T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK BENZERLİKLİ NESNELERİN MAKİNE ÖĞRENMESİ TABANLI SINIFLANDIRILMASI

Canan TAŞTİMUR TEMİZ

Doktora Tezi

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Kuramsal Temeller Bilim Dalı

KASIM 2022

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

YÜKSEK BENZERLİKLİ NESNELERİN MAKİNE ÖĞRENMESİ TABANLI SINIFLANDIRILMASI

Tez Yazarı
Canan TAŞTİMUR TEMİZ

Danışman
Prof. Dr. Erhan AKIN

KASIM 2022
ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

Başlığı:	Yüksek Benzerlikli Nesnelerin Makine Öğrenmesi Tabanlı Sınıflandırılması
Yazarı:	Canan TAŞTİMUR TEMİZ
İlk Teslim Tarihi:	26.09.2022
Savunma Tarihi:	01.11.2022

TEZ ONAYI

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

		İmza
Danışman:	Prof. Dr. Erhan AKIN Fırat Üniversitesi, Mühendislik Fakültesi	Onayladım
Başkan:	Prof. Dr. Yakup DEMİR Fırat Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Prof. Dr. Mehmet KARAKÖSE Fırat Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Dr. Öğr. Üyesi İbrahim Rıza HALLAÇ Alanya Alaaddin Keykubat Üniversitesi, Rafet Kayış Mühendislik Fakültesi	Onayladım
Üye:	Dr. Öğr. Üyesi Burak YILMAZ KTO Karatay Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi	Onayladım

Bu tez, Enstitü Yönetim Kurulunun ...../...../20..... tarihli toplantısında tescillenmiştir.

İmza

Prof. Dr. Burhan ERGEN
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım "Yüksek Benzerlikli Nesnelerin Makine Öğrenmesi Tabanlı Sınıflandırılması" başlıklı Doktora Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

01.11.2022

Canan TAŞTİMUR TEMİZ

ÖNSÖZ

Makine Öğrenmesi, yapısal olarak öğrenebilen ve tahmin yapabilen algoritmalar bütünüdür. Yapay Zekâ' dan geliştirilmiş olup onlarca sene boyunca sıklıkla kullanılmıştır. Nesne sınıflandırma ve tanıma konularında oldukça başarılı sonuçlar üretmiştir. Makine öğrenmesi alanından türeyen Derin Öğrenme yaklaşımı son yıllarda nesne tespit, nesne tanıma, nesne sınıflandırma, arıza tespit ve teşhisi, ses işleme ve tanıma gibi birçok alanda yaygın olarak kullanılmaktadır. Derin Öğrenme; sağlık, endüstri, biyo-bilişim, e-ticaret, robotik, doğal dil işleme vb. alanlarda uygulanmaktadır. Nesne sınıflandırma görevini zorlaştıran faktörler; eğitim verilerinin yeterli sayıda(miktarda) olmaması, veri setlerindeki görüntülerin düşük kalitesi, veri setinin eğitim ve doğrulama aşamalarındaki görüntüler açısından dengesiz olması ve sınıflandırılacak nesnelerin benzer olmasıdır. Bahsedilen problemlerden, sınıflandırılacak nesnelerin benzer olması konusu bu tez çalışması kapsamında ele alınmıştır. Görsel olarak birbirlerine büyük ölçüde benzerlik gösteren endüstriyel tabanlı görüntüler, sınıflandırma için derin öğrenme tabanlı yaklaşımlar önerilmiştir.

Lisansüstü eğitimim ve bu tez çalışması boyunca değerli bilgisini, zamanını ve emeğini benden esirgemeyen ve beni akademik hayata hazırlayan danışman hocam sayın Prof. Dr. Erhan AKIN' a katkılarından ötürü teşekkür ederim. Tez çalışmasında kullandığımız veri setimizin oluşturulmasında destek veren NİT Örme Tekstil San. ve Tic. Ltd. Şti' ye teşekkür ederim. Tez çalışması süresince, değerli fikirleriyle beni akademik hayata hazırlayan hocam Prof. Dr. Mehmet KARAKÖSE' ye teşekkür ederim.

Sadece bu süreçte olmayıp hayatın her alanında yardımını ve desteğini hissettiğim sevgili eşim Mehmet Fatih TEMİZ' e ve üzerimde sayısız emeği olan kıymetli aileme teşekkür ederim.

Bu tezi, varlığıyla bana güç veren canım oğlum Mustafa Tuğra TEMİZ' e ithaf ediyorum.

Canan TAŞTİMUR TEMİZ

ELAZIĞ, 2022

İÇİNDEKİLER

Sayfa

ÖNSÖZ ..... İV
İÇİNDEKİLER V
ÖZET ..... Vİİ
ABSTRACT ..... Vİİİ
ŞEKİLLER LİSTESİ ..... İX
TABLOLAR LİSTESİ ..... XİV
SİMGELER VE KISALTMALAR ..... XVİ

1. GİRİŞ 1
1.1. Literatür Taraması 1
1.2. Tezin Amacı ve Kapsamı 10
1.3. Tezin Yapısı 11

2. MATERYAL VE METOT 12
2.1. Veri Setlerinin Hazırlanması 12
2.1.1. Veri Setleri için Uygulanan Ön-İşleme Adımları 14
Görüntülerin Ön-işlenmesi ve Boyutlandırma 14
Görüntülere Histogram Eşitleme Uygulanması 14
2.1.2. Veri Setleri için Uygulanan Veri Arttırma Tekniği 15
2.2. Pantograf Yüzeyindeki Arızaların Derin Öğrenme Tabanlı Sınıflandırılması 15
2.2.1. Konvolüsyon Katmanı 18
2.2.2. Havuzlama (Pooling) Katmanı 21
2.2.3. Tam Bağlantı (Fully Connected) Katmanı 22
2.3. Bağlantı Elemanlarının Transfer Öğrenme Tabanlı İnce-Ayar ve Özellik Çıkarım Teknikleri ile Sınıflandırılması 22
2.3.1. Transfer Öğrenme Tabanlı İnce-Ayar ve Öznitelik Çıkarımı 23
İnce-Ayar ve Öznitelik Çıkarımı Hakkında Temel Bilgi 23
İnce-Ayar Uygulanma Yöntemleri 24
2.3.2. Önceden-Eğitilmiş Modeller 24
VGGNet Modeli 25
Inception Net Modeli 25
MobileNet Modeli 26
DenseNet Modeli 26
EfficientNet Modeli 27
ResNet Modeli 27
Xception Modeli 28
2.3.3. Önerilen Metot 28
2.4. Tekstil Makinesi Yedek Parçalarının Diferansiyel Gizlilik ve KSA ile Sınıflandırılması 29
2.4.1. Optimizasyon Algoritması 30
Stokastik Gradyan İnişi (SGD) Algoritması 31
Uyarlanabilir Gradyan (AdaGrad) Algoritması 32
Kök Ortalama Kare Yayılım Algoritması (RMSProp) 33
Uyarlanabilir Momentum Algoritması (Adam) 34
2.4.2. Diferansiyel Gizlilik Metodu 34
2.4.3. Önerilen Yöntem 37
2.5. Bağlantı Elemanlarının Siyam KSA ve Tek-Seferde Öğrenme ile Sınıflandırılması 40

2.5.1. Tek-Seferde Öğrenme 41
2.5.2. Siyam Konvolüsyonel Sinir Ağı 41
2.5.3. L1- Uzaklık Kaybı (L1-Distance Loss) 43
2.5.4. Kontrast Kaybı (Contrastive Loss) 43
2.5.5. Önerilen Yöntem 44
2.6. Bağlantı Elemanlarının ve Tekstil Makinesi Yedek Parçalarının Birkaç-Seferde Öğrenme Tabanlı İlişki Ağı ile Sınıflandırılması 46
2.6.1. Birkaç-Seferde Öğrenme 46
2.6.2. Derin Metrik Öğrenme 47
2.6.3. İlişki Ağı 49
2.6.4. Önerilen Yöntem 49
Öznitelik Modülü 51
Eğilim Modülü 51
İlişki Modülü 51
2.7. Bağlantı Elemanlarının Karşılaştırma Ağı ve Prototipik Ağı ile Birkaç-Seferde Öğrenme Tabanlı Sınıflandırılması 52
2.7.1. Önerilen Karşılaştırma Ağ Modeli 53
2.7.2. Önerilen Prototipik Ağ Modeli 54
3. BULGULAR VE TARTIŞMA 55
3.1. Performans Ölçüm Metrikleri 55
3.1.1. Karmaşıklık Matrisi 55
3.1.2. Doğruluk 55
3.1.3. Hassasiyet 56
3.1.4. Kesinlik 56
3.1.5. F1-Ölçüm 56
3.1.6. ROC Eğrisi ve Eğri Altında Kalan Alan(AUC) 56
3.2. Pantograf Yüzeyindeki Arızaların Derin Öğrenme Tabanlı Sınıflandırılmayla Elde Edilen Sonuçlar 57
3.3. Bağlantı Elemanlarının Transfer Öğrenme Tabanlı İnce-Ayar ve Özellik Çıkarım ile Sınıflandırma Sonuçları 61
3.4. Tekstil Makinesi Yedek Parçalarının Diferansiyel Gizlilik ve KSA ile Sınıflandırılması Sonuçları .. 72
3.5. Bağlantı Elemanlarının Siyam KSA ve TSÖ ile Sınıflandırılması Sonuçları 83
3.6. Bağlantı Elemanlarının ve Tekstil Makinesi Yedek Parçalarının BSÖ Tabanlı İlişki Ağı ile Sınıflandırılması Sonuçları 90
3.7. Bağlantı Elemanlarının Karşılaştırma Ağı ve Prototipik Ağı ile BSÖ Tabanlı Sınıflandırılma Sonuçları 99
3.8. Makine Öğrenmesi Sınıflandırma Yaklaşımları ile Tezdeki Yaklaşımların Karşılaştırılması 103
3.8.1. K-En Yakın Komşu Algoritması 103
3.8.2. Destek Vektör Makinesi 104
3.8.3. Naive Bayes Algoritması 104
3.8.4. Karar Ağaçları 104
3.8.5. Extra Ağaçlar 105
3.8.6. Rastgele Orman Algoritması 106
3.8.7. Gradyan Arttırma Algoritması 106
3.8.8. Ridge Regresyon Algoritması 106
3.8.9. Makine Öğrenmesi Algoritmaları ile Bizim Çalışmalarımızın Kıyaslanması 106
4. SONUÇLAR 109
KAYNAKLAR 111
ÖZGEÇMİŞ

ÖZET

Yüksek Benzerlikli Nesnelerin Makine Öğrenmesi Tabanlı Sınıflandırılması

Canan TAŞTİMUR TEMİZ

Doktora Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Kasım 2022, Sayfa: xviii +119

Bilgisayarlı görme teknolojileri ile nesnelerin sınıflandırılması, birçok alanda kolaylık sağlamaktadır. Derin öğrenme yaklaşımı, son yıllarda nesne sınıflandırma konusunda oldukça popülerdir. Sınıflandırılmak istenen nesnelerin görsel açıdan birbirine benzer olması, veri setindeki görüntü sayısının yetersiz miktarda olması, aynı sınıf içindeki nesnelerin farklı türevlerinin bulunmasıyla beraber farklı sınıflardaki nesneler arasındaki benzerliğin yüksek olması gibi faktörler sınıflandırma işini zorlaştırmaktadır. Bu tez çalışmasında, birbirine görsel açıdan yüksek benzerlik barındıran nesne gruplarının derin öğrenme tabanlı yaklaşımlarla sınıflandırılması amaçlanmıştır. Tez çalışması kapsamında, farklı türde ve sayıda verilerin sınıflandırıldığı derin öğrenme tabanlı altı adet farklı yaklaşım önerilmiştir. Tezde kullanılmak üzere oluşturulan veri setlerimizin içeriği; endüstriyel vida, cıvata ve somun görüntüleri, pantograf görüntüleri ve tekstil makinesi yedek parça görüntülerinden oluşmaktadır.

Tezde kullanılmak üzere tekstil makinesi yedek parça görüntülerinin elde edilmesinde, NİT Örme Tekstil San. ve Tic. Ltd. Şti' den yardım alınmıştır. Ayrıca tez kapsamındaki çalışmalarda kullanmak için birbirine benzer nitelikte olan nesne gruplarının bulunduğu veri setleri de hazırlanmıştır. Önerilen yöntemlerin literatür ile detaylı olarak karşılaştırılması yapılmıştır ve elde edilen sonuçlar incelendiğinde, önerdiğimiz yöntemlerin oldukça başarılı ve etkin olduğu saptanmıştır.

Anahtar Kelimeler: Derin öğrenme, Makine öğrenmesi, Nesne sınıflandırma.

ABSTRACT

Classification of High-Similarity Objects Based Machine Learning

Canan TAŞTİMUR TEMİZ

Ph.D. Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Computer Engineering

November 2022, Pages: xviii+ 119

The classification of objects using computer vision technologies is useful in a variety of situations. In recent years, the deep learning approach to object classification has grown in popularity. The visual similarity of the objects to be classified, the insufficient number of images in the data set, the presence of different derivatives of the objects in the same class, and the high similarity between the objects in different classes all make classification difficult. The goal of this thesis is to use deep learning-based approaches to classify object groups that have high visual similarity to each other. Many deep learning-based approaches for categorizing various types and numbers of data have been proposed within the scope of the thesis. The content of our data sets created for use in the thesis; industrial screw, bolt and nut images, pantograph images and textile machine spare parts images.

Assistance has been received from NİT Örme Textile Ind.Trade.Co.Ltd. in obtaining images of textile machine spare parts to be used in the thesis. Furthermore, data sets containing similar object groups have been prepared for use in studies within the scope of the thesis. The proposed methods have been thoroughly compared to the literature, and the results obtained revealed that the methods we proposed are quite successful and effective.

Keywords: Deep Learning, Machine learning, Object classification.

ŞEKİLLER LİSTESİ

Sayfa

Şekil 1.1. [10]' da önerilen derin öğrenme mimarisinin blok diyagramı....4

Şekil 1.2. [50]' de önerilen yaklaşımın blok diyagramı....8

Şekil 2.1. *VS_Pant* verisetimizin her sınıfına ait örnek görüntüler....12

Şekil 2.2. *VS_Baglanti* verisetimizin her sınıfına ait örnek görüntüler....12

Şekil 2.3. *VS_Nit1* verisetimizin her sınıfına ait örnek görüntüler....13

Şekil 2.4. *VS_Nit2* verisetimizin her sınıfına ait örnek görüntüler....13

Şekil 2.5. Histogram eşitleme sürecindeki temel fikir....15

Şekil 2.6. Genel bir pantograf-katener sistemi ve önemli temas noktaları....16

Şekil 2.7. Önerilen çalışmanın blok diyagramı....17

Şekil 2.8. Önerilen çalışmanın KSA modeli....17

Şekil 2.9. Örnek filtre ve örnek piksel matrisin konvolüsyon işlemi....20

Şekil 2.10. Adım değerinin konvolüsyonda işleyişi....20

Şekil 2.11. Maksimum ve ortalama havuzlama örneği....21

Şekil 2.12. Temsili transfer öğrenme gösterimi....23

Şekil 2.13. İnce-ayar uygulama senaryoları....24

Şekil 2.14. VGG16 ve VGG19 mimarileri....25

Şekil 2.15. InceptionV3 modeli....26

Şekil 2.16. DenseNet169 ve DenseNet201 model mimarileri....27

Şekil 2.17. ResNet mimarisi....27

Şekil 2.18. Xception mimarisi....28

Şekil 2.19. Önerilen çalışmanın genel çatısı....29

Şekil 2.20. Bir sorgulayıcı karşısında diferansiyel gizlilik....35

Şekil 2.21. Yerel diferansiyel gizlilik ve global diferansiyel gizlilik....36

Şekil 2.22. Önerilen yöntemin genel gösterimi....38

Şekil 2.23. Siyam ağının örnek gösterimi....42

Şekil 2.24. Kontrast kaybı fonksiyonunun işleyişi....44

Şekil 2.25. Önerilen çalışmanın blok şeması....45

Şekil 2.26. Derin metrik öğrenme [128]....48

Şekil 2.27. DMÖ ve KSA tabanlı bir eğitim yaklaşımı....48

Şekil 2.28. Literatürdeki standart İlişki ağının genel yapısı....49

Şekil 2.29. Önerilen İlişki ağının genel yapısı....50

Şekil 2.30. GMRNet' teki öznitelik modülü....51

Şekil 2.31. GMRNet' teki ilşiki modülü....52
Şekil 2.32. Önerilen Karşılaştırma ağı....53
Şekil 2.33. Önerilen Prototipik ağı....54
Şekil 3.1. Karmaşıklık matrisinin genel bilgisi....55
Şekil 3.2. Durum1' de eğitim ve doğrulama sonucunda elde edilen doğruluk ve hata oranları....59
Şekil 3.3. Durum2' de eğitim ve doğrulama sonucunda elde edilen doğruluk ve hata oranları....59
Şekil 3.4. Durum3' te eğitim ve doğrulama sonucunda elde edilen doğruluk ve hata oranları....59
Şekil 3.5. Durum4' 4e eğitim ve doğrulama sonucunda elde edilen doğruluk ve hata oranları....60
Şekil 3.6. VGG16 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....62
Şekil 3.7. VGG19 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....62
Şekil 3.8. InceptionV3 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....63
Şekil 3.9. InceptionResNetV2 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....63
Şekil 3.10. Xception mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....63
Şekil 3.11. DenseNet169 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....63
Şekil 3.12. DenseNet201 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....63
Şekil 3.13. MobileNet mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....63
Şekil 3.14. MobileNetV2 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....64
Şekil 3.15. ResNet50 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....64
Şekil 3.16. EfficientNetB0 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....64
Şekil 3.17. EfficientNetB1 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....64
Şekil 3.18. EfficientNetB2 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....64
Şekil 3.19. EfficientNetB3 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....64
Şekil 3.20. EfficientNetB4 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....65
Şekil 3.21. EfficientNetB5 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....65
Şekil 3.22. EfficientNetB6 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....65
Şekil 3.23. EfficientNetB7 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı....65
Şekil 3.24. EfficientNet mimarisinin üç senaryoya göre doğruluk oranı....66
Şekil 3.25. TL-FT1 senaryosuna göre tüm modellerin doğruluk oranı....66
Şekil 3.26. TL-FT2 senaryosuna göre tüm modellerin doğruluk oranı....66
Şekil 3.27. TL-FE senaryosuna göre tüm modellerin doğruluk oranı....67
Şekil 3.28. DG-GradientDescent' in eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı....73
Şekil 3.29. DG-Adagrad' ın eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı....74
Şekil 3.30. DG-RMSProp' ın eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı....74
Şekil 3.31. DG-Adam' ın eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı....74

Şekil 3.32. DG-GradientDescentGaussian' ın eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı 75

Şekil 3.33. DG-AdagradGaussian' ın eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı 75

Şekil 3.34. DG-RMSPropGaussian' ın eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı 75

Şekil 3.35. DG-AdamGaussian' ın eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı 76

Şekil 3.36. Önerilen modelin farklı optimizasyon algoritmalarına göre karmaşıklık matrisi 77

Şekil 3.37. γ=1.0, ϑ=1.1 ve α=0.25 koşulları altında gizlilik maliyeti ve doğruluk oranı 80

Şekil 3.38. γ=1.5, ϑ=1.1 ve α=0.15 koşulları altında gizlilik maliyeti ve doğruluk oranı 80

Şekil 3.39. γ=1.0, ϑ=1.1 ve α=0.15 koşulları altında gizlilik maliyeti ve doğruluk oranı 81

Şekil 3.40. γ=0.5, ϑ=1.1 ve α=0.15 koşulları altında gizlilik maliyeti ve doğruluk oranı 81

Şekil 3.41. γ=1.0, ϑ=1.5 ve α=0.15 koşulları altında gizlilik maliyeti ve doğruluk oranı 81

Şekil 3.42. γ=1.0, ϑ=1.8 ve α=0.15 koşulları altında gizlilik maliyeti ve doğruluk oranı 82

Şekil 3.43. γ=1.0, ϑ=2.0 ve α=0.15 koşulları altında gizlilik maliyeti ve doğruluk oranı 82

Şekil 3.44. Sigmoid aktivasyon fonksiyonu, L1-distance ve 3-way TSÖ ile olası sınıf sayısına göre doğruluk oranı 84

Şekil 3.45. Softplus aktivasyon fonksiyonu, L1-distance ve 3-way TSÖ ile olası sınıf sayısına göre doğruluk oranı 84

Şekil 3.46. Tanh aktivasyon fonksiyonu, L1-distance ve 3-way TSÖ ile olası sınıf sayısına göre doğruluk oranı 84

Şekil 3.47. Sigmoid aktivasyon fonksiyonu, L1-distance ve 5-way TSÖ ile olası sınıf sayısına göre doğruluk oranı 84

Şekil 3.48. Softplus aktivasyon fonksiyonu, L1-distance ve 5-way TSÖ ile olası sınıf sayısına göre doğruluk oranı 84

Şekil 3.49. Softplus aktivasyon fonksiyonu, Constrastive fonksiyon ve 5-way TSÖ ile olası sınıf sayısına göre doğruluk oranı 84

Şekil 3.50. Tanh aktivasyon fonksiyonu, L1-distance ve 5-way TSÖ ile olası sınıf sayısına göre doğruluk oranı 85

Şekil 3.51. Sigmoid aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile olası sınıf sayısına göre doğruluk oranı 85

Şekil 3.52. Softplus aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile olası sınıf sayısına göre doğruluk oranı 85

Şekil 3.53. Tanh aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile olası sınıf sayısına göre doğruluk oranı 85

Şekil 3.54. Sigmoid aktivasyon fonksiyonu, L1-distance ve 3-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı 85

Şekil 3.55. Softplus aktivasyon fonksiyonu, L1-distance ve 3-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı 85

Şekil 3.56. Tanh aktivasyon fonksiyonu, L1-distance ve 3-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı 86

Şekil 3.57. Sigmoid aktivasyon fonksiyonu, L1-distance ve 5-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı 86

Şekil 3.58. Softplus aktivasyon fonksiyonu, L1-distance ve 5-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı 86

Şekil 3.59. Softplus aktivasyon fonksiyonu, Contrastive fonkisyon ve 5-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı 86

Şekil 3.60. Tanh aktivasyon fonksiyonu, L1-distance ve 5-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı 86

Şekil 3.61. Sigmoid aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı 86

Şekil 3.62. Softplus aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı 87

Şekil 3.63. Tanh aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı 87

Şekil 3.64. Sigmoid aktivasyon fonksiyonu, L1-distance ve 3.way TSÖ ile iterasyon sayısına göre doğruluk oranı 87

Şekil 3.65. Softplus aktivasyon fonksiyonu, L1-distance ve 3-way TSÖ ile iterasyon sayısına göre doğruluk oranı 87

Şekil 3.66. Tanh aktivasyon fonksiyonu, L1-distance ve 3.way TSÖ ile iterasyon sayısına göre doğruluk oranı 87

Şekil 3.67. Sigmoid aktivasyon fonksiyonu, L1-distance ve 5-way TSÖ ile iterasyon sayısına göre doğruluk oranı 87

Şekil 3.68. Softplus aktivasyon fonksiyonu, L1-distance ve 5.way TSÖ ile iterasyon sayısına göre doğruluk oranı 88

Şekil 3.69. Softplus aktivasyon fonksiyonu, Contrastive fonksiyon ve 5-way TSÖ ile iterasyon sayısına göre doğruluk oranı 88

