle, confidence : 0.416 Original label:tuz_biber, Prediction :original, confidence : 0.532



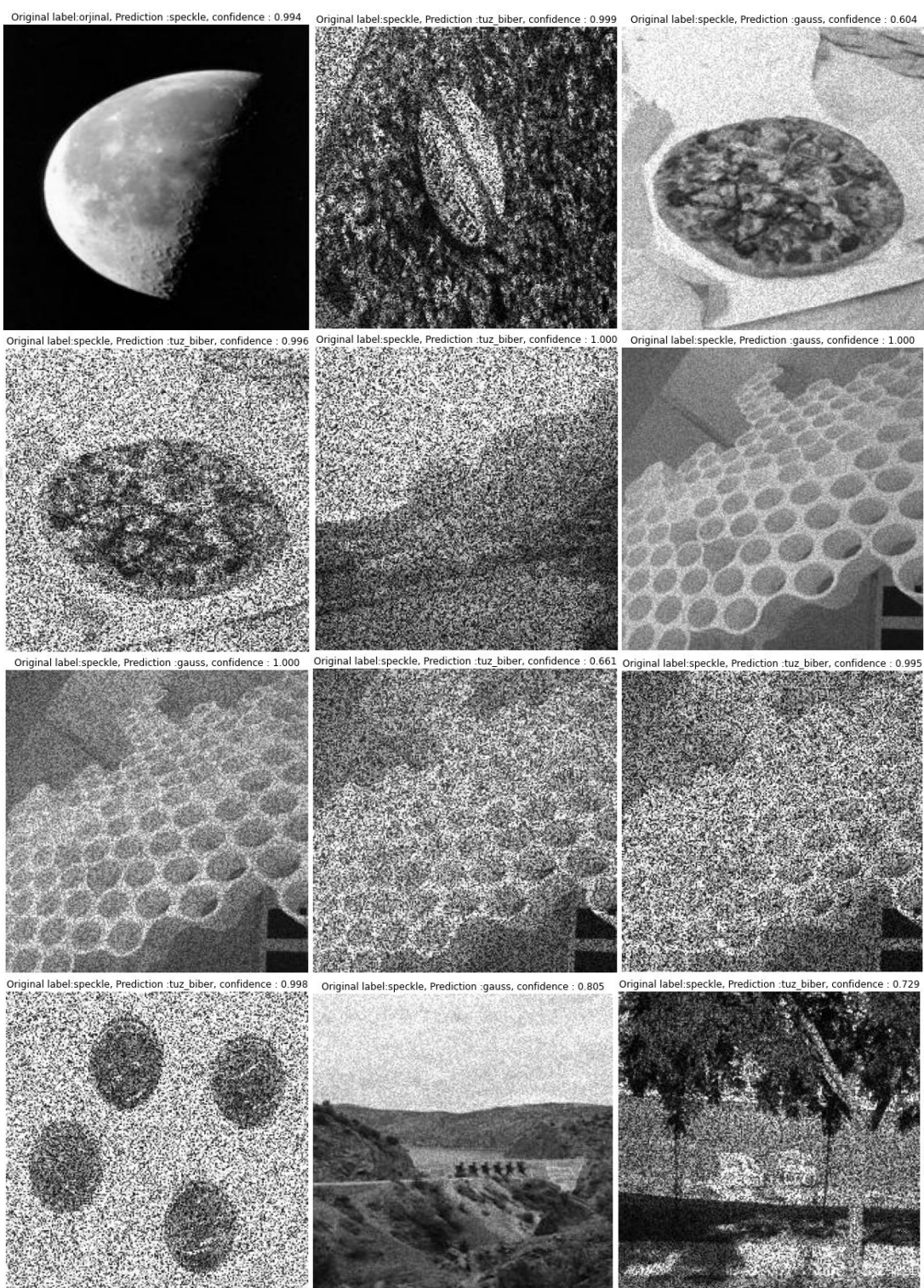
Original label:tuz_biber, Prediction :speckle, confidence : 0.838 Original label:tuz_biber, Prediction :speckle, confidence : 0.508 Original label:tuz_biber, Prediction :speckle, confidence : 0.996





RMSProp optimizasyon algoritması ile oluşturulan beş(5) katmanlı modelin test sonucunda hatalı bulunan görüntüler:





Original label: speckle, Prediction : tuz_biber, confidence : 1.000



Ek 3. Yazarın Özgeçmiři

Adı Soyadı

Kişisel Bilgiler

İletişim Bilgileri

Öğrenim Bilgileri

İş Deneyimi