Şekil 3.70. Tanh aktivasyon fonksiyonu, L1-distance ve 5.way TSÖ ile iterasyon sayısına göre doğruluk oranı 88

Şekil 3.71. Sigmoid aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile iterasyon sayısına göre doğruluk oranı 88

Şekil 3.72. Softplus aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile iterasyon sayısına göre doğruluk oranı 88

Şekil 3.73. Tanh aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile iterasyon sayısına göre doğruluk oranı 88

Şekil 3.74. Önerilen GMRNet modelinin *VS_Baglanti* verisetinde N-way K-shot sınıflandırma sonuçları 91

Şekil 3.75. Önerilen GMRNet modelinin *VS_Nit2* verisetinde N-way K-shot sınıflandırma sonuçları 92

Şekil 3.76. GMRNet modelinin veri setlerindeki N-way K-shot sınıflandırma yüzde oranları....93

Şekil 3.77. *VS_Baglanti_Arttir* veri setinin literatürdeki standart RN modeline uygulanmasıyla elde edilen t-SNE görüntüleri.....94

Şekil 3.78. *VS_Baglanti_Arttir* veri setinin GMRNet modeline uygulanmasıyla elde edilen t-SNE görüntüleri. 95

Şekil 3.79. *VS_Baglanti_Arttir* veri setinin standart ve önerilen İlişki ağlarına uygulanmasıyla elde edilen doğruluk oranları 95

Şekil 3.80. *VS_Nit2_Arttir* veri setinin literatürdeki standart RN modeline uygulanmasıyla elde edilen t-SNE görüntüleri. 96

Şekil 3.81. *VS_Nit2_Arttir* veri setinin GMRNet modeline uygulanmasıyla elde edilen t-SNE görüntüleri. 97

Şekil 3.82. *VS_Nit2_Arttir* veri setinin standart ve önerilen İlişki ağlarına uygulanmasıyla elde edilen doğruluk oranları 98

Şekil 3.83. Önerilen Karşılaştırma modelinin N-way K-shot sınıflandırma sonuçları 101

Şekil 3.84. Önerilen Prototipik modelinin N-way K-shot sınıflandırma sonuçları 102

Şekil 3.85. Karşılaştırma ağın ve Prototipik ağın N-way K-shot sınıflandırma yüzde oranları 103

Şekil 3.86. K-NN algoritmasının çalışma prensibi 103

Şekil 3.87. DVM algoritmasının çalışma prensibi.....104

Şekil 3.88. KA algoritmasının düğüm yapısı örneği 105

Şekil 3.89. EA algoritmasının çalışma prensibi.....105

Şekil 3.90. RO algoritmasının çalışma prensibi 106

TABLOLAR LİSTESİ

Sayfa

Tablo 2.1. SGD optimizasyon algoritmasının sözde kodu 32

Tablo 2.2. Adagrad optimizasyon algoritmasının sözde kodu....33

Tablo 2.3. RMSProp optimizasyon algoritmasının sözde kodu 33

Tablo 2.4. Adam optimizasyon algoritmasının sözde kodu....34

Tablo 2.5. DG-SGD optimizasyon algoritmasının sözde kodu 38

Tablo 2.6. DG-Adagrad optimizasyon algoritmasının sözde kodu....38

Tablo 2.7. DG-RMSProp optimizasyon algoritmasının sözde kodu 39

Tablo 2.8. DG-Adam optimizasyon algoritmasının sözde kodu....39

Tablo 2.9. DG algoritmalarındaki hiper-parametre ayarlamaları 40

Tablo 2.10. GMRNet algoritmasının sözde kodu....50

Tablo 3.1. Geliştirilen KSA mimarisinin parametreleri....58

Tablo 3.2. Bu çalışmadaki Durum1' in performans raporu 60

Tablo 3.3. Bu çalışmadaki Durum2' nin performans raporu 60

Tablo 3.4. Bu çalışmadaki Durum3' ün performans raporu 60

Tablo 3.5. Bu çalışmadaki Durum4' ün performans raporu 61

Tablo 3.6. Bu çalışmadaki tüm durumların performans raporu....61

Tablo 3.7. Kullanılan KSA modelindeki hiper-parametreler....62

Tablo 3.8. Tüm modellerin üç senaryoya göre doğruluk oranı karşılaştırması....67

Tablo 3.9. TL-FT1 senaryosuna göre performans metrikleri 68

Tablo 3.10. TL-FT2 senaryosuna göre performans metrikleri 68

Tablo 3.11. TL-FE senaryosuna göre performans metrikleri 69

Tablo 3.12. Literatürdeki Transfer Öğrenme metotlarıyla önerilen yöntem karşılaştırılması 70

Tablo 3.13. Geliştirilen KSA mimarisinin parametreleri....73

Tablo 3.14. DG-GradientDescent' in sınıflandırma raporu....77

Tablo 3.15. DG-Adagrad' ın sınıflandırma raporu....78

Tablo 3.16. DG-RMSProp' ın sınıflandırma raporu....78

Tablo 3.17. DG-Adam' ın sınıflandırma raporu....78

Tablo 3.18. DG- GradientDescentGaussian' ın sınıflandırma raporu 78

Tablo 3.19. DG- AdagradGaussian' ın sınıflandırma raporu 79

Tablo 3.20. DG- RMSPropGaussian' ın sınıflandırma raporu 79

Tablo 3.21. DG- AdamGaussian' ın sınıflandırma raporu 79

Tablo 3.22. Bu çalışmada kullanılan optimizasyon algoritmalarının performans metrikleri....79

Tablo 3.23. DG algoritmasındaki örnek senaryolar üzerinden gizlilik maliyeti ve doğruluk oranı ilişkisi 82

Tablo 3.24. Bu çalışmada kullanılan hiper-parametre değerleri.....83

Tablo 3.25. İlişki ağı modelindeki hiper-parametre değerleri 90

Tablo 3.26. Önerilen GMRNet modelinin N-way K-shot sınıflandırma sonuçları.....93

Tablo 3.27. Literatürdeki standart İlişki ağı modeli ile önerilen ilişki ağ modelinin performans karşılaştırılması 98

Tablo 3.28. Literatürdeki derin metrik öğrenme tabanlı çalışmalarla bizim çalışmanın karşılaştırılması 98

Tablo 3.29. Karşılaştırma ve Prototipik ağında ortak kullanılan hiper-parametre değerleri.....100

Tablo 3.30. Önerilen Karşılaştırma ağı ve Prototipik ağın N-way K-shot sınıflandırma sonuçları.....102

Tablo 3.31. Literatürdeki sınıflandırma metotları ile önerdiğimiz metotların performans karşılaştırılması.....107

SİMGELER VE KISALTMALAR

Simgeler

S_k : Görüntünün gri düzeyi dönüştürme sonucu
n_j : j. gri tonlamadaki piksel sayısı
n : Görüntüdeki piksel sayımlarının toplamı
L : İstenen toplam gri düzeyi sayısı
FM_{size} : Özellik haritasının boyutu
i_{size} : Konvolüsyon katmanının boyutu
k_s : Filtre boyutu
N' : Seri konvolüsyon filtresi için ağırlık sayısı
W_1 : Giriş görüntüsünün genişliği
H_1 : Giriş görüntüsünün yüksekliği
D_1 : Giriş görüntüsünün derinliği
f : Filtre boyutu
s : Adım değeri
SGI : Havuzlama sonucunda elde edilen görüntü boyutu
α : SGD algoritmasının öğrenme oranı
θ : SGD algoritmasının ağırlık değeri
L : SGD algoritmasının kayıp değeri
x : Ağa verilen girdi
y : Ağdan beklenen etiket çıktısı
δ : Modelin eğitimini durdurma kriteri
G_t : Adagrad algoritmasının eğimlerin dış çarpımının toplamı
g_j : Adagrad algoritmasının zaman içindeki eğim tahmini
I : Birim matris
v_t : RMSProp algoritmasının gradyanların dış çarpımının toplamı
g_t : RMSProp algoritmasındaki gradyan değeri
p : Azalan sabit
β_1 : Adam algoritmasının birinci zamandaki üstel olarak azalan oranı
m_t : Adam algoritmasının gradyanların üstel ortalaması
β_2 : Adam algoritmasının ikinci zamandaki üstel olarak azalan oranı
v_t : Adam algoritmasının karelerin üstel ortalaması
g_t : Adam algoritmasının t zamanındaki gradyan
D_1 : Örnek komşu veri seti
D_2 : Örnek komşu veri seti
D^* : Strelize veri seti
ε : Diferansiyel gizlilik maliyeti
γ : Diferansiyel gizlilikte gradyan kırpma oranı
ϑ : Diferansiyel gizlilikte gürültü çarpanı
α : Diferansiyel gizlilikte öğrenme oranı
$f(\theta)$: L1-uzaklık kaybının öznitelik fonksiyonu
$f(\theta)$: L1-uzaklık kaybının öznitelik fonksiyonu
$\mathcal{L}(B)$: Hesaplanan mesafe değerinin olasılığı
E : Eğitim setindeki görüntülerin temsili
x : Veri setindeki örnek görüntü temsili
$\widehat{c_E}(x)$: x örneğinin ait olduğu sınıf tahmini

D_w : İki çıktı vektörü arasındaki mesafe
T_a^b : a-sınıfındaki rastgele b. örnek
q_m : Örnek sorgu girdisi
l_m : q_m örneğinin etiketi
m : Bağımsız değişkenlerin ortalama değer vektörü
C^{-1} : Bağımsız değişkenlerin kovaryans matrisinin tersi
T : Transpoze işlemi
E : İlişki ağındaki eğitim iterasyonu
b : İlişki ağındaki parti boyutu
lr : İlişki ağındaki öğrenme oranı
M : İlişki ağındaki geometrik ortalama
$f_\varphi(x)$: İlişki ağının özellik haritası
w_g : İlişki ağının ağ parametresi
w_f : İlişki ağının ağ parametresi
r_w : İlişki ağının ilişki modülü
$\acute{w_g}$: İlişki ağının öznitelik modülündeki optimize edilmiş eğitim parametresi
$\acute{w_d}$: İlişki ağının eğilim modülündeki optimize edilmiş eğitim parametresi
$\acute{r_w}$: İlişki ağının ilişki modülündeki optimize edilmiş eğitim parametresi
$\acute{w_g}$: İlişki ağının öznitelik modülündeki optimize edilmiş eğitim parametresi
Q : İlişki ağının sorgu kümesi
$g_\varphi(x)$: İlişki ağının ilişki modül fonksiyonu
$m_\varphi(x)$: İlişki ağının geometrik ortalama fonksiyonu
$s_{g,m}$: İlişki ağının benzerlik puanı
$C(,)$: İlişki ağının birleştirme fonksiyonu
$\acute{x}$: Karşılaştırma ağının örnek girdisi
$\acute{y}$: Karşılaştırma ağının çıktı etiketi
$\propto$: Karşılaştırma ağının kosinüs uzaklığı
v_c : Prototipik ağının örneklerinin ortalama vektörü

Kısaltmalar

AHUG : Alzheimer hastalığı ulusal girişimi
AUC : Eğri altında kalan alan
BSÖ : Birkaç-seferde öğrenme
CCE : Kategorik çapraz entropi
DG : Diferansiyel gizlilik
DMÖ : Derin metrik öğrenme
DVM : Destek vektör makinesi
EA : Ekstra ağaçlar algoritması
FE : Özellik çıkarım
FT : İnce ayar
GA : Gradyan arttırma algoritması
KA : Karar ağaçları algoritması
KNN : K-en yakın komşu algoritması
KSA : Konvolüsyonel Sinir Ağı
NB : Naive Bayes algoritması

RR : Ridge regresyon algoritması
RO : Rastgele orman algoritması
ROI : İlgilenilen bölge
TSÖ : Tek-seferde öğrenme
TL : Transfer öğrenme
VS_Baglanti : Bağlantı elemanı içeren veri seti
VS_Nit1 : Yedek parça görüntülerinden oluşan veri seti
VS_Nit2 : Yedek parça görüntülerinden oluşan veri seti
VS_Pant : Pantograf görüntülerinden oluşan veri seti
$V_{meta-egitim}$: Birkaç-seferde öğrenme tabanlı sınıflandırmada eğitim veri seti
$V_{meta-test}$: Birkaç-seferde öğrenme tabanlı sınıflandırmada test veri seti

1. GIRIŞ

Makine öğrenmesi, makinenin açık şekilde programlanmadan ve makineye sürekli bir şekilde kontrol olmadan makinenin kendi kendine öğrenmesidir. Makine, sonuçlara bakarak çıkarım yapmasının yanında otomatik olarak öğrenme yapar. Makine öğrenmesi, Yapay Zekâ' nın bir alt alanı olmasıyla beraber makine öğrenmesi daha belirgin bir çalışma alanına sahiptir. Yapay zekâ uygulamalarından farklı olarak kural bilgisi oluşturmaya ihtiyaç duymamaktadır. Çünkü makine öğrenmesi, kendisine girdi olarak verilen veri kümesini çözümleyerek gerçekleştirilmek istenen amacı öğrenmektedir. Makine öğrenmesi, veri miktarının artmasına paralel olarak doğru sonuç üretme ihtimali de artmaktadır.

Makine öğrenmesinin en önemli becerisi, veri kümesini model haline dönüştürebilmesidir. Makine öğrenmesinin bir alt dalı Derin Öğrenme'dir. Makine öğrenmesi tek katmanda işlem gerçekleştirirken derin öğrenme birçok katmanda işlem gerçekleştirmektedir [1]. Ayrıca derin öğrenmenin makine öğrenmesinden farkı, kodlanmış kuralları kullanarak öğrenme yapmamasıdır, bunun yerine veriler üzerinden otomatik olarak öğrenme yapmaktadır [2]. Derin öğrenmenin bir başka avantajı ise makine öğrenmesinden farklı olarak özellik çıkarma işlemine gerek duymamasıdır. Derin öğrenmenin bu özellikleri sayesinde uygulamalarda kullanımı gün geçtikçe artmaktadır. Derin öğrenmenin uygulama alanlarından birisi de nesne sınıflandırmadır. Literatürdeki nesne tespit ve sınıflandırma uygulamaları incelendiğinde, birçok farklı alanda derin öğrenme yaklaşımı ile nesne sınıflandırmanın uygulanarak başarılı sonuçlara ulaşıldığı görülmektedir.

1.1. Literatür Taraması

Togootogtokh ve Amartuvshin [3] çalışmalarında, birbirine görsel olarak çok benzeyen Chihuahua köpek cinsi ile Muffin pasta çeşidinin birbirinden ayırt edilmesi üzerinde durmuşlardır. Bu problemi, normal yapılandırılmış derin makine öğrenmesi hâlâ çözememektedir. Önerilen yöntem sadece 'Chihuahua ve Muffin' problemi için değil, diğer problemler için de uyarlanabilmektedir. Bu problemin çözümü için Derin Transfer Öğrenme yöntemi önermişlerdir. Ancak önerdikleri bu yöntemin dezavantajları, hem büyük miktarda veri gerektirmeyen küçük veri setleri için uygun olmasıdır hem de sadece iki grubu sınıflandırma yeteneğinin bulunmasıdır. Önerilen bu yöntemin avantajları ise çok küçük nesneleri tanımak için ideal olmasıdır ve çok büyük işlem gücü gerektirmemesidir. [3]' teki kullanılan mimari, Derin Transfer Öğrenme' dir. VGG19 modeline benzeyen 20 katmanlı bir mimari önermişlerdir.

Deri hastalıklarının dermoskopi görüntüleri için derin öğrenme kullanılarak yeni çoklu sınıflandırma yöntemi Zhou vd. [4] tarafından önerilmiştir. Deri hastalıklarının benzer görünümü nedeniyle, doku bozulması görüntüleri yoluyla otomatik sınıflandırma oldukça zor bir iştir. [4]'deki çalışmada ise dermoskopi görüntüleri için Konvolüsyonel Sinir Ağına (KSA) dayanan yeni çoklu

sınıflandırma yöntemi önerilmiştir. Önce, iç içe kalıcı yapıya sahip olan bir KSA ağı tasarlanmıştır ve bunlar sayesinde ilk kalıcı yapıdan daha fazla bilgi edinilebilir. Sonra tasarlanan ağ, Transfer öğrenimiyle eğitilmiştir. Eğitilen ağ ile 'Nevüs', 'Seboreik Keratoz', 'Sedef Hastalığı', 'Seboreik Dermatit', 'Egzema' ve 'Bazal Hücreli Korsinom' olmak üzere 6 çeşit deri hastalığı sınıflandırılmıştır. Deneysel çalışmalar iki yönde denenmiştir: Altı gruba sınıflandırma işlemi ve iki gruba sınıflandırma yönteminin doğruluk oranı ise %90' dır. KSA' lı yapının bir çeşidi olarak ResNet, daha derin ağların eğitimini kolaylaştırmak için kalıcı bir öğrenme çerçevesi sunmuştur. Önerilen mimaride, esas kalıcı yapıya atlamalı bağlantıları ekleyerek iç içe geçmiş yapı tasarlanmıştır. Atlamalı bağlantıları tasarlanan ağdan çıkartılınca standart bir ResNet-50 artık ağı elde edilmiştir [4]. Bu sebeple ağı eğitmek için transfer öğrenimi kullanmışlardır. Sonuçların karşılaştırma aşamasında, Dermatologlardan yardım da alınmıştır.

Son yıllarda derin öğrenme algoritmaları, nöro-görüntüleme verilerine de uygulanmıştır [5]. Bu çalışmalarda kullanılan boru hatları; çoğunlukla görüntü çıkarımı için derin öğrenmedeki modern gelişmeler, otomatik özellik üretimi ve daha kolay analiz için güçlü bir yapı sunmasına rağmen özellik çıkarımı işlemi için çok sayıda işlem aşaması gerektirmektedir. Bu çalışmada, esasen artık ve normal 3B KSA mimarileri ile bu özellik çıkarma adımlarını geçerek sadece benzer performansın nasıl elde edileceği incelemiştir.

Bir diğer çalışma [6] ise Alzheimer hastalığının hafif kognitif bozulmaya karşı sınıflandırılması için önerilen yaklaşımın performansını ve 3B yapısal MRI beyin taramalarının Alzheimer Hastalığı Ulusal Girişimi(AHUG) veri setindeki normal kontrolleri göstermiştir. Önceki ADNI tabanlı çalışmaların, çok sınıflı bir sınıflandırma yapmadığı ve ikili bir sınıflandırma uygulaması gerçekleştirdikleri bilinmektedir. Önerilen yöntemi, önceki çalışmalarla performansını karşılaştırabilmek için ikili birebir sınıflayıcı olarak test edilmiştir. Bu çalışmanın avantajı, karmaşık bir önişleme adımına ihtiyaç duymaması ve model kümelemesi olmadan daha iyi sonuçlar elde etmesidir. Çalışmada MRI görüntülerinin sınıflandırması, iki farklı yaklaşımlı KSA ile karşılaştırılmıştır. İlk yaklaşım, konvolüsyon ve havuzlama katmanlara sahip ortak bir ileri beslemeli ağdır. İkincisi ise modern bir artık sinir ağıdır. Her mimaride katman sayısı, konvülasyon sayısı, her katmandaki filtre sayısı, havuzlama boyutu ve öğrenme oranı verilerini belirlemek için deneme yoluyla ilerlenmiştir. Önerilen metodun ana avantajı; kullanım kolaylığının bulunması ve el yapımı özellik üretimine gerek duymamasıdır. Metodun dezavanatajı ise görüntü üzerinde nesnenin çevrilmesinin değişmez olmasından kaynaklanan hizalama için önceden işlenmemiş görüntüler üzerinde bir uygulamaya dayanmamasıdır.

Algorry vd. [7] gerçek zamanlı nesne tespit ve sınıflandırması için görüntü işleme mimarisi sunmuşlardır. Yaygın olarak bilinen YOLO V2 ve KSA sınıflandırıcısının kombinasyonu, bu çalışmada kullanılmıştır. Görüntüde az yer kaplayan ve birbirine benzeyen nesneleri tespit etmek için kullanılır. Bu teknikle, gerçek zamanlı video işleme yeteneğine ulaşılmıştır. 10 farklı trafik levhasının sınıflandırılması sağlanmıştır. Test sonuçları, Fast R-CNN ve Faster R-CNN gibi yaygın olarak bilinen

tanıma algoritmalarından daha hızlı işleme süresine sahiptir. Eğitim için trafik levhalarından 10 tanesi toplam 1000 adet görüntü üzerinde sağlanmıştır. Dört kademeli bir mimari nesnelerin tanınması ve tespiti için önerilmiştir. Küçük nesnelerin düşük tespit gücünü iyileştirmek için YOLO V2 kullanılmıştır. Bu çalışmadaki ilk aşama, görüntüdeki ilgilenilen bölgenin (Region of Interest-ROI) tespit edilmesidir. Bu durum, YOLO V2 mimarisi ile tek bir trafik levhasına ait sınıftaki tüm etiketli görüntülerle eğitilmiştir. İkinci aşamada, tespit edilen ROI bölgesi 64 piksel genişliğinde kırpılmıştır. Çalışmada ortalama 0.04 sn süreyle nesne tespit ve sınıflandırma gerçekleştirilmiştir.

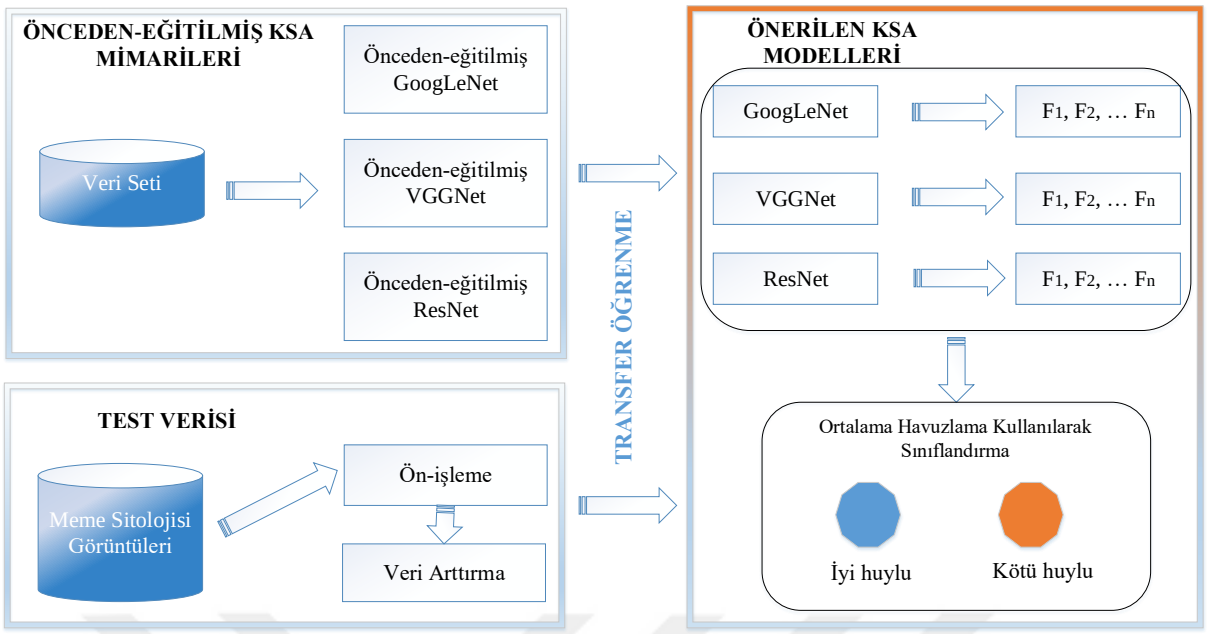
Kuş türü seslerinin otomatik sınıflandırılması, büyük ölçekli zamansal ve mekânsal çevresel izleme için faydalıdır [8]. Bu çalışmada, kuş seslerinin sınıflandırılması için akustik özellikleri, görsel özellikleri ve derin öğrenme incelenmiştir. Derin öğrenme yaklaşımında, genelleştirilmiş özellikleri öğrenme için KSA katmanları kullanılmıştır. Sınıflandırma için ise geleneksel bir tam bağlantı katmanı kullanılmıştır. Daha sonra ise kuş seslerinin sınıflandırılması için bu üç katman birleştirilerek uçtan uca bir model inşa edilmiştir. Görsel ve akustik özelliklerde kuş seslerini sınıflandırmak için iki geleneksel sınıflandırıcı karşılaştırılmıştır.

Tıbbi görüntülemede derin öğrenmenin kullanılması, tıbbi görüntülerin genelde üç boyutlu olması ve üç boyutlu KSA kullanılmasından bahsedilmiştir [9]. Üç boyutlu KSA, iki boyutlu eşdeğerleri kadar iyi gelişmiştir. İki boyutlu KSA'lerin ardındaki fikirleri, üç boyuta yükseltilebileceği görülmüştür [9]. Ancak üç boyutlu bir modelde, yeni bir boyutun eklenmesindeki zorluklardan bahsedilmiştir.

Khan vd. [10] meme kanserinin gelişiminin, erken seviyelerde tespiti ve sınıflandırılması hastaların uygun tedavi görmesini sağlayabilmesi için transfer öğrenme kavramı kullanılarak meme sitolojisi görüntülerinde yeni bir derin öğrenme metodu önermişlerdir. Genel olarak derin öğrenme mimarileri, probleme özgü olarak modellenmiştir. [10]' daki yöntemde, görüntülerden gelen önceden eğitilmiş KSA mimarileri olarak GoogleNet, VGGNet ve ResNet kullanılarak çıkarılmıştır. Bu çalışmada önerilen derin öğrenme mimarisinin blok diyagramı, Şekil 1.1' de gösterilmiştir.

Bir diğer çalışmada [11], klinik patolojilerinin çeşitliliğinin neden olduğu sınıf içi farklılık ve sınıf içi benzerlik zorlukları incelenmiştir. Bu çalışmada, aynı anda birden fazla derin KSA kullanarak ve birbirlerinden karşılıklı olarak öğrenmelerini sağlayarak, bu konuyu ele almak için bir derin öğrenme modeli önerilmiştir.

Rulman makinelerindeki arızaların tespitinde de derin öğrenme modelleri kullanılmaktadır. Motor hızının düşünülmesi işlemi için Han vd. [12] uygun olan KSA' ya görüntü olarak titreşim sinyali kullanılmışlardır. Bhadra vd. [13] iç ırk, dış ırk, rulman elemanlarının arızaları KSA ile sınıflandırmışlardır. Li vd. [14] dişli aşınmasının arıza teşhisi için artırılmış derin seyrek oto-kodlayıcı yöntemi sunmuşlardır. Yöntem, nispeten az ham titreşim sinyali verisi ile dişli aşınma arızasını teşhis etmek için kullanılmıştır.



Şekil 1.1. [10]' da önerilen derin öğrenme mimarisinin blok diyagramı

Yousif vd. [15], bir kamera ile alınmış insan ve hayvan görüntüleri üzerinde ortak arka plan modelleme yapmışlardır. Ardından ise derin öğrenme teknikleri kullanılarak nesnenin insan mı yoksa hayvan mı olduğu tespit edilmiştir. Bu süreçte, çapraz çerçeve görüntü yolu doğrulama ve karmaşıklık-keskinlik analizi ile derin KSA önermişlerdir.

Dolph vd. [16], yapısal MRI'dan türetilen benzer özellikleri kullanarak Alzheimer hastalarının çoklu bir derin öğrenme sınıflandırması gerçekleştirmiştir. İki farklı model önermişlerdir. Her iki model de ince farklılıkları öğrenmiştir.

Anavi vd. [17], X-ışını görüntülerinin elde edilmesine ve özellikle memenin patolojik iyileşmesine yönelik çeşitli yaklaşımları araştırmışlardır. Veri setindeki tüm nesneler, toplam 443 görüntüden oluşmuştur ve bu araştırmanın amacı ise görüntüleri benzerliklerine göre sıralamaktır.

Sevakula vd. [18], kanser tespiti için bir transfer öğrenme tabanlı sınıflandırma yöntemi sunmuşlardır. Özellik seçimi ve normalizasyon teknikleri kullanılmıştır. Seth ve Biswas [19] eletronik-postaları, resim veya metin olarak analiz etmek için KSA' yı kullanmıştır ve sonuç olarak e-posta içeriğinin 'Spam' veya 'Spam değil' olarak sınıflandırmıştır.

Karahan ve Akgül [20], derin öğrenme ile göz tespiti gerçekleştirmiştir. Karabulut [21], derin öğrenme yaklaşımının biyomedikal veriler için başarılı sonuçlar elde edip edemeyeceğini incelemiştir. Bu doğrultuda, derin inanç ağı ve KSA kullanmıştır. Hatipoğlu [22], zaman serilerinin sınıflandırılması için derin öğrenme tabanlı bir yöntem önermiştir. Bu çalışma kapsamında, birçok araştırma alanından elde edilen veriler için hem derin inanç ağları hem de yığınlanmış otomatik kodlayıcı tabanlı mimariler kurulmuş ve eğitilmiştir.

Görüntüler arasında benzerlik oranı yüksek olan nesneleri sınıflandıran bir çalışmada, iki tip vida sınıflandırılmıştır [23]. Bu vidaları görünüş olarak birbirinden ayırt etmek oldukça zordur. Bu vidalar

sadece boyut olarak farklılık göstermektedir. Bu nedenle; vidaların sınıflandırılması için bir referans nesnesi gereklidir. Görüntüden vidanın boyutunu anlamak ancak sabit bir mesafeye kurulmuş kameralardan alınan görüntülerden anlaşılabilir. KSA temel alınarak geliştirilen sınıflandırıcı model ile iki tip vida sınıflandırılmıştır. Referans nesne kullanılmadan elde edilen doğruluk oranı %57.5 iken referans nesne ile elde edilen doğruluk oranı ise %97,5 olmuştur. [23]' te bir referans nesnesi olarak madeni para kullanılmıştır.

Taheritanjani vd. [24] cıvata ve pul sınıflandırması yapılmıştırlar. 20 cıvata ve 14 pulu sınıflandırmak için InceptionV3 ve Tensorflow kullanmışlardır. Doğruluk oranı %99.4 olarak ölçülmüştür. Çalışmada, tek kamera kullanılmıştır. Çalışmada veri setinin oluşturulması sürecinde, ışık yansımalarını ve gölgeleri en aza indiren bir sistem önerilmiştir.

[25]'te, bir cıvata gevşeme açısını tespit etmek için derin öğrenme kullanılmıştır. Kullanılan veri seti oluşturulduktan sonra 'tek atış çoklu kutu dedektör' tekniği ile eğitilmiştir. Modelin doğruluk oranı %91.4 olarak bulunmuştur. Geliştirilen modeli test etmek için farklı aydınlatma koşullarından ve farklı açılardan elde edilen cıvata görüntüleri kullanılmıştır.

[26]' da, KSA tabanlı teknikler kullanılarak havacılık üretim süreci için bağlantı elemanları tanımlanmış ve sınıflandırılmıştır. Önerilen sistem, her görüntü için 0.8 milisaniye işlem süresine sahiptir ve %98.3 doğruluk oranı elde edilmiştir. Bir başka çalışmada, boyut ve renk olarak farklılık gösteren otuz bağlantı elemanını sınıflandırmak için özellik çıkarımı yapılmıştır [27]. Öznitelik çıkarımı için Zernike momentleri renk öznitelikleri ve şekil tabanlı öznitelik çıkarma teknikleri kullanılmıştır. Hem renk hem de şekil tabanlı öznitelik çıkarma tekniklerinden sonra, ileri beslemeli tek gizli katman, sinir ağı tarafından sınıflandırılmıştır. [27]' nin çalışmanın doğruluk oranı %99.89 olarak elde edilmiştir.

Huang [28], vida başlarının arızalı olduğu durumları tespit etmek için KSA ağı oluşturmuştur. Her vida görüntüsü, arızalı veya sağlam olarak etiketlemiştir ve sonuç olarak %95.13 oranında doğruluk elde etmiştir. Song vd. [29], metal vida yüzeylerindeki mikro arızaları bulmak için bir KSA ağı geliştirmişlerdir. Sınıflandırılan arızalar; bir yüzey arızası, yüzey kiri ve soyulmuş vidalardır. Çalışmanın doğruluk oranı %98 olup, her bir görüntüdeki arızayı tespit etmek için geçen süre 1.2 saniye olarak ölçülmüştür.

Wei vd. [30] görüntü işleme ve derin öğrenme kullanarak çevrimiçi durum izleme için pantograf kayan plakalara odaklanmıştır. Bu yöntemin ilk adımında, şerit tespiti ve arıza tespiti hedeflenmiştir. Derin öğrenme ile arızanın türü bulunmuştur. Ayrıca önerilen çalışma; AlexNet, LeNet ve Opt-AlexNet gibi mimarilerde karşılaştırılarak önemli sonuçlar elde edilmiştir.

COVID hastalığının nasıl tespit edileceğine ilişkin bir çalışmada [31], bir hastaneden 4986 adet göğüs BT görüntüsü alınmıştır ve transfer öğrenmeyi test etmek için DenseNet121, DenseNet201, VGG16, VGG19, InceptionResnetV2 ve Xception modelleri kullanılmıştır. İnce ayar, bu göğüs görüntülerine dayalı olarak en iyi COVID sınıflandırma sonuçlarını üreten modelin, DenseNet201 üzerindeki özellik çıkarıcı katmanların dondurulmasıyla %98.18 doğruluk oranıyla DenseNet21 olduğu saptanmıştır.

Başka bir COVID tespit çalışması [32] transfer öğrenmeyi; VGG19, ResNet50V2, DenseNet121 ve MobileNet ile kullanmıştır. İnce ayar, 460 adet görüntü içeren bir veri kümesi için modellerin son katmanlarına 2 boyutlu konvolüsyon, ortalama havuzlama, düzleştirme ve sigmoid katmanlarının eklenmesini içermiştir. En iyi sonuç, MobileNet ile elde edilmiştir.

Sravan vd. [33] 20.639 adet mahsul görüntüsünde, hastalık sınıflandırması için ince ayarlı(fine-tuning) ResNet50 modelini kullanmıştır. Bu model, hiper-parametre değerlerinden bazılarını değiştirmiştir. Ayrıca, bu çalışma ResNet50 modelinin son üç katmanını kaldırmıştır ve yerine yeni katmanlar eklenmiştir: sırayla 2 boyutlu konvolüsyon, ortalama havuzlama, tam bağlantı ve softmax katmanıdır. Her sınıftan 300 adet görüntü ile 15 farklı hastalık durumunun sınıflandırılması %99.26 doğrulukla sonuçlanmıştır.

Kudva vd. [34], rahim ağzının 2198 adet pozitif ve negatif görüntüsüne dayalı olarak rahim ağzı kanseri tespiti için VGG16 ve AlexNet ağları kullanılarak hibrit bir transfer öğrenme modeli önermişlerdir. Hibrit transfer öğrenme yöntemi, AlexNet ve VGG16'nın başlangıç ağırlık değerlerini kullanarak sıfırdan bir KSA ağı oluşturmuşlardır. %91.46 doğruluk oranıyla başarılı sonuçlar elde edilmiştir.

Diğer bir çalışma [35] ise, çiftçilerin, mahsulün büyüme aşamalarında aynı göründükleri için böcek zararlılarını tanımlamakta zorlandıkları gerçeğine dayanmaktadır. Bu problem, böcekleri sınıflandırmak için kullanılan üç veri seti ile derin bir sinir ağı ile çözülmüştür. Birinci veri setinde 40, ikinci veri setinde 24 ve üçüncü veri setinde 40 sınıf bulunmaktadır. [35], böcek sınıflandırması için AlexNet, VGGNet, GoogLeNet ve Resnet TL yaklaşımlarını karşılaştırmıştır. Önceden eğitilmiş ağlar için hiper-parametre ayarı, sınıflandırma performansını iyileştirmiştir: doğruluk oranı ilk veri setinde %96.75, ikinci veri setinde %97.47 ve üçüncü veri setinde %95.47 dir.

Talo vd. [36], beyindeki anormallikleri saptamak için MRI görüntülerini analiz eden KSA tabanlı ResNet-34 transfer öğrenme modelini kullanmışlardır. 613 adet MR görüntüsü üzerinde, beş katlı bir sınıflandırma ile %100 doğruluk elde etmişlerdir.

Beyin tümörlerini sınıflandırmak için yapılan başka bir çalışma [37], önceden eğitilmiş modelin yalnızca Softmax kısmını güncellemiştir. Çalışmada kullanılan ağlar AlexNet, GoogLeNet, SqueezeNet, ResNet50 ve ResNet101 dir. Hiper-parametre ayarlamasından sonra, AlexNet en yüksek doğruluk oranı olarak %99.04' e ulaşmıştır. Yang vd. [38], transfer öğrenmeyi kullanarak yedek parçaları üç kategoriye ayırmıştır. VGG19, Alexnet ve ResNet50 modellerini kullanmışlardır ve VGG19 modeli 0,963 ile en yüksek başarı oranını elde etmiştir.

Başka bir çalışma [39], derin bir KSA kullanarak iyi huylu ve kötü huylu cilt lezyonlarını sınıflandırmıştır. Bu yöntem AlexNet, VGG16, DenseNet, MobileNet ve ResNet ile karşılaştırılmıştır ve 0,9143 doğrulukla en iyi sonuçları verdiği saptanmıştır. Bir başka cilt lezyonu sınıflandırmasında [40], ResNet152V2, DenseNet201 ve Xception, modelleri sırasıyla 0,874, 0,874 ve 0,891 doğruluk oranlarıyla kullanılmıştır.

Kumar vd. [41], kötü amaçlı yazılım görüntülerini sınıflandırmak için hem özellik çıkarıcı olarak hem de sınıflandırıcı olarak VGG16, VGG19, ResNet50 ve InceptionV3 mimarilerini kullanılmıştır. VGG16 mimarisinin son iki katmanını yeni dense katmanları eklemiştir ve diğer ön işleme tabi tutulmuş ağların, yalnızca Softmax katmanlarını güncellemiştir. Yine aynı düşünce ile [42]' de, Softmax katmanındaki yalnızca sınıf sayısı güncellenerek beş farklı çiçek kategorisi sınıflandırılmak üzereVGG16 modelinde ince ayar kullanılmıştır.

21 farklı kedi ve köpek ırkının sınıflandırılması üzerine yapılan bir çalışmada [43], VGG16 ve VGG19 modelleri ile ince ayar yapılmıştır ve var olan ağın sonuna iki adet 4096 boyutlu tam bağlantılı katman eklenmiştir. VGG16 ve VGG19 modelleri, sırasıyla 0.9847 ve 0.9859 doğruluk seviyelerine ulaşmıştır.

Lee [44], ince ayar ile plastik atık sınıflandırması için InceptionResNetV2, VGG19, VGG16, InceptionV3 ve MobileNet modellerini kullanmıştır. Bu çalışma, önceden eğitilmiş modelin sonuna birkaç katman eklemiştir ve modelin hiper-parametreleri üzerinde deneyler yapmıştır.

Qian vd. [45], taze çay yapraklarını sınıflandırmaya yönelik bir model olarak InceptionV3 modelini kullanmışlardır ve 0.98'lik bir başarı oranı elde etmişleridir.

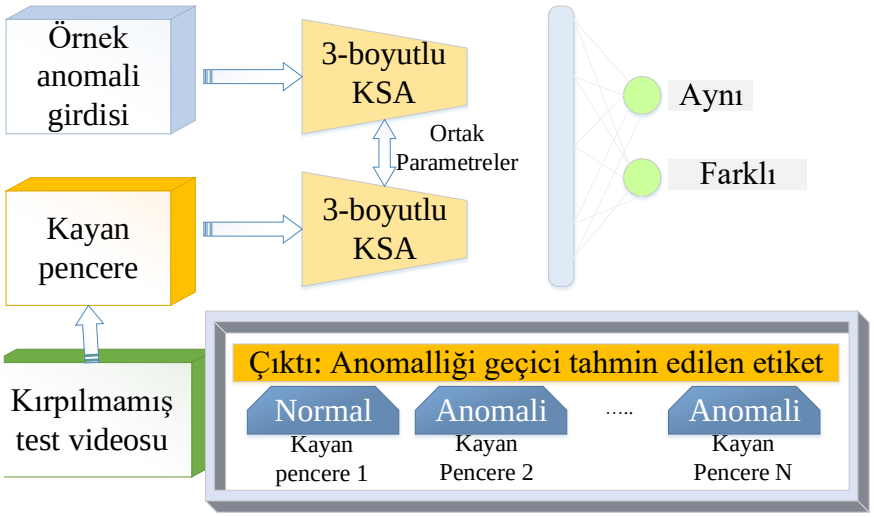
Junaidi vd. [46] civcivleri, kuluçkalık yumurtaları ve çıkmamış yumurtaları sadece sınıflandırıcı katmanları üzerinde VGG16 ve VGG19 modelleri kullanarak sınıflandırmışlardır ve sırasıyla 0.90 ve 0.92 başarı oranları elde etmişlerdir.

Rajayogi vd. [47], hint yemeklerini InceptionV3, VGG16, VGG19 ve ResNet modelleri ile sınıflandırmışlardır. Global ortalama havuzlama ve Softmax katmanları önceden eğitilmiş ağa uygulanmıştır ve InceptionV3 modeli en yüksek doğruluk oranına ulaşmıştır.

Espejo-Garcia vd. [48] besinleri EfficientNetB0, EfficientNetB4, Dense201 ve MobileNetV3 modellerini kullanarak sınıflandırmıştır. MobileNetV3 modeli, modelin sonuna eklenen global ortalama havuzlama, bırakma(dropout) ve Softmax katmanlarıyla en yüksek doğruluk oranına sahip olmuştur.

Hossain vd. [49] tek-seferde öğrenme (TSÖ) ve Siamaese ağları, küçük bir veri seti ile Bangladeş banknotlarını tespit etmek için kullanılmışlardır. Beş farklı banknotu tanımak için yirmi resim kullanılmıştır. Deneysel sonuçlar incelendiğinde, eğitim doğruluğu %97.38 ve doğrulama doğruluğu %94.48 olarak belirlenmiştir. Çalışmada kayıp fonksiyonu olarak düzenlileştirilmiş çapraz entropi kullanılmıştır. Bahsedilen yaklaşım, 50 iterasyon ile çalıştırılmıştır.

Başka bir çalışmada [50], anomali tespiti için 3boyutlu (3D) KSA Siamese (Siyam) ağı kullanılarak iki anomali dizisi arasındaki benzerlik ölçülmüştür ve veri eksikliği sorununu çözmek için TSÖ kullanılmıştır. Ağ bir kez eğitildikten sonra, yalnızca yeni veriler için değil, aynı zamanda yeni sınıflar için de anormallik algılamaya yönlendirilebilir olduğu fark edilmiştir. Yaklaşımdan elde edilen deneysel sonuçlar; 5-way, 10-way ve 14-way ile karşılaştırmalı olarak sunulmuştur. Bu çalışmada önerilen yöntemin blok şeması, Şekil 1.2' de sunulmuştur.



Şekil 1.2. [50]' de önerilen yaklaşımın blok diyagramı

Lungu vd. [51], görüntü çiftleri arasındaki benzerliği ölçmek ve yeni nesneleri hızla sınıflandırmak için TSÖ ve Siyam ağ modeliyle çalışmışlardır. Geliştirilen TSÖ ve Siyam ağ modeli Omniglot, Tiny-Imagenet, CIFAR-100 ve RoShamBo veri setleri kullanılarak test edilmiştir. Çok çözünürlüklü Siyam ağının kullanılması, doğrulukta yaklaşık %10'luk bir artışa katkıda bulunmuştur.

[52]' de, ilişki çıkarımı için göze çarpan özellikleri öğrenen ve olağandışı ilişkiler için yalnızca sınırlı sayıda örnek gerektiren konvolüsyonel bir Siyam ağı önerilmiştir. Önerilen ağ, etki alanına özgü bilgi çıkarımı için 1-shot ve 10-shot öğrenme durumlarında iyi sonuçlar vermiştir.

Agarwal vd. [53], geri dönüştürülebilir atıkları ayırma sorunu için TSÖ' yü kullanarak plastik atıkların görüntü tabanlı sınıflandırmasına yönelik bir yaklaşım sunmuştur. Bu çalışmada, 5 tür plastik atık arasında ayrım yapmak için Siyam ve Triplet Loss (üçlü kayıp) tabanlı KSA kullanılmıştır. WaDaBa veri setinde, %99.74 doğruluk elde edilmiştir.

Pho vd. [54], radar sensör verilerine, TSÖ uygulanarak insan yüzü tanıma üzerine bir çalışma yapmıştırlar. TSÖ yöntemi, etiketli örneklerden özelliklerin çıkarılmasını ve bu özelliklerin yeni örneklere uygulanmasını kolaylaştırmıştır. TSÖ modelinin girdisini oluşturmak için birden fazla kanaldan gelen sinyaller birleştirilmiştir. TSÖ ve Siyam ağının birleştirilmesiyle üretilen yöntemin doğruluk oranı %97.6 olmuştur.

Başka bir çalışmada [55], bir yüz çizimi tanıma çalışması için Siyam ağı ve TSÖ kullanılmıştır. Her görüntünün yalnızca bir fotoğrafı olduğundan TSÖ uygulanmaya karar verilmiştir. [55]' teki çalışmada, iki görüntü arasındaki benzerliği bulmak için aynı mimariye ve ağırlıklara sahip Siyam ağları kullanılmıştır. Çalışmada benzerlik hesabında Öklid uzaklığı kullanılmıştır. 300 iterasyon, sigmoid aktivasyon fonksiyonu ve 10-way TSÖ kullanılarak %100 doğruluk elde edilmiştir. Çalışmada kullanılan veri seti ise CUHK' dır.

Siyam KSA' daki son gelişmeleri araştırmak için bir çalışma yapılmıştır [56]. Bu çalışmada, sözel ve görsel rakamlardan oluşan bir veri seti kullanılmıştır. Geliştirilen Siyam model görüntülerde piksel

mesafesi kullanılarak yüksek doğruluk elde edilmiştir. Hayvan tanımlaması için Siyam ağlarında derin transfer öğrenmesi araştırılmıştır [57]. Vorugunti vd. [58], çevrimiçi imza doğrulamasında yüksek sınıflandırma doğruluğu elde etmek için derin bir konvolüsyonel Siyam ağı önermişlerdir. Üç veri seti MCYT-100, MCYT-330 ve SVC -2004-Task2 kullanılmıştır.

Putra ve Setumin [59], TSÖ yaklaşımını yüz tanıma için farklı aktivasyon özellikleriyle Siyam ağlarında çalıştırmışlardır. Sigmoid, tanh, softmax, softsign ve softplus aktivasyon fonksiyonları denenmiş ve en yüksek performans %92 doğrulukla sigmoid aktivasyon fonksiyonu ve N-way TSÖ ile elde edilmiştir.

Liu vd. [60], Siyam KSA kamera görüntüsünde rastgele sayıda kişiyi tanımlamak için kullanılmışlardır. Gerçek zamanlı TSÖ, daha önce hiç görülmemiş birey sınıflarına uygulanmıştır.

Sokar vd. [61], Siyam KSA ve Destek Vektör Makinesi (DVM) görsel karakter tanıma için kullanılmak üzere önermişlerdir. Mevcut veri seti yetersiz olduğunda, sınıflandırma doğruluğu azalmıştır. Ayrıca, yeni bir sınıf sınıflandırılacağı zaman modelin yeniden eğitilmesi gerekmektedir. Bu durumlardan kaçınmak için, derin Siyam KSA önemli özellikler çıkarmıştır. Eğitim, bir grup sınıf için bir kez gerçekleştirilmiştir. Daha sonra, modeli yeniden eğitmeye veya ince ayar yapmaya gerek kalmadan farklı sınıflar tanınabilir hale gelmiştir. Siyam ağı kullanılarak ağın performansı %12 oranında iyileştirilmiştir.

Li [62], görüntü segmentasyonu işlemini DMÖ ve birkaç-seferde öğrenme (BSÖ) ile yapılmıştır. VGG16 ağının öznitelik çıkarıcı olarak kullanımı ve kayıp fonksiyonu olarak ise karşılaştırmalı kayıp kullanımı ile model tamamlanmıştır. 120.000 iterasyon ve SGD optimize algoritması parametreleriyle 1-way 1-shot durumunun, doğruluğu %45.8 olarak bulunmuştur.

Başka bir çalışmada [63], düşük boyutlu insan görüntüsünün yerleştirmesi için derin özniteliklerin haritalama öğrenimi önerilmiştir. Özellik sunumu ve mesafe metriği, önerilen mimaride birleştirilmiştir. GoogLeNet mimarisi kullanılmıştır. Önerilen çalışmanın önemi, global kayıp fonksiyonunun amaç fonksiyonuna eklenmesi ve pozitif-negatif örneklerin karşılaştırılması, pozitif çift katılığa dayalı bir ağırlık değeri atanmasıdır. Yöntem üç büyük veri kümesi üzerinde test edilmiştir. Veri setlerinin ürettiği doğruluk oranları sırasıyla %35.67, %54.76 ve %63.66' dır.

Duan vd. [64], iki farklı veri setindeki benzerlik kriterini, derin çekişmeli metrik öğrenme, Triplet ağı ve sentetik negatif verileri kullanarak hesaplamışlardır. Yüz ve akrabalık doğrulaması için ayrımcı derin metrik öğrenme (DMÖ) önerilmiştir [65]. Birden çok özellikten genel bilgi çıkarmak için beş farklı veri kümesi üzerinde test edilir. Mahalanobis mesafe kaybı fonksiyonu kullanılmış ve %94.8 oranında performans elde edilmiştir.

Başka bir uygulama alanında, gemi tanıma için üçlü ağlar ve DMÖ kullanılmıştır [66]. 3965 adet gemi görüntüsünün sınıflandırılmasında %60 oranında başarı sağlanmıştır. [67]'de, ürün açıklama verilerini Triplet Loss ile etiketlenmesi amaçlanmıştır. Veri kümesi, her sınıf için dört görüntü içermektedir. ResNet50 mimarisinin parametreleri kullanılarak %81.2 başarı sağlanmıştır.

Li vd. [68], görüntü öznitelikleri ve görüntü etiketleri arasındaki ilişkiyi bulmak için iki yönlü DMÖ önermişlerdir. k-NN (k-en yakın komşu) yöntemi ile görüntü özellikleri hesaplandıktan sonra görüntü özellikleri ile etiket özellikleri arasındaki mesafe hesaplanmıştır. Önerilen yöntem üç aşamadan oluşmaktadır: iki yönlü mesafe metriği, yeniden yapılandırma yöntemi ve sınıflandırma yöntemi. İlk aşamada, görüntüler ve etiketler arasındaki yöntemlerin öğrenilmesi, ikinci aşamada etiket yerleştirmelerinin düzenlenmesi ve sınıflandırmanın yapılması son aşamadır. %66.28 oranında bir başarı oranı elde edilmiştir.

Sıfır-seferde öğrenme (Zero-shot learning-SSÖ), BSÖ ve Relation (ilişki) ağı iki farklı veri setinde sınıflandırma yapılmıştır [69]. Başka bir çalışmada [70], BSÖ ve ilişki ağı, dört farklı veri kümesini sınıflandırmıştır. Önerilen yöntem, özellik modülü, dikkat modülü ve ilişki modülünden oluşmuştur.

Rhee vd. [71], Relation ağı ve lokalize graf konvolüsyon kombinasyonu ile meme kanseri %86.29 oranında sınıflandırmışlardır. Kişi yeniden tanımlaması uygulaması ise [72]' de uygulanmıştır. Üç farklı veri setinde %88.9, %75.6 ve %78.6 sonuç üretilmiştir. Aynı yöntem ile hiper-spektral görüntü sınıflandırması, üç farklı veri kümesi için [73]' te denenmiştir. Bu çalışma, ayrıca üç farklı veri seti üzerinde test edilmiştir. Veri setlerinden elde edilen doğruluk oranları %90.49, %98.03 ve %89.20' dir.

1.2. Tezin Amacı ve Kapsamı

Makine öğrenmesinin alt alanı olan derin öğrenmenin yaygınlaşması, nesne tespit ve sınıflandırma konularında da derin öğrenmeyi popüler yapmıştır. Hem başarılı sonuçlar üretmesi hem de özellik çıkarma işlemine ihtiyaç duymaması sayesinde derin öğrenme ile nesne sınıflandırma konusunda literatürde oldukça fazla çalışma bulunmaktadır. Nesne sınıflandırmada, nesnelerin görsel açıdan birbirine çok benzer olması sınıflandırma işini zorlaştırarak performansı düşürebilmektedir. Benzer nesneler bulunduran veri setlerinin ortak özellikleri; aynı sınıf içindeki nesnelerde farklılık olması ve farklı sınıflar arasındaki nesnelerin benzer olması, her sınıfa ait örnek sayısındaki yetersizlik olmasıdır.

Bu tezde, derin öğrenme yöntemleri kullanılarak benzerlik gösteren nesnelerin sınıflandırılması amaçlanmıştır. Bu amaçla üç farklı türde verilerin bulunduğu veri setleri oluşturulmuştur. Pantograf kömür arızası, bağlantı elemanı görüntüleri ve tekstil makinesi yedek parça görüntüleri tez çalışmasındaki yöntemlerde veri seti olarak kullanılmıştır.

Bu tezin kapsamı şu şekildedir:

- Veri setleri, görüntü arttırma ve çeşitli ön-işlem adımları uygulanarak çalışmalarda kullanılmak üzere hazır hale getirilmiştir.
- Aktarım öğrenme yoluyla ve sıfırdan KSA mimarisi tasarlamayla sınıflandırma uygulamaları yapılmıştır.

- Tasarlanan KSA modelin, modeldeki parametrelerin ve kaynak veri setimizin üçüncü şahıs sorgulamalarına karşın güvenli kılmak adına diferansiyel gizlilik yöntemi uygulanmıştır.
- Nesneler arasındaki benzerliği ölçmek için yeni bir ağ tasarımı yapılmıştır. Bu sayede, benzer nesnelerin sınıflandırılmasında literatüre yeni bir ağ mimarisi kazandırılmıştır.

1.3. Tezin Yapısı

Bu tez, dört ana bölümden oluşmaktadır.

Birinci bölüm, Giriş bölümüdür. Bu bölümde, makine öğrenmesinin ve derin öğrenmenin nesne sınıflandırmaya etkisinden bahsedilmiştir. Ayrıca, nesne sınıflandırma konusunda detaylı bir literatür incelemesi yapılmıştır.

İkinci bölüm, tez kapsamında gerçekleştirilen sınıflandırma çalışmalarını içeren Materyal ve Metot bölümüdür. Tez kapsamında, altı adet nesne sınıflandırma çalışması uygulanmıştır. Bu çalışmaların metotları ve teknik detayları, bu bölümde bahsedilmiştir. İlk yöntemde, pantograf kömür arızasının sıfırdan bir KSA ağı tasarlayarak sınıflandırılması yapılmıştır. İkinci yöntemde, birbirine çok benzeyen vida, cıvata ve somun verilerinin olduğu bağlantı elemanlarının sınıflandırılması için hem sıfırdan bir KSA ağı oluşturulmuş hem de aktarım öğrenme ile sınıflandırılmıştır. Üçüncü yöntemde; veri, model ve model parametre bilgilerinin üçüncü şahıslara karşı gizliliğini sağlayan yöntem geliştirilmiştir. Bu yöntemde, model doğrulama için örme makinesi yedek parça görüntüleri kullanılmıştır. Dördüncü yöntemde, bağlantı elemanlarının TSÖ ile yeni bir Siyam ağı tasarlanmıştır ve bağlantı elemanları sınıflandırılmıştır. Altıncı yöntemde, nesneler arasındaki benzerliği ölçen yeni bir ilişki ağı tasarlanmıştır. Ayrıca BSÖ tekniği de bu çalışmada kullanılmıştır. Bağlantı elemanı ve yedek parça görüntülerinin bulunduğu her iki veri seti de çalışmada uygulanarak elde edilen sonuçlar karşılaştırılmıştır. Son yöntemde ise bağlantı elemanlarının Karşılaştırma ağı ve Prototipik ağlar kullanılarak sınıflandırılması yapılmıştır. Bu çalışmada birkaç-seferde öğrenme yöntemi, geliştirilen mimariler üzerinde uygulanarak sınıf ve örnek sayılarının performansa olan etkisi incelenmiştir.

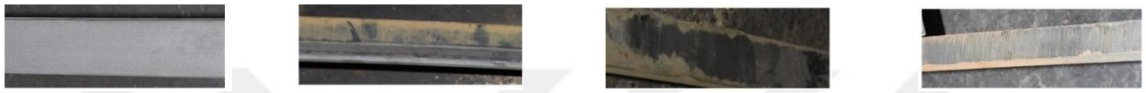
Üçüncü bölüm, Bulgular ve Tartışma bölümüdür. Bu bölümde, ikinci bölümde anlatılan yöntemlere ilişkin bulgular ve sonuçlar verilmiştir. Ayrıca literatürdeki çalışmalarla tez kapsamındaki geliştirilen yöntemler detaylı olarak karşılaştırılmıştır.

Dördüncü bölüm, Sonuçlar bölümdür. Bu bölümde, tezin genel olarak değerlendirilmesi yapılmıştır. Tezde geliştirilen yöntemler ve elde edilen sonuçlar özetlenmiştir.

2. MATERYAL VE METOT

2.1. Veri Setlerinin Hazırlanması

Bu tezde, tezin amacı doğrultusunda birbirine yüksek benzerlikte olan nesnelerden oluşan veri setleri hazırlanmıştır. Kullanılan veri setlerinden ilki, pantograf-katener sistemlerindeki pantografın sürtünmeden kaynaklı oluşan üç farklı yüzey arızası ve normal pantograf yüzey görüntülerinden oluşmaktadır. Bu veri seti üzerinde veri arttırma teknikleri uygulanarak eğitim verimiz toplam 5375 adet olmuştur ve '*VS_Pant*' olarak etiketlenmiştir. Örnek pantograf arızaları görselleri Şekil 2.1' de sunulmuştur.



(a) Normal (b) Az aşınmış (c) Oyuk şeklinde aşınma (d) Fazla aşınma

Şekil 2.1. *VS_Pant* verisetimizin her sınıfına ait örnek görüntüler

Vida, cıvata ve somun türlerinden oluşan bağlantı elemanı veri setimiz tezde kullanılan ikinci veri setidir. Veri arttırma teknikleri uygulanarak eğitim verimiz toplam 1800 adet olmuştur. Bu veri setinde altı çeşit vida, yedi çeşit cıvata ve beş çeşit somun bulunmaktadır ve veri seti '*VS_Baglanti*' olarak etiketlenmiştir. Veri setindeki örnekler incelendiğinde, görsel olarak nesnelerin birbirine çok benzer olduğu fark edilmektedir. Örnek bağlantı elemanı görselleri Şekil 2.2' de gösterilmiştir.



Şekil 2.2. *VS_Baglanti* verisetimizin her sınıfına ait örnek görüntüler

Tez kapsamında kullanılmak üzere tekstil makinesi yedek parça görüntüleri, NİT Örme Tekstil Sanayiden alınmıştır. Alınan görüntüler incelendiğinde, nesnelerin birbirinden gözle ayırt edilmesinin bile oldukça güç olduğu görülmüştür. Bu görüntüler kullanılarak tez için iki farklı veri seti

oluşturulmuştur. Bahsedilen veri setlerinden ilki, tezde kullanılmak için hazırlanan üçüncü veri setidir. Üçüncü veri setinde, altı farklı yedek parça sınıfı bulunmaktadır. Veri arttırma ve ön-işleme tekniklerinin uygulanması ile her sınıfa ait 200 tane görüntü ile toplam eğitim setindeki görüntü sayısı 1200 adet olmuştur. Bu veri setine '*VS_Nit1*' adı verilmiştir. *VS_Nit1*' e ait örnek görseller Şekil 2.3' te verilmiştir.



Şekil 2.3. *VS_Nit1* verisetimizin her sınıfına ait örnek görüntüler

Elde edilen yedek parça görüntülerinden hazırlanan ikinci veri setinde, yirmi farklı sınıf bulunmaktadır. Veri arttırma tekniklerinin kullanılması ile eğitim setimizdeki toplam görüntü sayısı 4100 adet olmuştur. Bu veri seti, tezimizde '*VS_Nit2*' olarak etiketlenmiştir ve veri setine ait örnek görseller Şekil 2.4' te verilmiştir.



Şekil 2.4. *VS_Nit2* verisetimizin her sınıfına ait örnek görüntüler

2.1.1. Veri Setleri için Uygulanan Ön-İşleme Adımları

Görüntülerin Önişlenmesi ve Boyutlandırma

Veri setlerinin hepsi için ortak olarak uygulanan ön-işleme adımları, bu bölümde anlatılmıştır. Veri setindeki görüntülere uygulanan ilk işlem olarak RGB (red-green-blue)-gri tonlamalı renk dönüşümü uygulanmıştır. Daha sonra görüntüler, belirlenen boyuta yeniden boyutlandırılmıştır. Yeniden boyutlandırma işlemi ile görüntünün hiçbir detayını kaybetmeden küçültme sağlanmıştır.

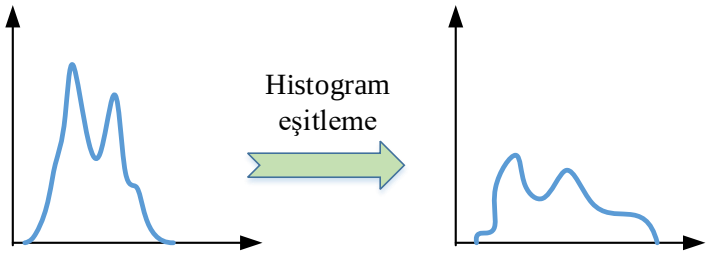
Görüntülere Histogram Eşitleme Uygulanması

Veri setindeki görüntülerin sınıflandırıcının performansını etkilediği bilinmektedir. Görüntünün çözünürlüğü, görüntünün pozlanmasından kaynaklanan ışığın yönü ve görüntüden çıkarılan öznitelikler sınıflandırıcının başarı oranını doğrudan etkilemektedir. Bu bağlamda histogram, görüntülerdeki başlangıç ve bitiş noktalarına dağıtılmaktadır. Böylece görüntünün tüm bölgelerinde piksel değerlerinin dengeli bir dağılımı sağlanmıştır ve piksellerin, görüntünün sadece bir noktasında toplanması gibi durumların önüne geçilmiştir. Histogram eşitleme sonucunda, görüntüdeki nesnenin parlaklık oranı tüm resimde aynı olmaktadır ve nesneden özelliklerin çıkarılması sayesinde bu özellikler sınıflandırıcı performansının artmasına katkı sağlamıştır.

Histogram eşitleme, kontrast değerini tüm görüntü boyunca eşit olarak iyileştirmektedir [74]. Girdi histogramı, her gri seviyede aynı sayıda piksel içerecek şekilde histograma dönüştürme işlemidir. Daha iyi kontrastlı bir görüntü elde etmek için gri düzeyinde dönüştürme sonucu S_k, j. gri tonlamadaki piksel sayısı olarak n_j, giriş görüntüsündeki piksel sayımlarının toplamı olarak n ve istenen toplam gri düzeyi sayısı olarak L tanımlanmıştır. Bu bilgiler doğrultusunda, histogram eşitleme Denklem 2.1 ile hesaplanmıştır.

$$S_k = \sum_{j=0}^{k} \frac{n_j}{n} * (L-1) \;\; , \;\; k = 0, 1, 2, \dots, L-1 \tag{2.1}$$

Histogram eşitleme ile görüntünün düşük görünürlüğü iyileştirilmektedir. Bu işlem sayesinde görüntüdeki olasılık dağılım fonksiyonu doğrusal hale gelmektedir [75]. Histogram eşitlemede, tüm görüntünün yoğunluk dağılımı kullanıldığından ortalama yoğunluk değeri orta seviyeye getirildiğinde bu durum bazı görüntülerde soluk bir etki üretebilmektedir [76]. Dar bir alanda çok fazla yoğunluk dağılımına sahip görüntülerde ise çok sayıda gürültü pikseline neden olabilmektedir. Bu sorunları çözmek için yerel histogram eşitleme teknikleri geliştirilmiştir [77]. Histogram eşitlemedeki ana fikir, görüntüyü ızgara şeklindeki dikdörtgen bölgelere bölmek ve her bölgeye histogram eşitleme uygulamaktır. Optimum bölge boyutları ve bölge sayısı görüntüye göre değişmektedir. Görüntü alt bölgelere ayrıldıktan ve histogram eşitleme uygulandıktan sonra, daha iyi bir görüntü elde etmek için alt bölgeler çift doğrusal enterpolasyon yöntemiyle birleştirilmektedir. Histogram eşitleme sürecine ait esas işlem, Şekil 2.5' te verilmiştir.



Şekil 2.5. Histogram eşitleme sürecindeki temel fikir

2.1.2. Veri Setleri için Uygulanan Veri Arttırma Tekniği

Veri arttırma, orijinal verileri değiştirerek yeni veriler üretmektir. Görüntü üzerinde döndürme, kaydırma, kırpma, rengini değiştirme, ölçekleme; metin üzerinde eşanlamlıları değiştirme, rastgele kelimeleri silme gibi çeşitli tekniklerle işlenmiş sentetik kopyalar oluşturma işlemidir. Bu kopyalar, eğitim verilerine eklenerek büyük miktarda eğitim verisi kullanıma hazır hale gelir. Verileri artırmanın temel amacı, sınıflandırıcının daha doğru tahminler yapmasını sağlamak için modeli yeterli veri ile desteklemektir.

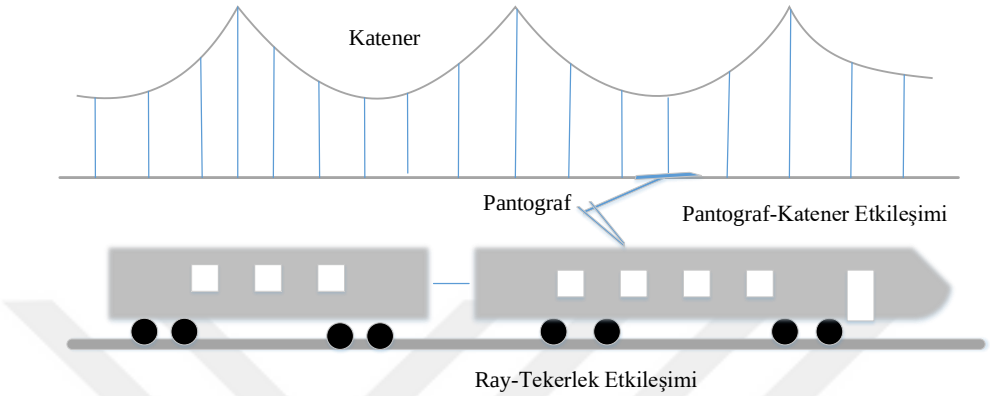
Yapay sinir ağları, belirli bir veri kümesinin özelliklerini çıkarmak için tasarlanmıştır. Ancak küçük veri seti modelleri, verilerin ezberlenmesine neden olmaktadır. Bu durum, modelin düşük doğruluk oranı üretmesine sebep olmaktadır. Başka bir deyişle, veri arttırma, modelin ezberleme sorununu ortadan kaldırmaktadır. Bu tez çalışmasında, verileri artırmak için her bir görüntü x-ekseninde 0.2 oranda, y-ekseninde 0.2 oranda kaydırılmıştır, 30 derecelik bir açıyla döndürülmüştür ve yüzde 20 oranında yakınlaştırılmıştır. Ayrıca görüntüler yatay eksende döndürülmüştür. Böylece geliştirilen KSA modellerinin verileri ezberlemesi engellenmiştir.

Veri arttırma çevrimiçi ve çevrimdışı olarak yapılmaktadır. Veri arttırma çevrimiçi ise, eğitim sırasında veriler arttırılır ve eğitimden sonra veri arttırma yapılmaz [78]. Veri arttırma çevrimdışıysa, eğitim başlamadan önce veriler artırılmaktadır. Her iki yöntemin de avantajları ve dezavantajları vardır. Bu tez çalışmasında, çevrimdışı veri arttırma uygulanmıştır. Bu yöntem, daha fazla depolama alanı gerektirmektedir. Ancak, eğitim süreci çevrimiçi veri arttırmaya kıyasla daha hızlıdır. Çevrimdışı veri arttırmada, sabit sayıda veri üretilirken, çevrim içi eğitim daha fazla veri üretebilmektedir.

2.2. Pantograf Yüzeyindeki Arızaların Derin Öğrenme Tabanlı Sınıflandırılması

Elektrikli raylı ulaşım sisteminde, araçların hareketini ve elektrik enerjisini sağlayan sistem pantograf-katener ikilisidir. Sistemde belirli bir çekme kuvveti ile gerdirilen kol, katener teli ile temas etmekte ve gerekli enerjiyi sağlamaktadır. Araçların hareketi ancak pantograf-katener temasının devamlılığı ile mümkün olmaktadır. Pantograf bileşeni, katener hattı ile temas eden ve genellikle karbon grafit malzemeden yapılmış bir şerittir. Pantograf kömürünün en önemli özelliği, yüksek elektrik

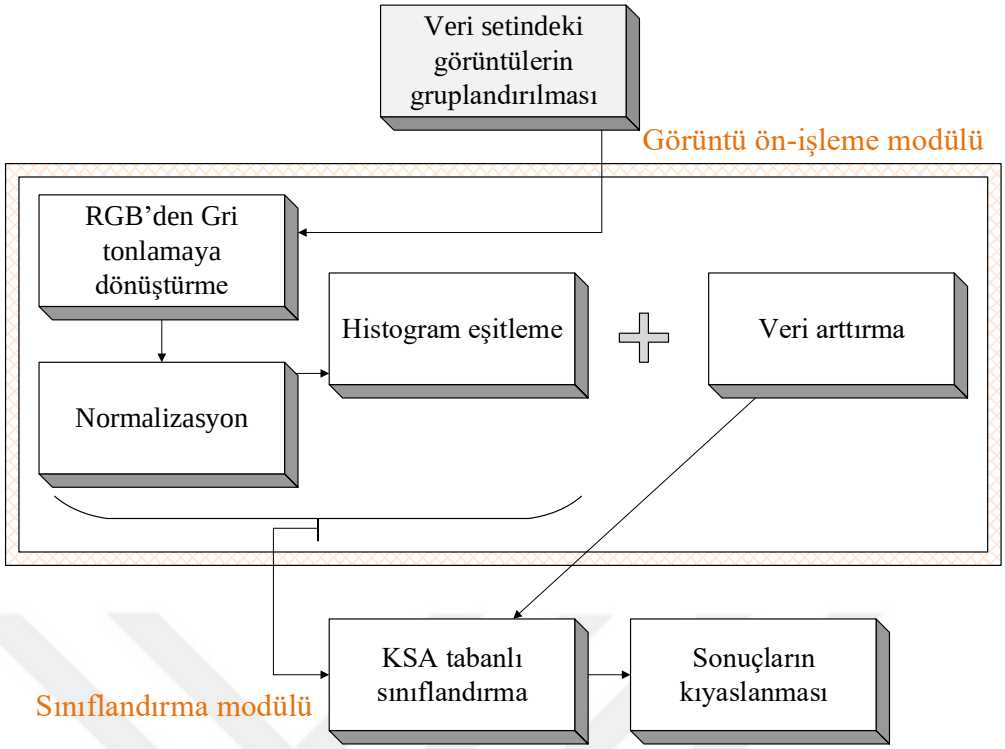
iletkenliği ve katener hattını aşındırmamasıdır. Genel bir pantograf-katener sistemindeki kritik temas noktaları; pantograf-katener etkileşimi ve tekerlek-ray etkileşimidir. Bu bileşenler Şekil 2.6' da gösterilmiştir. Pantograf, elektrikli trenlerde katenerin kontak telinden gelen elektrik enerjisini demiryolu aracına aktaran bileşendir. Akan akım, dönüş iletkeni ve raylar aracılığıyla devreyi tamamlamaktadır.



Şekil 2.6. Genel bir pantograf-katener sistemi ve önemli temas noktaları

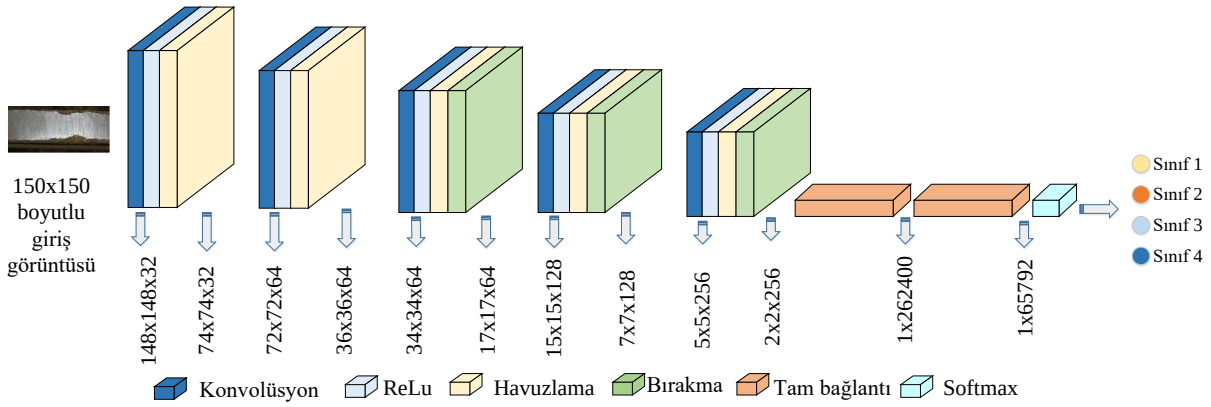
Pantograf ile katener arasında meydana gelen titreşimler ve dış kuvvetler, temas kuvvetinde değişikliğe neden olabilmektedir [79]. Bu durum, sistemde dengesizlik ve kayıplara neden olabilmektedir. Temas değeri, gerekli olan değerden büyükse pantografın temas noktasında aşınma olabilir, eğer daha küçükse elektrik arkı oluşumuna neden olabilmektedir. Bu tür olayların olmasını önlemek için sistemin davranışının iyi analiz edilmesi ve modellenmesi gerekmektedir. Ancak, elektrikli raylı sistemler karmaşıktır ve modellenmesi zordur. Bu sebeplerden ötürü pantograf yüzeyindeki arızanın erken teşhis edilmesini sağlamak üzere derin öğrenme tabanlı arıza sınıflandırma çalışması yapılmıştır.

Pantograf yüzeyindeki arızaların sınıflandırılmasında, KSA ve görüntü işleme tekniklerine dayalı bir yöntem sunulmaktadır. Geliştirilen yöntem, görüntü işleme teknikleri ile desteklenen bir veri seti kullanmanın KSA tabanlı bir derin öğrenme modeline katkısını incelemektedir. Görüntü sayısı ve kalitesi açısından zayıf olan veri seti, görüntü işleme teknikleri ile ön işleme ve veri zenginleştirme işlemine tabi tutulmuştur. Pantograf görüntülerinde aşınmayı daha belirgin ve net hale getirmek için önce gri tonlama dönüşümü yapılmıştır ve ardından farklı pozlanma koşullarının etkisini yok etmek için histogram eşitleme görüntülere uygulanmıştır. Son olarak görüntülere normalizasyon adımı uygulanmıştır. Bu çalışmada, *VS_Pant* veri seti kullanılmıştır. Şekil 2.7 önerilen çalışmanın genel blok şemasını göstermektedir.



Şekil 2.7. Önerilen çalışmanın blok diyagramı

Önerilen KSA mimarisi, 5 adet konvolüsyon katmanı, 5 adet havuzalama katmanı, 5 adet aktivasyon katmanı, 5 adet bırakma katmanı, 2 adet tam bağlantı katmanı ve Softmax sınıflandırıcısından oluşmaktadır. Önerilen KSA mimarisinin blok şeması Şekil 2.8' de ifade edilmiştir.



Şekil 2.8. Önerilen çalışmanın KSA modeli

KSA mimarisi, nesne sınıflandırma alanında yaygın olarak kullanılmaktadır. Derin öğrenme yaklaşımları arasındaki en temel mimarinin, KSA mimarisi olduğu düşünülmektedir. Bu mimariye göre, ilk birkaç aşama konvolüsyon ve havuzlama katmanlarından oluşmakta, bu aşamalardan sonra tam bağlantılı katmanı bulunmaktadır. En son olarak ise sınıflandırma katmanı vardır. KSA mimarisi, arka arkaya yerleştirilmiş birden fazla eğitilebilir parçadan oluşmaktadır. Girdi verileri, KSA üzerinde

alındıktan sonra eğitim süreci katman katman gerçekleştirilmektedir. Son olarak üretilen çıktı ile doğru sonucun elde edilip edilmediği anlaşılmaktadır. Modelin ürettiği sonuç ile istenen sonuç arasındaki fark hata değerini oluşturmaktadır. Geri yayılım algoritması, hata değerini tüm ağırlıklara aktarmak için kullanılmaktadır. Her iterasyonda, ağırlıklar güncellenerek hata değeri azaltılmaktadır. Görüntü sınıflandırma sürecinde, KSA modelleri piksellerden, kenarlardan, kenar kombinasyonlarında ve nesne parçalarından oluşan görselleri birleştirerek nesneleri oluşturmaktadır[80]. Ağın derinliği, araştırmacıların kararına dayanmaktadır. Çünkü ağın verimli olması için derinlik bilgisi gerekli değildir. Nihai modeli oluşturmak ve modeldeki parametre değerlerini belirlemek için deneme yanılma yöntemi kullanılmaktadır. Ağın derinliğinin artması yani gizli katman sayısının artması performansı arttırmaktadır. Ancak belirli bir seviyeden sonra bu derinlik değeri, ağın performansını da düşürebilmektedir. Bu nedenle, ağın gizli katman sayısı ve diğer parametreleri optimum şekilde ayarlanmalıdır.

2.2.1. Konvolüsyon Katmanı

Konvolüsyon işleminde amaç; görüntünün piksel matrislerini kullanarak özellik haritalarının çıkarılmasını sağlamaktır. Bu katmanda, görüntünün piksel matrislerine sol üstten başlanmasıyla sağ alt noktaya inecek şekilde 3x3, 5x5 gibi farklı boyutlarda filtreler(ağırlıklar) ile kaydırma yapılarak uygulanmaktadır. Filtreler uygulanırken her bir kaydırma işleminde, filtre ile aynı boyutta seçilen piksel matrisine çarpma işlemi yapılmakta ve elde edilen matris elemanlarının değerleri toplanarak sonuca ulaşılmaktadır. İşlem, en son piksele gelene kadar tekrarlandıktan sonra özellik haritaları oluşturulmuş olmaktadır.

Giriş katmanından sonraki katmanda uygulanan filtre sayısı, görüntünün derinliği kadardır ve düşük seviye özellik çıkarımı yapılmasını sağlamaktadır. Sonraki konvolüsyon katmanlarında ise görüntünün belirli parçalarını içeren yüksek seviyeli özellikler belirlenmektedir. Görüntü üzerinde odaklanılacak özelliklerin ortalama boyutunun belirlenmesi, ilk katmanda uygulanacak filtre boyutunun belirlenmesinde bir etken olarak kullanılabilmektedir. Filtreler sayesinde bir görüntünün çeşitli özelliklerine odaklanma sağlanmış birden farklı sürümü elde edilmektedir. Uygulanan filtreler, ağırlıklardan oluşmaktadır. Ağırlıklar ise ağın eğitimi ile öğrenilmektedir. Özellik çıkarımının yapıldığı bölüm, KSA' yı diğer yapay sinir ağlarından farklı kılan aşamadır. Ağda ne kadar fazla konvolüsyon katmanı bulursa o kadar fazla karmaşık özellikler belirlenebilmektedir.

Filtreler, KSA mimarisinin ayrılmaz bir parçasıdır. Filtreler, istenilen boyutta işlenebilmektedir. Konvolüsyon işlemi sonucunda, özellik (aktivasyon) haritasına ulaşılmaktadır. Bir özellik haritası, her filtreye özgü özelliklerin keşfedilmesine olanak tanmaktadır. KSA ağının eğitimi sırasında, bu filtrelerin katsayıları, eğitim setindeki her öğrenme iterasyonu ile değişmektedir. Böylece ağ özelliklerinin belirlenmesinde, verilerin hangi bölgelerinin önemli olduğu belirlenmektedir. Konvolüsyon işleminde filtre, görüntü etrafında belirtilen adım miktarı kadar sağa ve sola kaydırılarak hareket ettirilmektedir.

Filtre katsayıları, her bir renk kanalındaki değerlerle çarpılmakta ve bu değeler daha sonra toplanmaktadır. RGB görüntülerindeki üç kanal için bu işlemi yaptıktan sonra, üç bileşenin toplamı, aktivasyon haritasını oluşturmaktadır. Her bir renk kanalı matrisine uygulanan filtre, farklı katsayılara sahip olabilmektedir. Özellik haritasındaki değerler, giriş ve çıkış boyutu arasında aynı yoğunluk aralığını sağlamak için normalleştirilmektedir. Normalleştirmede, her bir renk kanalı için seçilen değerler, filtre katsayılarının toplamına bölünmektedir. Konvolüsyon katmanından sonraki özellik haritasının boyutu, filtre boyutuna, boşluk değerine (padding) ve adım değerine (stride) bağlıdır. Genellikle adım değeri 1'e ayarlanmaktadır. Özellik haritasının boyutu, Denklem 2.2 ile hesaplanmaktadır. Özellik haritasının boyutu olarak FM_{size}, konvolüsyon katmanının boyutu olarak i_{size} ve filtre boyutu olarak k_s tanımlanmıştır.

$$FM_{size} = i_{size} + (2 \times padding - k_s) + 1 \tag{2.2}$$

k_s boyutundaki küçük bir konvolüsyon filtresi için m-katmanlı konvolüsyon gerçekleştirildiği durumda özellik haritasının boyutu Denklem 2.3' teki gibi hesaplanmaktadır. Küçük boyuttaki konvolüsyon filtresi k_s ve tek bir konvolüsyon katmanı k_b arasında Denklem 2.4' teki gibi bir bağıntı elde edilebilmektedir. Konvolüsyon katmanının öznitelik haritasının sayısı 'n' ise, Denklem 2.5'te ifade edildiği gibi N ağırlık sayısı elde edilmektedir. Benzer şekilde, k_s boyutunda m katman kullanan seri konvolüsyon filtresi için N' ağırlık sayısı Denklem 2.6'daki gibi hesaplanabilmektedir. Denklemler incelendiğinde, küçük ölçekli konvolüsyon filtrelerinin uygulanması ağdaki ağırlıkların sayısını önemli ölçüde azaltabildiği sonucuna varılmaktadır [81].

$$FM_{size} = i_{size} + (2m \times padding - k_s \times m) + 1 \times m \tag{2.3}$$

$$k_b = mk_s - (2 \times padding + 1)(m - 1) \tag{2.4}$$

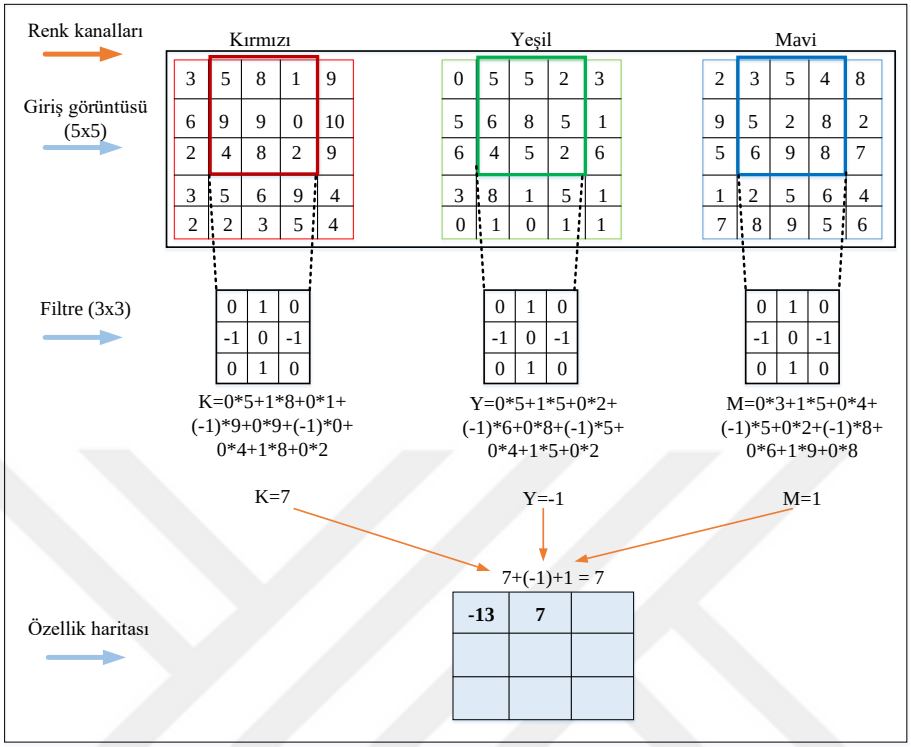
$$N = (k_b^2 \times n + 1) \times n \tag{2.5}$$

$$N' = (k_s^2 \times n + 1) \times n \times m \tag{2.6}$$

Konvolüsyon katmanında, filtre uygulama adımlarında hangi işlemlerin gerçekleştirildiğinden şu şekilde bahsedilmiştir [82]:

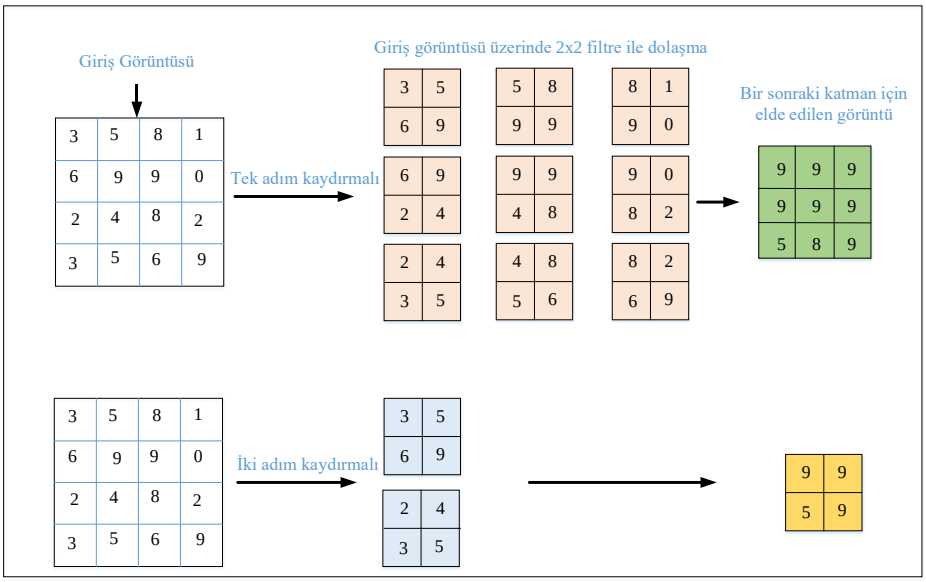
1. Piksel matrislerine uygulanacak filtre boyutu belirlenir.
2. Adım ve boşluk değerleri belirlenir.
3. Piksel matrisinin en sol üst kısmına filtre uygulanır.
4. Filtre, adım değerine göre eşleşme bitene kadar sağa kaydırılarak uygulanır.
5. En sol kısma gelinir ve adım değerine göre aşağıya doğru kaydırma yapılarak filtre uygulanır.
6. Eşleşme bitene kadar sırasıyla dördüncü ve beşinci adımdaki işlemler gerçekleştirilir.

Örnek bir 5x5 boyutundaki piksel matrisine, örnek bir 3x3 boyutundaki filtrenin, özellik haritasını elde etmek üzere uygulanış şekli, Şekil 2.9' da görselleştirilmiştir.



Şekil 2.9. Örnek filtre ve örnek piksel matrisin konvolüsyon işlemi

Seçilen adım değerinin, konvolüsyon işlemine nasıl bir etkisi olduğunu ve nasıl işlediğini gösteren görsel Şekil 2.10' da verilmiştir.

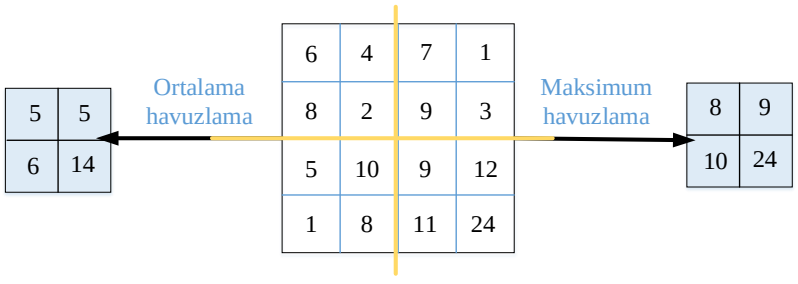


Şekil 2.10. Adım değerinin konvolüsyonda işleyişi

2.2.2. Havuzlama (Pooling) Katmanı

Havuzlama işlemi, özellik çıkarım aşamasında konvolüsyon katmanında sonraki aşamada uygulanmaktadır. Bu işlemin amacı, elde edilen özellik haritalarında boyutlarını indirgemektir. Ayrıca aşırı uyum (overfitting) probleminden kaçınmak için kullanılan bir yöntemdir. Havuzlama işleminde, konvolüsyon katmanında olduğu gibi çeşitli boyutlarda filtreler kullanılarak matrisler üzerinde hesaplamalar yapılmaktadır. Havuzlama çerçevesinin değerinin seçilmesiyle beraber belirlenen adım parametresi dikkate alınarak özellik haritaları üzerinde maksimum ve ortalama havuzlama olmak üzere iki farklı işlem yapılabilmektedir. Bu çalışmada, maksimum havuzlama katmanları kullanılmıştır. Maksimum havuzlama katmanında havuzlama çerçevesi ile özellik matrisinde eşleşen değerlerin en büyüğü alınmaktadır. Ortalama havuzlama katmanında, havuzlama çerçevesi ile özellik matrisinde eşleşen değerlerin ortalaması alınmaktadır. Maksimum ve ortalama havuzlama Şekil 2.11' de gösterildiği gibi yapılmaktadır. Havuzlama katmanında, filtre uygulamasının gerçekleştirilme adımları şu şekildedir:

1. Havuzlama çerçevesinin boyutu belirlenir.
2. Adım parametresi değeri belirlenir.
3. Özellik matrisinin en üst sol köşesine, belirlenen havuzlama çerçevesi kadar havuzlama işlemi uygulanır.
4. Çerçeve, adım parametresi değerine göre en sona gelene kadar sağa kaydırılarak uygulanır.
5. En sol kısma gelinir ve çerçeve, adım parametresi değerine göre aşağıya doğru kaydırılarak uygulanır.
6. Eşleşme bitene kadar sırasıyla dördüncü ve beşinci adımdaki işlemler gerçekleştirilir.



Şekil 2.11. Maksimum ve ortalama havuzlama örneği

Havuzlama işlemi sonucunda; W_1 giriş görüntüsünün genişliği, H_1 giriş görüntüsünün yüksekliği, D_1 giriş görüntüsünün derinliği, f filtre boyutu, s adım değeri olarak tanımlanınca oluşturulan görüntünün boyutu Denklem 2.7' deki gibi hesaplanabilmektedir.

$$SGI = \left(\frac{W_1 - f}{s} + 1\right) \times \left(\frac{H_1 - f}{s} + 1\right) \times (D_1) \tag{2.7}$$

2.2.3. Tam Bağlantı (Fully Connected) Katmanı

KSA tabanlı sinir ağının son adımında, tam bağlantı katmanı bulunmaktadır. Tam bağlantı katmanı adından da anlaşılacağı üzere bir katmandaki bütün düğümlerin, sonraki katmanların hepsine bağlanması işlemidir. Bu katman, hesaplama açısından karmaşıktır.

2.3. Bağlantı Elemanlarının Transfer Öğrenme Tabanlı İnce-Ayar ve Özellik Çıkarım Teknikleri ile Sınıflandırılması

Bağlantı elemanları, endüstriyel makinelerin düzgün çalışması için kritik öneme sahiptir. Kalite kontrol ve arıza teşhis uygulamaları, makinelerdeki bağlantı elemanlarının eksiksiz ve kusursuz olmasını sağlamalıdır. Bağlantı elemanları ile ilgili sorunların tespitinde, bilgisayarlı görme teknolojilerinin kullanımından faydalanabilir. Endüstriyel makinelerdeki olası büyük problemler (bağlantı elemanı arızası, kırılma, aşınma ve yıpranma vb.), bilgisayarlı görme teknolojileri kullanılarak gelişmiş arıza teşhisi sayesinde önlenebilir. Endüstriyel makinelerin durumunu belirlemek için bilgisayarlı görme tekniklerini kullanmak, gereksiz insan kaynağı kullanımından kaçınmaya ve arızalardan kaynaklanan maliyetli harcamalardan kaçınmaya yardımcı olmaktadır. Ayrıca endüstriyel makinelerin güvenliğini sağlamaya yardımcı olmaktadır.

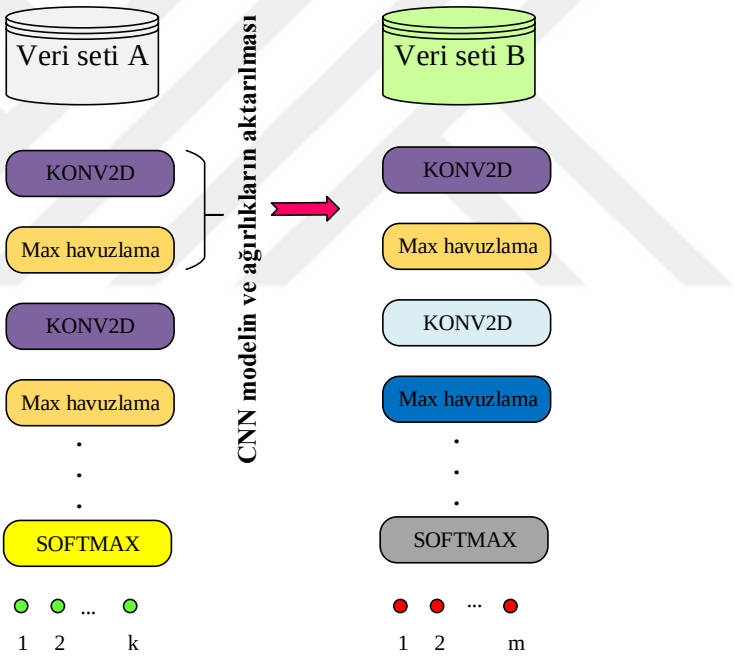
Son zamanlarda, derin öğrenme yaklaşımı, arıza teşhisi, tespiti ve sınıflandırmasında yaygın olarak kullanılmaktadır. Bağlantı elemanlarının birbirine oldukça benzer olması, bağlantı elemanının tanımlanmasını, tespit edilmesini ve sınıflandırılmasını zorlaştırmaktadır. Birçok çalışmada, nesne algılama ve sınıflandırma için derin öğrenme kullanılmıştır ve transfer öğrenme (transfer learning-TL) yöntemi öğrenme kapasitesini geliştirerek daha etkili sonuçlar sağlamaya yardımcı olmaktadır. TL, modellerin büyük veri kümeleri üzerinde eğitildiği ve eğitilen ağın diğer veri kümelerinden hazır olarak kullanılmasına izin veren bir tekniktir. Sınıflandırma çalışmalarında, önceden eğitilmiş bir ağ zamandan tasarruf sağlamakta ve küçük veri kümelerinin sınıflandırılmasında yüksek performans sağlamaktadır. Daha önce büyük ölçekli bir TL veri seti üzerinde eğitilmiş bir modelden elde edilen bilgiler sayesinde sınıflandırma, aynı veya farklı bir model ile farklı bir veri seti üzerinde kullanılmaktadır. Böylece önceden eğitilmiş modeldeki bilgiler aktarılmakta ve ağın performansı iyileşmektedir. Tezin bu bölümünde yaptığımız çalışmada, veri setimiz küçük olduğu için TL yöntemini kullanmamız sayesinde literatürdeki diğer birçok çalışmaya kıyasla daha iyi performansa ulaşılmıştır.

Bu çalışma, TL tabanlı özellik çıkarma (feature extraction-FE) ve ince ayar (fine-tuning-FT) teknikleri kullanılarak 18 bağlantı elemanı tipini sınıflandırmıştır. Bu çalışmada, *VS_Baglanti* veri seti kullanılmıştır. Önerilen yöntem, üç farklı sınıflandırma senaryosu içermektedir. İlk senaryo, önceden eğitilmiş modelin sınıflandırıcı katmanındaki yalnızca sınıf sayısını güncelleyerek ince ayar içermektedir. İkincisinde, önceden eğitilmiş modelin sonuna iki yoğun (dense) katman eklendikten sonra sınıflandırıcı katmanı güncellenerek ince ayar uygulanmıştır. Son senaryoda, önceden eğitilmiş modelin öznitelik çıkarıcı özelliği kullanılarak önceden eğitilmiş ağ ile öznitelikler çıkarılmıştır. Bu

nedenle, son senaryodaki hibrit KSA modeli, girdileri yerel özellikler olarak alması sayesinde daha doğru sonuçlara ulaşmıştır.

2.3.1. Transfer Öğrenme Tabanlı İnce-Ayar ve Öznitelik Çıkarımı

TL, bir problemi çözerken elde edilen bilgileri depolayan ve daha sonra bu bilgileri, ilgili başka bir problemi çözmek için uygulayan bir tekniktir. Yani bir görev için eğitilen bir modelin, başka bir görev için yeniden kullanılabileceği bir makine öğrenme yöntemidir [83]. Bu yöntemi kullanmanın sayesinde, ağın eğitimi daha az zaman ve hesaplama gücü almaktadır. Ayrıca, bir veri kümesindeki görüntü sayısı sınıflandırma için yetersiz olduğunda TL kullanılabilmektedir. Sonuç olarak küçük veri kümelerini TL ile sınıflandırmak, yüksek performans elde edilmesine yardımcı olabilmektedir. TL yaklaşımıyla birlikte kullanılan FT tekniği, bir TL modelinde katman veya parametre değişiklikleri yapmaya olanak sağlamaktadır. Bahsedilen süreç, Şekil 2.12' de gösterildiği gibi yürümektedir.



Şekil 2.12. Temsili transfer öğrenme gösterimi

İnce-Ayar ve Öznitelik Çıkarımı Hakkında Temel Bilgi

Son zamanlarda çeşitli uygulamalarda, önceden eğitilmiş modelleri kullanılmaktadır. TL tekniği, önceden eğitilmiş ImageNet veri kümesi dışındaki yeni görüntüleri sınıflandırabildiğinden gelişmiştir. Önceden eğitilmiş bir modelde FT gerçekleştirmek, modelin hiper-parametrelerini ve/veya ağ yapısını değiştirmektedir. FT, önceden eğitilmiş ağın ağırlık değerleri değiştirilmeden gerçekleştirilmektedir. Bir modele FT uygulamanın 3 yolu bulunmaktadır:

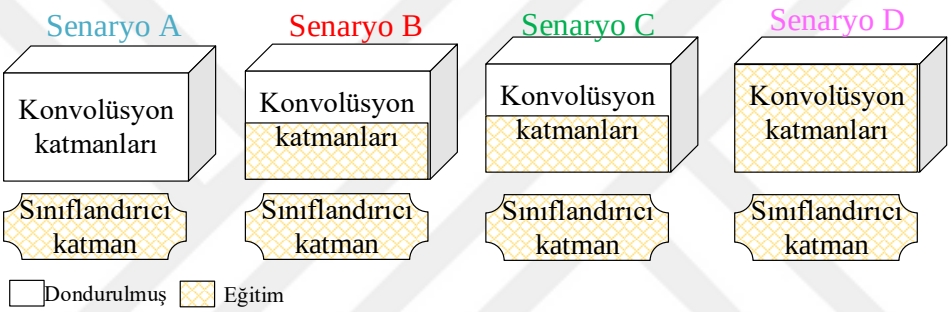
1. *Özniteliklerin Çıkarılması*: Bu yöntem, önceden eğitilmiş ağın bir özellik çıkarıcı olarak kullanılmasıdır. Çıkış katmanı ağdan silindikten sonra tüm ağ sınıflandırılacak yeni veri

kümesinin sınıf sayısını dikkate alarak güncellenmektedir. Çalışmamız bu seçeneği kullanmıştır ve buna '*TL-FE*' senaryosu adını verilmiştir.

2. *Önceden Eğitilmiş Model Mimarisini Kullanma*: Bu yaklaşımda, önceden eğitilmiş modelin ağırlıkları rasgele atanmakta ve tüm ağın yeniden eğitilerek yalnızca ağın mimarisi kullanılmaktadır.
3. *İnce-Ayar*: Önceden eğitilmiş bir ağı kullanmanın bir başka yolu ise ağı yalnızca kısmen eğitmek ve başlangıç ağırlık değerlerini sabit tutmaktır.

İnce-Ayar Uygulanma Yöntemleri

İnce ayarın çalışmalarda kullanım şekli olarak dört temel yol bulunmaktadır. Bu kullanım şekilleri Şekil 2.13' te gösterilmiştir. Bizim çalışmamızda ise bu yaklaşımlardan iki tanesi kullanılmıştır.



Şekil 2.13. İnce-ayar uygulama senaryoları

Senaryo A, yeni veri kümesi küçük olduğunda ve önceden eğitilmiş ağ tarafından kullanılan veri setine çok benzer olduğunda kullanılmaktadır. Bu durumda, ağın sıfırdan eğitilmesine gerek yoktur; sadece çıktı katmanındaki hiper-parametre değeri güncellenir. Çalışmamızdaki '*TL-FT1*' adlı senaryo bu mantığa göre çalışmaktadır.

Senaryo B, yeni veri kümesi küçük olduğunda ve ImageNet veri kümesiyle çok az benzerliği olduğunda kullanılmaktadır. Önceden eğitilmiş modelin, ilk katmanları dondurulmakta ve kalan katmanlar eğitilmektedir. Bu senaryo, ağa yeni katmanlar eklemek, mevcut katmanları kaldırmak ve ağın hiper-parametre değerlerini değiştirmek gibi güncellemeleri içermektedir. Çalışmamızdaki '*TL-FT2*' adlı senaryo bu şekilde işlemektedir.

Senaryo C, yeni veri kümesi büyük olduğunda ve ImageNet veri kümesine oldukça benzer olduğunda kullanılmaktadır. Bu durumda, önceden eğitilmiş modelin mimarisi ve başlangıç ağırlıkları sabit kalmaktadır ve model yeniden eğitilmektedir.

Senaryo D, yeni veri kümesinin büyük olduğu ancak ImageNet veri kümesine çok benzemediği durumlar içindir. Bu durumda, sinir ağı sıfırdan eğitilmektedir.

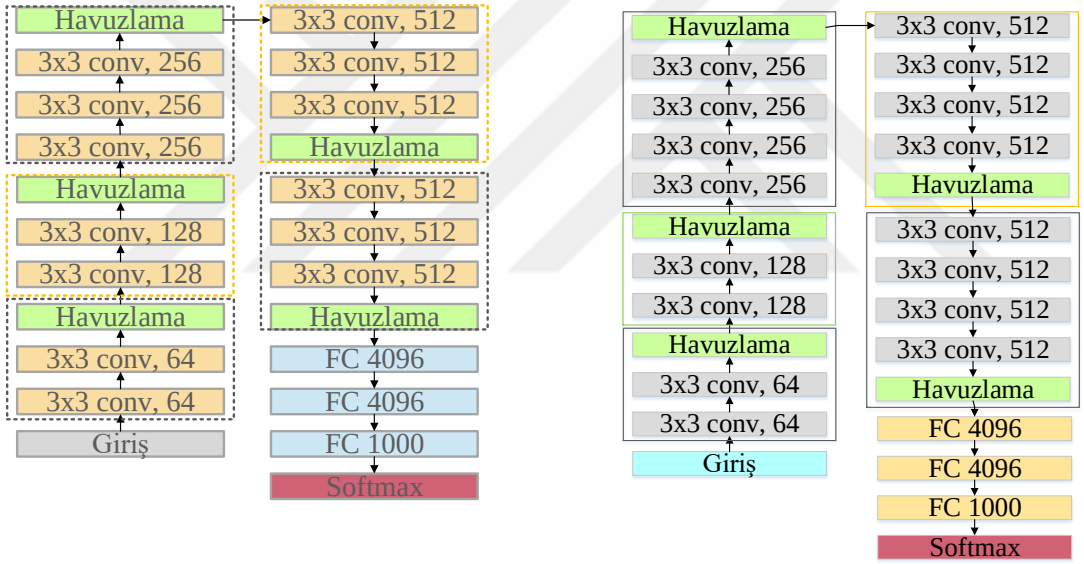
2.3.2. Önceden-Eğitilmiş Modeller

TL ile kullanılan önceden eğitilmiş modellerden bu çalışmada incelenen ve aşağıda kısaca açıklananlar şunlardır: VGG16, VGG19, InceptionV3, InceptionResNetV2, Xception, ResNet50, MobileNet, MobileNetV2, DenseNet169, DenseNet201, EfficientNetB0, EfficientNetB1, EfficientNetB2, EfficientNetB3, EfficientNetB4, EfficientNetB5, EfficientNetB6 ve EfficientNetB7.

VGGNet Modeli

VGG16 modeli, Simonyan ve Zisserman tarafından geliştirilen KSA tabanlı bir mimaridir [84]. VGG16 modeli, 13'ü konvolüsyon ve 3'ü tam bağlantılı olmak üzere 16 katmandan oluşmaktadır. Konvolüsyon katmanlarındaki filtre boyutu 3x3 pikseldir.

VGG19 modelinde, 16 konvolüsyon, 5 havuzlama ve 3 tam bağlantılı katman bulunmaktadır. VGG19 derin bir ağa sahip olduğundan, konvolüsyon katmanında kullanılan filtreler parametre sayısının azaltılmasına yardımcı olmaktadır. Mimaride kullanılan filtrenin boyutu ise 3x3 pikseldir. Her iki model Şekil 2.14' te resmedilmiştir.



(a) VGG16 (b) VGG19

Şekil 2.14. VGG16 ve VGG19 mimarileri

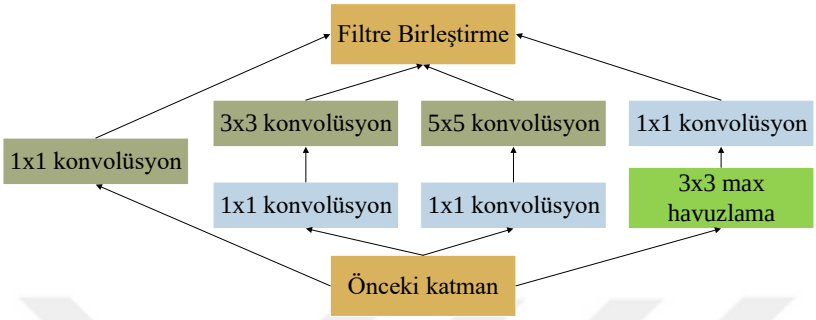
Inception Net Modeli

Bu model, Google tarafından geliştirilen derin öğrenme konvolüsyon mimarisi serisinin üçüncü versiyonudur [85]. Görüntü sınıflandırma alanında kullanılan, en gelişmiş mimarilerden biridir ve birden fazla farklı boyuttaki konvolüsyonel filtreleri yeni bir filtrede birleştiren bir model içermektedir; böylece hem eğitilecek parametre sayısı hem de hesaplama karmaşıklığı azaltılmıştır [86].

InceptionV3, 22 katman kümesi derinliğine sahiptir ve 144 katman içermektedir [86]. Başlangıç modülünde, boyutu azaltmak için çeşitli filtreler kullanılmaktadır. Başlangıç modülündeki filtre

elemanları 1x1, 3x3 ve 5x5 boyutlarındadır. Diğer derin öğrenme mimarilerinden farklı olarak bu model katmanlı bir yapı yerine derin bir yapı oluşturmaktadır. Bu model, Şekil 2.15' te gösterilmiştir.

InceptionResnNetV2, InceptionV3'ün bir varyasyonudur; ancak önemli ölçüde daha derin bir yapıya sahiptir. Başlangıç yapısı ve artık bir bağlantının birleşimidir. Çok boyutlu konvolüsyon filtreleri, ağın eğitim süresini azaltan artık bağlantılarla birleştirilmektedir [87].



Şekil 2.15. InceptionV3 modeli

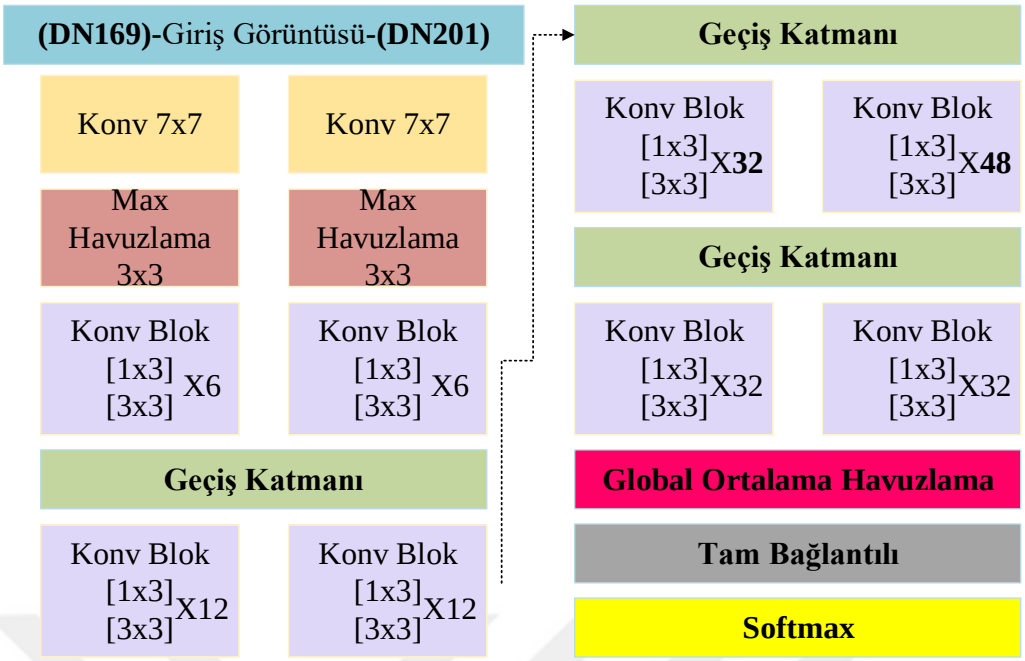
MobileNet Modeli

Bu model, Google tarafından 2017 yılında önerilen bir derin sinir ağıdır. Diğer modellere göre daha küçük ve daha hızlıdır. Derinlemesine ayrılabilir konvolüsyonlar kullanarak her giriş kanalına tek bir filtre uygulamaktadır ve bazı konvolüsyonlarda 1x1 filtre kullanmaktadır [88]. Ayrılabilir katmanlar, derinlik bakımından konvolüsyon katmanlarına benzemektedir; ancak filtreleme ve birleştirme işlemlerini iki katmana ayırarak gerçekleştirdikleri için farklılık göstermektedirler. Bu mimaride, nokta (point) konvolüsyonları hariç toplam 28 katman bulunmaktadır. Softmax katmanını besleyen tam bağlı katman dışında, her katmanı toplu normalleştirme ve yeniden etkinleştirme katmanları takip etmektedir [89].

MobileNetV2 mimarisi ise MobileNet modelinin geliştirilmiş halidir, ancak daha hızlı ve daha verimlidir. Özellik haritalarının boyutu 1x1 konvolüsyon yoluyla küçültülmüştür. Ayrıca ResNet modellerinde kullanılan atlamalı bağlantı tekniği sayesinde hesaplama işlemi daha hızlı olmaktadır [90].

DenseNet Modeli

Sinir ağlarını eğitirken konvolüsyon ve alt örnekleme işlemleri nedeniyle özellik haritaları azalmaktadır. Katmanlar arasında geçiş yaparken de görüntü kalitesi kaybı olmaktadır. DenseNet, görüntü özellik bilgilerini daha etkin kullanmak için geliştirilmiştir [91]. Bu modelde, her katman diğer katmanlara iletilmektedir; böylece herhangi bir katman önceki tüm katmanlardan özellik bilgilerine erişebilmektedir [92]. Böylece özelliklerin ağa yayılma hızı artmaktadır ve parametre sayısı azalmaktadır. DenseNet169 ve DenseNet201 mimarilerindeki katmanların ortak ve farklı parametre sayıları ve değerleri Şekil 2.16' da ifade edilmiştir.



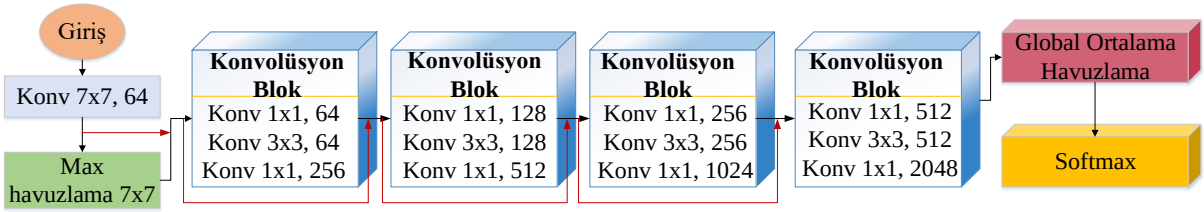
Şekil 2.16. DenseNet169 ve DenseNet201 model mimarileri

EfficientNet Modeli

Bu mimari, sekiz modelden oluşan bir ağdır. Model sayısı arttıkça hesaplanan parametre sayısı nispeten sabit kalmaktadır [93]. EfficientNet, diğer gelişmiş modellerin aksine modeli küçültmeye çalışırken derinlik, genişlik ve çözünürlük açısından ölçeklendirerek daha verimli sonuçlar üretmektedir. EfficientNet mimarisi, kanal boyutunun sekizin katı olmasını gerektirmektedir. Ağın genişliği, derinliği ve çözünürlüğü, bir dizi sabit ölçeklendirme faktörü tarafından eşit olarak ölçeklenmektedir.

ResNet Modeli

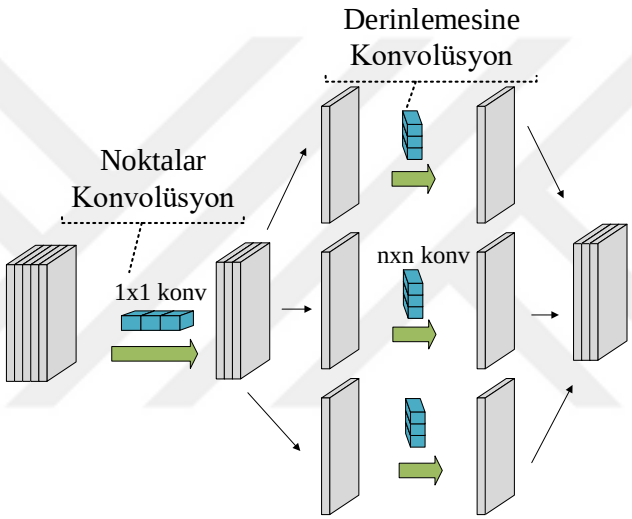
Bu model, sinir ağlarını derinlemesine eğitmenin zorluğunu azaltmak için Microsoft araştırma ekibi tarafından geliştirilen mimaridir. Standart KSA mimarisinden farklı olarak bu mimari kısa yol bağlantıları kullanmaktadır [94]. Kısayol bağlantılarının, fazladan parametresi bulunmamaktadır ve hesaplama karmaşıklığını arttırmamaktadır [95]. Önemli bilgilerin, bir önceki katmandan diğerine aktarılmasına izin vermektedir. Bu mimari, küresel ortalama havuzlamayı ve ağın sonunda tamamen bağlı katmanları içermektedir. ResNet50 modeli Şekil 2.17' de gösterilmiştir.



Şekil 2.17. ResNet mimarisi

Xception Modeli

Bu model, temel olarak InceptionV3 ağının üzerine eklemelerle gelişen bir ağdır. Bu ağı, diğer ağlardan farklı kılan özellik konvolüsyon katmanıdır. Normal bir ağ konvolüsyon bölümü, bir filtreyi genişlik, yükseklik ve derinlik gibi çok boyutlu matrisler üzerinde hareket ettirerek işlemler oluşturmaktadır. Xception mimarisinde ise konvolüsyon bölümü iki farklı yaklaşım sunmaktadır: derinlemesine konvolüsyon ve noktasal konvolüsyondur [96]. Her iki konvolüsyon işlemi Şekil 2.18' de gösterilmiştir. Derinlemesine konvolüsyon, yalnızca bir kanalı işleyerek sonuca ulaşmaktadır. Bu işlem, özellik kaybına neden olacağından bir kanal üzerinden işlenerek elde edilen görüntüye noktasal konvolüsyon uygulanmaktadır. Elde edilen sonuç klasik bir konvolüsyon işlemindeki gibi 1x1xkanal_sayısı'dır [97].



Şekil 2.18. Xception mimarisi

2.3.3. Önerilen Metot

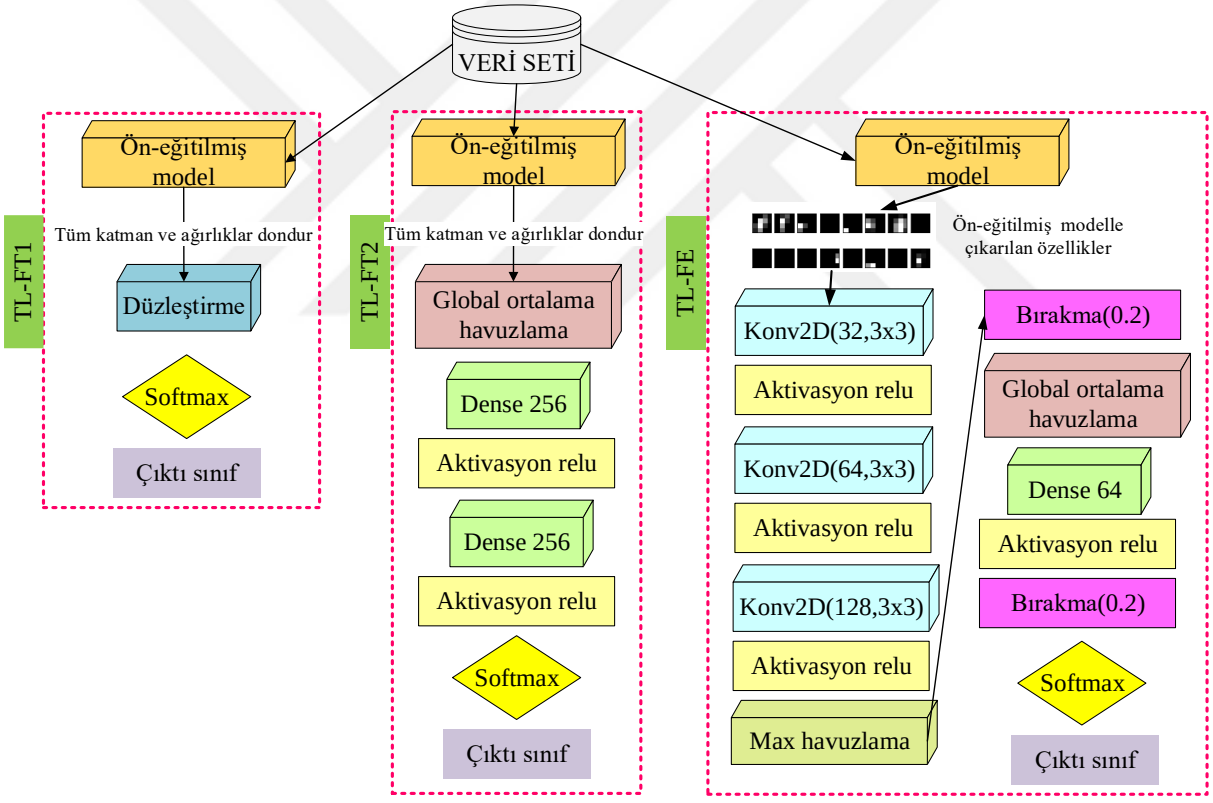
Tezin bu bölümündeki çalışma, vida, cıvata ve somun bağlantı elemanlarının sınıflandırılması için TL tabanlı üç yaklaşım önermektedir. Önerilen yaklaşımlardan ikisi FT, diğeri FE kullanmaktadır. Senaryo 1'de, TL ve FT kullanılarak yalnızca önceden eğitilmiş ağ modelindeki sınıflandırıcı katmanı güncellenmiştir. Senaryo 1, '*TL-FT1*' olarak etiketlenmiştir. Senaryo 2, modelin son katmanlarına TL ve FT eklemektedir. Yani önceden eğitilmiş modele yeni katmanlar eklemektedir. Senaryo 2, '*TL-FT2*' olarak adlandırılmıştır. Senaryo 3 ise hibrit bir yaklaşım önermektedir: TL' nin öznitelik çıkarma yeteneğinden yararlanılarak FE ve KSA birlikte kullanılarak hibrit bir yapı oluşturulmuştur. Senaryo 3 '*TL-FE*' olarak belirlenmiştir. Tüm senaryolar, Şekil 2.19'da bahsedilmiştir.

Sınıflandırılacak veri kümelerinin küçük olduğu göz önüne alındığında, bu çalışmada yüksek performansları ve düşük hesaplama süreleri nedeniyle önceden-eğitilmiş ağlar kullanılmıştır. Her senaryonun uygulanmasından önce veri setinin hazırlanması gerekmektedir.

TL-FT1 senaryosunda, önceden-eğitilmiş ağın sınıflandırıcı katmanı 18 olarak güncellenmiştir, ardından ağın sonuna Düzleştirme (Flatten) katmanı eklenmiştir. Önceden-eğitilmiş ağın ağırlık ve katman değerleri tamamen dondurulmuştur.

TL-FT2 senaryosunda, önceden eğitilmiş ağın tüm katmanları ve ağırlıkları dondurulduktan sonra, ağın sonuna global ortalama havuzlama, 256 yoğun (dense), aktivasyon relu, 256 yoğun ve aktivasyon relu katmanları eklenir. Ağın sonuna, Softmax sınıflandırıcının değeri, 18 olarak güncellenmiştir. Ağa yeni eklenen kısımların eğitilmesi ve eski kısımların dondurulması ile ağ eğitimi tamamlanmıştır.

TL-FE senaryosunda, önceden-eğitilmiş ağın tüm katmanları ve ağırlıkları dondurulduktan sonra, ilk ağın eğitilmesiyle elde edilen özellikler, sıfırdan oluşturulan KSA ağına girdi olarak verilmiştir. Böylece görüntünün yerel özellikleri elde edilmiştir. KSA ağına girdi olarak bu yerel görüntülerin verilmesi ile ağın sınıflandırma performansı önemli ölçüde iyileşmiştir.



Şekil 2.19. Önerilen çalışmanın genel çatısı

2.4. Tekstil Makinesi Yedek Parçalarının Diferansiyel Gizlilik ve KSA ile Sınıflandırılması

Örme makinesi yapı malzemeleri kırılabilir, çatlayabilir ve aşınabilir. Yedek parçaların yenisi ile değiştirilmesi gerektiği bu durumlarda, bu parçaları elde edebilmek için kişilerin örme makinesi parçalarını içeren bir ürün kataloğuna ihtiyacı olmaktadır. Ancak örgü makineleri ürün kataloğunun

maliyeti oldukça yüksektir. Ayrıca örme makinesi yedek parçalarına ilişkin gerçek bilgiler, yalnızca üreticiden alınabilmektedir. Örme makinesi yedek parçasına ihtiyacı olan bir kişinin, yedek parçayı elde edebilmesi için ürünün gerçek kodunu bilmesi gerekmektedir. Örme makineleri ürün kataloğunun fiyatı oldukça yüksek olduğu için bilgisayar yazılımı kullanılarak hangi ürünün hangi koda karşılık geldiğini belirlemek daha yararlı olmaktadır.

Derin öğrenme, nesne sınıflandırma uygulamalarında son zamanlarda oldukça yaygın hale gelmiştir. Literatürde bu konu ile ilgili birçok çalışma bulunmaktadır. Ancak yedek parça sınıflandırması için makine öğrenimi tabanlı algoritmaları içeren herhangi bir yönteme literatürde rastlanmamıştır. Yedek parçalarla ilgili doğru bilgiler yalnızca üreticiden alınabileceğinden, bilgisayarlı görme yöntemleri kullanılarak yedek parçaların sınıflandırılmasına ihtiyaç olmaktadır. Yedek parça bilgilerinin gizliliğinin sağlanması, Diferansiyel Gizlilik (Differential Privacy-DG) algoritmasının kullanılmasını gerektirmektedir. DG yaklaşımı, bir algoritma tarafından sağlanan gizlilik parametrelerini ölçmek için kullanılan bir tekniktir.

Tezin bu bölümündeki çalışma, bu yönüyle bu tekniği kullanan literatürdeki çalışmalara da katkı sağlamaktadır. Bu çalışmada, KSA tabanlı bir yaklaşımla birbirine büyük ölçüde benzeyen yedek parçalar sınıflandırılmıştır. Yedek parça sınıflandırma aşamasından önce eğitim verilerine bazı görüntü ön işleme adımları uygulanmıştır. Böylelikle bu çalışma, verilerin önemli karakteristik noktalarına açıklık getirerek sınıflandırma performansına katkı sağlamayı amaçlamaktadır. Bu çalışmada kullanılan veri seti, *VS_Nit1'* dir. Ön işleme sürecinden geçen eğitim görüntüleri, geliştirilen KSA ağı üzerinde eğitilmiştir. Ancak yedek parça kod bilgilerinin gizliliği önemli olduğundan KSA ağının eğitimi DG yöntemi ile birleştirilmiştir. DG yönteminin başarısına, optimizasyon algoritmasının etkisi de bu çalışmada incelenmiştir. Geliştirilen DG-KSA yöntemi, deneysel olarak test edilmiş ve elde edilen deneysel sonuçlar detaylı olarak Bulgular ve Tartışma bölümünde gösterilmiştir.

2.4.1. Optimizasyon Algoritması

Makine öğrenimi algoritmaları, hatayı en aza indirmek için optimizasyon algoritmaları kullanmaktadırlar. Derin öğrenme modellerinde en sık kullanılan optimizasyon algoritmaları, Gradient Descent, Adam, Adagrad ve RMSProp' tur. Bu algoritmalar, bir sinir ağının daha hızlı eğitilmesini ve daha hızlı tahminler üretmesini sağlamaktadır [98]. Geliştirilen sinir ağını eğitmek için kayıp fonksiyonu tanımlanmalıdır. Kayıp fonksiyon, ağın tahmin sonucunda aldığı değer ile ağın üretmesi beklenen gerçek etiket değerleri arasındaki farktır. Hata hesaplamasında, işlevleri belirlemek için gerçek değer ile tahmin edilen değer arasındaki fark kullanılmaktadır. Kayıp fonksiyonu kullanılarak hesaplanan değerler arasında, en az hataya sahip değerlere ulaşmak için optimizasyon algoritmaları kullanılmaktadır [99]. Bu algoritmaların amacı, hatanın en düşük olduğu bölgeye ulaşmaktır. Matematiksel olarak, bu en düşük bölgeye ulaşmak, optimizasyonun temelidir. Optimizasyonun en temel kısmı, hatanın minimumda çizdiği bölgeye ulaşmak için gradyan inişi bölümüdür.

Optimizasyon algoritmalarının kullanıldığı sisteme sağlanan fayda, kayıp fonksiyonunun düşük olduğu bir değere ulaştığı ağırlık değerlerinin elde edilmesi ve ağın gerçek etiketlerle en yakın tahmini yapmasıdır. Optimizasyon algoritmasının türü ve kayıp fonksiyonu, kullanıcının seçimine bağlıdır. Ağdan beklenen değer ile tahmin edilen değer arasındaki fark miktarı, hata hesaplamasında işlevlerin belirlenmesi için kullanılmaktadır. Kayıp fonksiyonu ile hesaplanan değerler arasında en az hata ile değerlere nasıl ulaşılacağı optimizasyon algoritmaları ile gerçekleştirilmektedir.

Bir eğri üzerindeki bir noktada eğim ölçüldüğünde eğim negatifse, bir sonraki adımda pozitif ilerleme, eğrinin daha da aşağılarına gideceği anlamına gelmektedir. Eğim pozitif ise, negatif ilerleme bir sonraki adıma indireceği anlamına gelir. Kademeli iniş (gradient descent) bu işi yapmaktadır. Optimizasyon algoritmalarında, öğrenme katsayısı da önemli olmaktadır. Algoritma, alt noktayı ararken öğrenme katsayısı kadar ileri geri hareket etmektedir. Öğrenme oranı büyük ise dip nokta sürekli gözden kaçabilmektedir. Eğer öğrenme oranı küçük ise minimum noktayı bulma süreci uzayıp performans düşebilmektedir. Bazı algoritmalar, çözüm olarak uyarlanabilir (adaptif) öğrenmeyi kullanmaktadır. Öğrenme oranı, algoritma çalışırken belirlenmektedir ve hem performans artışında hem de minimum noktaya ulaşmada avantaj sağlamaktadır.

Stokastik Gradyan İnişi (SGD) Algoritması

Yinelemeli bir yöntem olmasına rağmen her yinelemede eğitim veri kümesi, rastgele bir veri grubu örneklenmektedir. Kayıp değeri, modelin tahmin edilen değeri ve eğitim etiketleri karşılaştırılarak hesaplanmaktadır. Bu değere, kayıp oranı denmektedir. Modeldeki parametrelerin değişimine göre kayıp değeri farklılık göstermektedir. Bu eğimler, modeli gerçek etiketi tahmin etmeye yaklaştırmak için her parametreyi nasıl güncellenmesi gerektiğini söylemektedir. Bu algoritma, tüm örnekler üzerindeki eğim değerini hesaplamaktadır. Bu nedenle bu algoritma, diğer algoritmalardan daha yavaştır [100].

Mini-parça SGD ise örnekleri tek tek değil toplu olarak işlemektedir ve daha hızlıdır. SGD algoritması, ağırlık güncellemeleri için tek bir eğitim katsayısı kullanmaktadır ve bu oran, eğitim sırasında değişmemektedir. Uyarlanabilir öğrenme katsayısını kullanan algoritmalar: Uyarlanabilir Gradyan Algoritması (AdaGrad), Ortalama Kare Yayılımı (RMSProp), Uyarlanabilir Moment Tahmini (Adam).

SGD algoritmasında, ağdan üretmesi beklenen değer ile tahmin edilen değer arasında kayıp fonksiyonu hesaplandıktan sonra gradyan (eğim) değeri hesaplanmaktadır ve ardından ağırlık değeri Denklem 2.8' deki gibi güncellenmektedir.

$$\theta_{t+1} = \theta_t - \alpha \frac{\partial}{\partial \theta} L(\theta_t, x^i, y^i) \tag{2.8}$$

Denklem 2.8' de α öğrenme oranıdır, θ ağırlıktır, L kayıp fonksiyonudur, x girdidir ve y ağdan beklenen etiket çıktısıdır. SGD' deki her parametre güncellemesi, tek bir örneğe kıyasla mümkün olduğu

kadar çok eğitim örneği veya bir mini toplu iş olarak hesaplanmaktadır. Bunun iki nedeni bulunmaktadır: birincisi, parametre güncellemesindeki değişimi azaltmakta ve daha kararlı yakınsama getirmektedir; ikincisi, yüksek düzeyde optimize edilmiş matris işlemlerinden faydalanmasını sağlamaktadır [101, 102]. Bu algoritmanın sözde kodu Tablo 2.1' de sunulmuştur.

Tablo 2.1. SGD optimizasyon algoritmasının sözde kodu

Girdi: $\boldsymbol{x, y, f, \delta, L, \alpha}$

Çıktı: $\boldsymbol{\theta}$

while ($\boldsymbol{\delta}$ sağlanmıyor) :

$\boldsymbol{N \leftarrow karistir\ (x)}$ # rastgele mini-parça oluşturma $\boldsymbol{N\ x^1, x^2, \ldots x^N}$ ile $\boldsymbol{y^i}$

$\boldsymbol{\dot{G} \leftarrow \frac{1}{N} \nabla_\theta \sum_i^N L\left(f\left(x^i, \theta\right), y^i\right)}$ # eğimi değerlendirme

$\boldsymbol{\theta \leftarrow \theta - \alpha . \dot{G}}$

End while

Tablo 2.1' de $x, y, f, \delta, L, \alpha, \theta$ parametreleri sırasıyla girdi, etiket değeri, modelin tahmin fonksiyonu, durdurma kriteri, kayıp fonksiyonu, öğrenme oranı ve model parametreleridir. Veriler SGD algoritmasına, anlamlı bir sırayla verilirse zayıf yakınsama meydana gelebilmektedir. Bu durumu önlemek için veriler her eğitim aşamasından önce rastgele karıştırılmaktadır.

Uyarlanabilir Gradyan (AdaGrad) Algoritması

Uyarlanabilir bir öğrenme oranı yöntemidir. Bu sayede, öğrenme hızı değişkenlerine göre değiştirilebilmektedir. Bu algoritma, görüntü tanıma gibi seyrek veriler için daha uygundur. Her parametrenin, kendi öğrenme hızı olduğundan algoritmanın özelliklerine bağlı olarak öğrenme hızı kademeli olarak azalmaktadır. Bu nedenle, öğrenmesi için zamanın belirli bir noktasında eğitilmiş modeli bırakmaktadır. Adagrad' ın en büyük avantajı, öğrenme hızının manuel olarak ayarlanmasına gerek olmamasıdır. Dezavantajı ise öğrenme oranının her zaman azalmasıdır ve belli bir oranda sistem öğrenmeyi durdurabilmektedir [103]. Adagrad' ın farklı öğrenme oranlarına ihtiyaç duymasının nedeni, seyrek öznitelik parametrelerinin öğrenme oranının yoğun öznitelik parametresinden daha yüksek olmasını gerektirmesidir. Çünkü seyrek özelliklerin ortaya çıkma olasılığı daha düşüktür.

$$G_t = \sum_{j=1}^{t} g_j g_j^T \tag{2.9}$$

$$g_t = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial}{\partial \theta} L(x^i, y^i, \theta_t) \tag{2.10}$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\varepsilon I + diag(G_t)}} g_t \tag{2.11}$$

Denklem 2.9' da G_t değeri, eğimlerin dış çarpımının toplamıdır, g_j zaman içindeki eğim tahminidir. Denklem 2.11' de α öğrenme oranıdır, ε sıfıra bölünmeyi önleyen küçük bir değerdir, I bir birim matrisidir. Adagrad algoritmasının sözde kodu Tablo 2.2' de sunulmuştur.

Tablo 2.2. Adagrad optimizasyon algoritmasının sözde kodu

Girdi: $\boldsymbol{x, y, f, \delta, L, \alpha, \varepsilon, I, g_t}$

Çıktı: $\boldsymbol{\theta}$

while (δ sağlanmıyor) :

$\boldsymbol{G_t \leftarrow \sum_{j=1}^{t} g_j \odot g_j}$ # eğimlerin dış çarpımının toplamı

$$\boldsymbol{g_t \leftarrow \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta L(x^i, y^i, \theta_t)}$$

$$\boldsymbol{\theta_{t+1} \leftarrow \theta_t - \frac{\alpha}{\sqrt{\varepsilon I + diag(G_t)}} g_t}$$

End while

Kök Ortalama Kare Yayılım Algoritması (RMSProp)

Adagrad' da, geçmiş eğimlerin karelerinden elde edilen bütün değerleri kullanmak yerine değer miktarını belirli bir çerçeve boyutuyla sınırlandırmıştır [104-105]. RMSProp, salınımları azaltmaya çalışmaktadır; ancak momentumun aksine RMSProp öğrenme hızını değiştirme ihtiyacını da ortadan kaldırmaktadır ve bunu otomatik olarak yapmaktadır. RMSProp, her öznitelik için farklı bir öğrenme oranı tercih etmektedir. Bu algoritmanın sözde kodu Tablo 2.3' te gösterilmiştir.

$$v_t = pv_{t-1} + (1-p) * {g_t}^2 \tag{2.12}$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{v_t + \varepsilon}} * g_t \tag{2.13}$$

Denklem 2.12' de, v_t gradyanların dış çarpımının toplamıdır, g_t gradyandır, p azalan sabittir, α ilk öğrenme oranıdır, ε sıfıra bölünmeyi önleyen küçük bir değerdir.

Tablo 2.3. RMSProp optimizasyon algoritmasının sözde kodu

Girdi: $\boldsymbol{x, y, f, \delta, L, \alpha, \varepsilon, p, g_t}$

Çıktı: $\boldsymbol{\theta}$

while (δ sağlanmıyor) :

$\boldsymbol{v_t \leftarrow \sum_{t=1}^{N} pv_t + (1-p)g_t^2}$ # yerel olarak birikmiş kare eğimler

$$\boldsymbol{\theta_{t+1} \leftarrow \theta_t - \frac{\alpha}{\sqrt{\varepsilon + v_t}} g_t}$$

End while

Uyarlanabilir Momentum Algoritması (Adam)

Bu algoritma, Adagrad ve RMSProp algoritmalarının bir kombinasyonudur. Güçlü ve düşük bellekli bir algoritmadır ve en belirgin gradyan iniş algoritmalarından biridir [104-106]. Bu yöntemde, RMSProp algoritmasında yapıldığı gibi geçmiş eğimlerin karelerinin üstel ağırlıklı ortalamalarının kullanılmasına ek olarak momentum değişiklikleri de önbellekte saklanmaktadır. Bu algoritma, RMSProp ve momentumun birleşimidir. Adam algoritmasının sözde kodu Tablo 2.4' te gösterilmiştir.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{2.14}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{2.15}$$

$$m_t' = \frac{m_t}{1-\beta_1^t} \tag{2.16}$$

$$v_t' = \frac{v_t}{1-\beta_2^t} \tag{2.17}$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{v_t'+\varepsilon}} * m_t' \tag{2.18}$$

Denklem 2.14' te, birinci zaman için β_1, üstel olarak azalan orandır, m_t, θ_t boyunca gradyanların üstel ortalamasıdır. Denklem 2.15' te, ikinci zaman için β_2 üstel azalma oranıdır, v_t, θ_t boyunca karelerin üstel ortalamasıdır. Denklem 2.18' de, α başlangıç öğrenme oranıdır, g_t ise t zamanındaki gradyandır.

Tablo 2.4. Adam optimizasyon algoritmasının sözde kodu

Girdi: $\boldsymbol{x, y, f, \delta, L, \alpha, \varepsilon, g_t, m_t, v_t, \beta_1, \beta_2}$

Çıktı: $\boldsymbol{\theta}$

while (δ sağlanmıyor) :

$\boldsymbol{m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t}$

$\boldsymbol{v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2}$

$\boldsymbol{m_t' = \frac{m_t}{1-\beta_1^t}}$

$\boldsymbol{v_t' = \frac{v_t}{1-\beta_2^t}}$

$\boldsymbol{\theta_{t+1} \leftarrow \theta_t - \frac{\alpha}{\sqrt{\varepsilon + v_t'}} m_t'}$

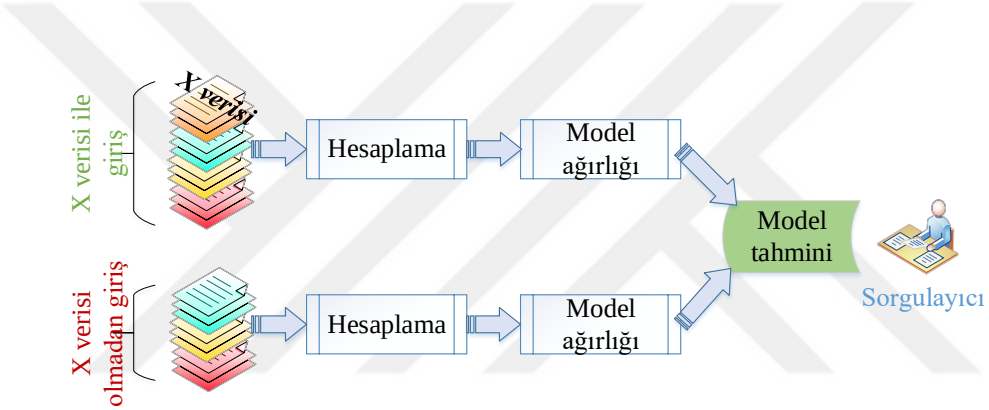
End while

2.4.2. Diferansiyel Gizlilik Metodu

Bu yöntem, bir algoritma tarafından sağlanan gizlilik parametresini ölçme yöntemidir. DG parametresi ile modeli özel veriler üzerinde eğiten algoritmalar oluşturulabilmektedir. DG sayesinde hassas verilerin açığa çıkma riski azalmaktadır. DG ile eğitilmiş bir model, veri setindeki herhangi bir

eğitim örneğinden veya küçük eğitim örneklerinden etkilenmemelidir. Bu şekilde, hassas eğitim verilerinin açığa çıkma riski azaltılmaktadır [107]. DG yöntemi, belirli bir veri setindeki verilerin üçüncü şahıslara ne ölçüde açıklanabileceğini ifade etmektedir [108]. Gizlilik katsayısı hakkında fikir veren kriter, gizlilik maliyetidir. Gizlilik maliyeti, ne kadar yüksek olursa gizlilik kaybı o kadar az olmaktadır. DG kullanımı, hassas eğitim setlerini içeren problemlerde faydalı olmaktadır.

Derin öğrenme, birçok alanda yaygın olarak kullanılmaktadır, ancak bazı gizlilik sorunları bulunmaktadır [108]. Gizliliği korumak için DG yaklaşımını, derin öğrenme eğitimine ve sınıflandırma süreçlerine uygulamak etkili bir çözüm olmaktadır [109]. Eğitim verilerini elde etme ve kaydetme yeteneği, bir saldırganın eğitim veri kümesinde uygun verilerin bulunup bulunmadığını bulmak için yapacağı saldırı için yararlı bir çözüm sunmaktadır [110]. Test veya tahmin işlemi için girdilerin gizliliği, modelin kendi gizliliği, eğitim görüntülerinin gizliliği ve model çıktısının gizliliği model öğreniminin gizlilik bileşenleridir.



Şekil 2.20. Bir sorgulayıcı karşısında diferansiyel gizlilik

DG tabanlı bir derin mimari modelinin tahmin sonucu, veri kümesine bir sorgu eklenmesine veya çıkarılmasına bağlı olarak değişmemektedir; bu nedenle veri kümesinin üçüncü şahıslar tarafından incelenmesi imkânsız hale gelmektedir. Şekil 2.20' de görüldüğü üzere herhangi bir verinin model eğitimde olup olmaması, tahmin sonucunu etkileyemeyeceğinden sorgulayıcı şahıs veri seti ile ilgili bir bilgiye sahip olamayacaktır. Hesaplama sırasında, gürültü eklemek herhangi bir örneğin tespit edilme olasılığı azaltılmaktadır. Ancak bu gürültü eklenmesi, aynı zamanda hesaplama doğruluğunda bir azalmaya neden olabilmektedir. Bu nedenle, doğruluk ve gizlilik koruması arasında bir denge bulunmaktadır. Gizlilik seviyesi, gizlilik maliyeti ile ölçülmektedir ve bu değer gizlilik koruma değeri ile ters orantılıdır. Yani, gizlilik maliyet değeri ne kadar yüksekse, verilerin koruma derecesi o kadar düşük ve örnek bilgilerinin ortaya çıkma olasılığı o kadar yüksek olur.

DG yaklaşımının temeli, verilere rastgele olarak gürültü eklemektir. Modelin güvenliğini sağlamak için model ağırlıkları hesaplanırken gürültü eklenerek yapılmaktadır. İlk olarak, kayıp ağırlıkların eğimi alınarak eğim hesaplanmaktadır. Bu eğimler, daha sonra kırpılmaktadır ve rastgele

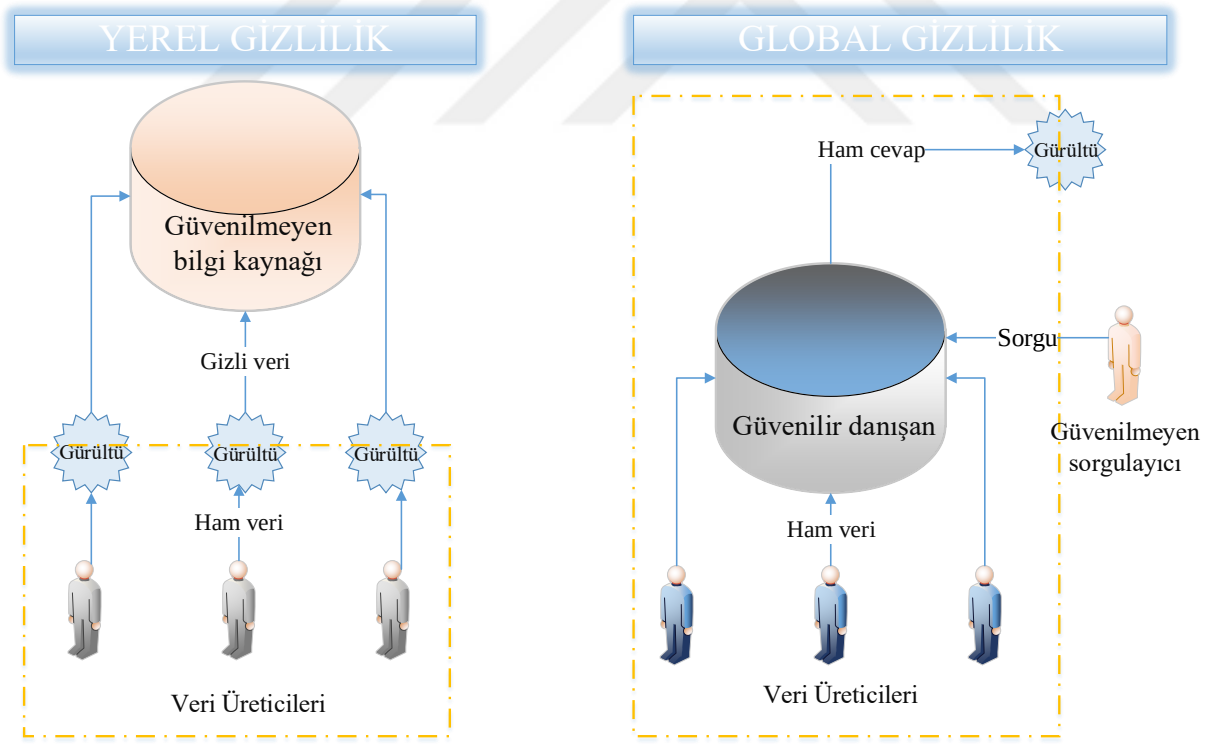
gürültü eklenmektedir. DG model mimarileri, eğitim sürecinde herhangi bir değişikliğe neden olmadan diğer sistemlere uyarlanabilmektedir.

$$(D_1 \, \Delta \, D_2) = 1 \tag{2.19}$$

$$\Pr[f(D_1) \in D^* \leq e^{\varepsilon} \times \Pr[f(D_2) \in D^*]] \tag{2.20}$$

Herhangi bir komşu veri kümesi D_1 için DG' nin gizlilik maliyeti (ε) Denklem 2.20' daki gibi hesaplanmaktadır. D^* sterilize edilmiş bir veri kümesidir. D_1 ve D_2 komşu veri kümeleridir ve yalnızca bir örnekte farklılık göstermektedirler. ε, gizlilik düzenlemesi için kullanılan gizlilik maliyetidir. Gizlilik maliyet değeri, ne kadar düşükse gizlilik koruması o kadar yüksek olmaktadır. DG yaklaşımı, herhangi bir sorgu sonucuna duyarsız olmasını sağlamaktadır. Saldırganın, tek bir örneğin veri setinde var olup olmadığını tahmin etme olasılığı en fazla e^{ε} ile sınırlıdır.

Daha fazla sorgulama yapıldıkça gizlilik maliyeti ε birikmiş olarak kabul edilebilir. Bu maliyet değeri, önceden belirlenmiş bir gizlilik maliyet değerine ulaşana kadar birikmeye devam etmektedir [111-112]. Üç veri örneğinin ürettiği tahmin ile iki veri örneğinin ürettiği tahmin, bir saldırgan tarafından ayırt edilemiyorsa DG' nin elde edildiği anlamına gelmektedir.



Şekil 2.21. Yerel diferansiyel gizlilik ve global diferansiyel gizlilik

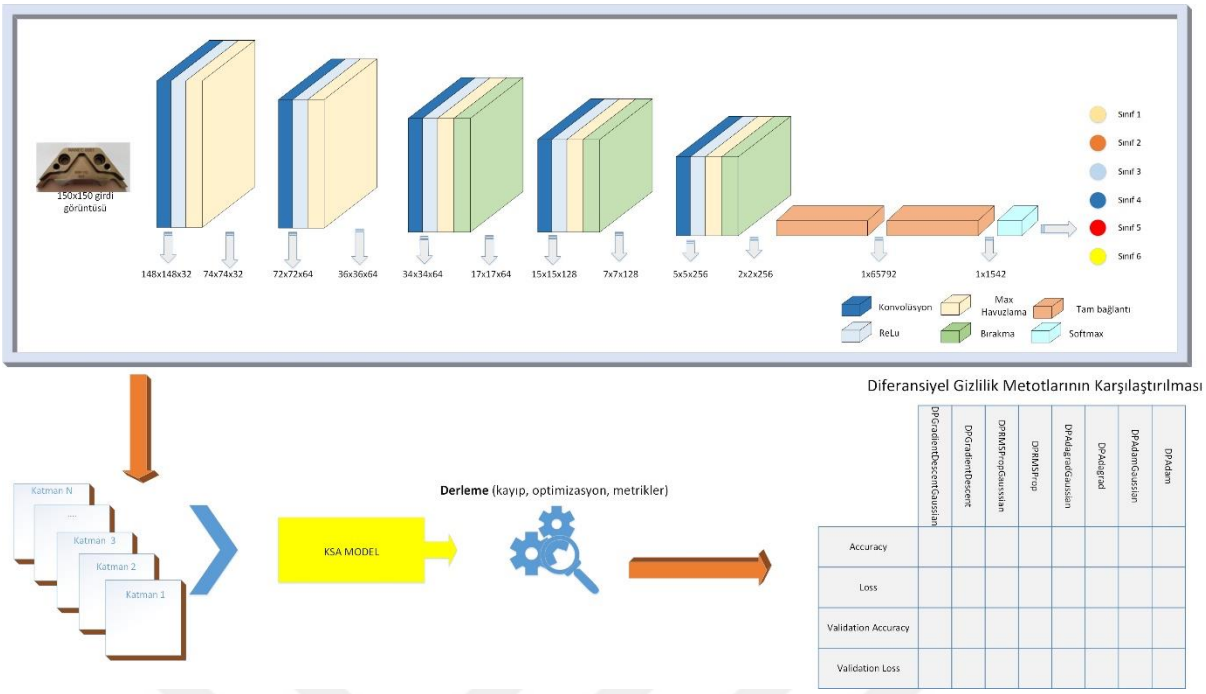
DG yaklaşımı, iki kategoriye ayrılmaktadır: Yerel gizlilik ve Global gizlilik. Bunlar Şekil 2.21' de gösterilmiştir. Yerel gizlilik kavramında, veri toplamada sorgu performansını daraltmak için veriye dayalı gürültü eklenerek gizlilik koruması sağlanmaktadır. Küresel gizlilik kavramında, bir sorgu

çıktısındaki verilerin gizliliği, gürültü eklenmesiyle oluşmaktadır. Bu işlemin bir sonucu olarak sorgu çıktısı, gerçek çıktıdan biraz farklı olmaktadır. Tezdeki çalışmamızda, yerel DG gizliliği kullanılmıştır.

2.4.3. Önerilen Yöntem

Tezin bu bölümündeki çalışmada, tekstil makinelerinin büyük ölçüde benzer yedek parçalarının çoklu sınıflandırılması için bir yöntem sunulmaktadır. Birbirine oldukça benzeyen bu parçaların görüntüleri incelendiğinde, benzerliğin oldukça yüksek olduğu fark edilmektedir. Bu benzerliğin sınıflandırma sürecinde, performansı düşürmesini önlemek için veri setindeki görüntülere ön işleme adımları uygulanarak yedek parçaların ayırt edici noktalarının daha belirgin hale getirilmesi hedeflenmiştir. Ön işleme adımları gerçekleştirilmiş ve üretilen KSA modeli veri setine uygulanmıştır. Bu çalışmada, '*VS_Nit1*' veri seti kullanılmıştır. Eğitim süreci gerçekleştirilirken modele ait görüntülerin ve bilgilerin üçüncü şahıslar tarafından elde edilememesi için DG tekniği uygulanmıştır. Bu aşama önemlidir çünkü yedek parça üreticisi, parçaların bilgilerini gizli tutmaktadır. DG tekniğinin uygulanması sırasında, en başarılı optimizasyon algoritmasını belirlemek için Adam, Adagrad, SDG, RMSProp gibi birçok algoritma çalışmada tek tek uygulanmıştır. Bununla ilgili ayrıntılı performans analizi, Bulgular ve Tartışma bölümünde verilmiştir. Bu çalışmada geliştirilen KSA modeli, birçok konvolüsyon ve havuzlama katmanından oluşmaktadır. KSA modelinin eğitimi boyunca, çeşitli optimizer algoritmaları kullanılmış, en iyi sonucu veren algoritma belirlenmiş ve DG ile gizlilik sağlanmıştır. DG tekniği kullanılarak standart derin öğrenme algoritmalarında sıklıkla kullanılan optimizasyon algoritmalarının gradyanları değiştirilmiştir.

Normal optimizasyon algoritmalarında yapılan iki değişiklik vardır: İlk olarak, her eğimin duyarlılığı sınırlı tutulmuştur. Mini-partide toplanan her eğitim noktasının ve model parametrelerine uygulanan sonuç güncellemelerinin kaç tane eğim hesaplamasının etkileyeceğini kısıtlamak önemlidir. Bu kısıtlama, her eğitim noktasında hesaplanan her bir eğimi kırparak yapılmaktadır. Bu ayarlama, her bir eğitim noktasının, modelin parametrelerini ne kadar etkileyeceğine karar vermesini sağlar. İkinci olarak, eğitim setinde belli bir nokta ile veya bu nokta olmadan çalıştığında optimizasyon algoritması parametre güncellemelerini karşılaştırarak algoritma davranışını rastgele hale getirmeyi amaçlamaktadır. Bu işlem, rastgele gürültüyü örnekleyerek ve kırpılmış eğimleri ekleyerek elde edilmektedir. Önerilen çalışmanın blok şeması Şekil 2.22' de gösterilmiştir. DP-SGD, DP-Adagrad, DP-RMSProp ve DP-Adam algoritmalarının sözde kodları sırasıyla Tablo 2.5, Tablo 2.6, Tablo 2.7, Tablo 2.8 de gösterilmiştir.



Şekil 2.22. Önerilen yöntemin genel gösterimi

Tablo 2.5. DG-SGD optimizasyon algoritmasının sözde kodu

Girdi: $x, y, f, \delta, L, \alpha, C, \mu, I$

Çıktı: θ

while (δ **sağlanmıyor**) **:**

$N \leftarrow shuffle\ (x)$ # rastgele mini-parti $N\ x^1, x^2, \ldots x^N$ ile y^i oluşturma

$\dot{G} \leftarrow \frac{1}{N} \nabla_\theta \sum_i^N L\left(f\left(x^i, \theta\right), y^i\right)$ # eğim hesaplama

for $j \in N$ **do**

$\dot{G}_j \leftarrow \dot{G}_j / \max\left(1, \frac{\|\dot{G}_j\|_2}{C}\right)$ # l2_norm_clip eğim hesaplama

$\dot{G}_j \leftarrow \frac{1}{N}\left(\sum_j^N \dot{G}_j + \mathcal{N}\left(0, \mu^2,\ N^2, I\right)\right)$ # gürültü ekleme

$\theta \leftarrow \theta - \alpha . \dot{G}$

End for

End while

Tablo 2.6. DG-Adagrad optimizasyon algoritmasının sözde kodu

Girdi: $x, y, f, \delta, L, \alpha, \varepsilon, I, g_t, C, \mu$

Çıktı: θ

while (δ **sağlanmıyor**) **:**

$N \leftarrow shuffle\ (x)$ # rastgele mini-parti $N\ x^1, x^2, \ldots x^N$ ile y^i oluşturma

$G_t \leftarrow \sum_{j=1}^{t} g_j \odot g_j$ #eğimlerin dış çarpımının toplamı

$g_t \leftarrow \frac{1}{N} \sum_{i=1}^N \nabla_\theta L\left(x^i, y^i, \theta_t\right)$

for $j \in N$ **do**

$g_l \leftarrow {g_l}/{\max(1, \frac{\|g_l\|_2}{C})}$ #l2_norm_clip eğim hesaplama

$g_l \leftarrow \frac{1}{N}(\sum_l^N g_l + \mathcal{N}(0, \mu^2, N^2, I))$ # gürültü ekleme

$\theta_{t+1} \leftarrow \theta_t - \frac{\alpha}{\sqrt{\varepsilon I + diag(G_t)}} g_t$

End for

End while

Tablo 2.7. DG-RMSProp optimizasyon algoritmasının sözde kodu

Girdi: $x, y, f, \delta, L, \alpha, \varepsilon, p, g_t, C, \mu, I$

Çıktı: θ

while (δ **sağlanmıyor**) **:**

$N \leftarrow shuffle\ (x)$ # rastgele mini-parti $N\ x^1, x^2, \ldots x^N$ ile y^i oluşturma

for $j \in N$ **do**

$v_j \leftarrow p \sum_j^N v_t + (1-p) g_t^2$ # kare eğimlerin biriktirilmesi

$v_j \leftarrow v_j / \max(1, \frac{\|v_j\|_2}{C})$ # l2_norm_clip eğim hesaplama

$v_j \leftarrow \frac{1}{N}\left(\sum_j^N v_j + \mathcal{N}(0, \mu^2, N^2, I)\right)$ # gürültü ekleme

$\theta_{t+1} \leftarrow \theta_t - \frac{\alpha}{\sqrt{\varepsilon + v_t}} g_t$

End for

End while

Tablo 2.8. DG-Adam optimizasyon algoritmasının sözde kodu

Girdi: $x, y, f, \delta, L, \alpha, \varepsilon, g_t, m_t, v_t, \beta_1, \beta_2, \mu, C, I$

Çıktı: θ

while (δ **sağlanmıyor**) **:**

$N \leftarrow shuffle\ (x)$ # rastgele mini-parti $N\ x^1, x^2, \ldots x^N$ ile y^i oluşturma

$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$ eğimlerin dış çarpımının toplamı

$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$

$m_t' = \frac{m_t}{1-\beta_1^t}$

$v_t' = \frac{v_t}{1-\beta_2^t}$

for $j \in N$ **do**

$\theta_z \leftarrow \theta_z / \max(1, \frac{\|\theta_z\|_2}{C})$ # l2_norm_clip eğim hesaplama

$\theta_z \leftarrow \frac{1}{N}\left(\sum_z^N \theta_z + \mathcal{N}(0, \mu^2, N^2, I)\right)$ # gürültü ekleme

$\theta_{z+1} \leftarrow \theta_z - \frac{\alpha}{\sqrt{\varepsilon + v_z'}} m_z'$

End for

End while

DG' de gizliliği sağlamak üzere özel üç hiper-parametre vardır. '*l2 norm clip -γ*' hiper-parametresi, model parametrelerini değiştirmek için kullanılan her eğimin maksimum öklid normudur. Bu hiper-parametre değeri, optimizasyon algoritmasının hassasiyetini bireysel eğitim noktalarına bağlamak için kullanılmaktadır. '*gürültü_çarpanı- ϑ*' hiper-parametresi, eğitim sırasında örneklenen ve eğimlere eklenen gürültü miktarıdır. Bu gürültü ekleme işlemini uygulamak, gerekli olmasa da ne kadar fazla gürültü eklenirse gizlilik o kadar iyi olmaktadır. '*mikro-parti boyutu*' değeri, her bir veri grubunu bölümlere ayırmaktadır. Her mikro-parti değeri, tek bir eğitim örneğini içermektedir. '*öğrenme_oranı- α*' hiper-parametresi, ne kadar yüksek olursa her güncelleme o kadar önemli hale gelmektedir. Güncellemeler gürültülü ise düşük öğrenme oranı eğitim sürecinin yakınsamasına yardımcı olmaktadır. Bu hiper-parametre, normal optimizasyon algoritmalarına zaten dahil edilmiştir. Bu çalışmada gerçekleştirilen DG algoritmalarındaki en yüksek doğruluk oranı elde edilen hiper-parametre ayarlamaları, Tablo 2. 9' da gösterilmektedir.

Tablo 2.9. DG algoritmalarındaki hiper-parametre ayarlamaları

Hiper-parametre	Değer
İterasyon	50
Parça-boyutu	128
L2 norm clip (γ)	1.0
Gürültü çarpanı (ϑ)	0.001
Öğrenme oranı (α)	0.001
Mikro-parça boyutu	1
DG Toplam sorgu	Gaussian ($\gamma, \gamma * \vartheta$)

2.5. Bağlantı Elemanlarının Siyam KSA ve Tek-Seferde Öğrenme ile Sınıflandırılması

Bağlantı elemanları, makinelerin düzgün çalışması için önemlidir. Bağlantı elemanı arızalarının bilgisayarla görü tabanlı tespiti, büyük sorunların oluşmasını engellemektedir. Bilgisayarlı görü tekniklerinin gelişmesiyle birlikte nesne tespiti ve sınıflandırması üzerine birçok çalışma yapılmıştır. Bağlantı elemanlarının birbirine çok benzer olması nesne sınıflandırmasını zorlaştırmaktadır. Derin öğrenme tabanlı nesne sınıflandırma uygulamalarında, her sınıfa ait yeterli sayıda görüntü olmasına rağmen büyük veri setlerine ihtiyaç duyulmaktadır. Ayrıca veri setine yeni bir nesne eklendiğinde, konvolüsyonlu sinir ağı modelinin yeniden eğitilmesi ve ağ parametrelerinin yeniden ayarlanması gerekmektedir. Derin metrik öğrenme tabanlı, Siyam ağlarında bu sorunlar bulunmamaktadır. Siyam ağına yeni bir sınıf eklemek oldukça basittir. Tezin bu bölümündeki çalışmada, bağlantı elemanı görüntülerinin her sınıfta bir adet bulunması ile yeni bir veri seti tasarlanmıştır. Amacımız: çok sayıda verisi bulunmayan bağlantı elemanlarının da sınıflandırma sürecine kolayca adapte edilebilmesini sağlamaktır. Bunu gerçekleştirebilmek üzere Tek-Seferde Öğrenme (One-Shot Learning-TSÖ) tekniği kullanılmıştır. Siyam ağları, görüntü girişleri arasındaki benzerliği ölçmektedir. Tezin bu bölümündeki

çalışmada 6 vida, 7 somun ve 5 cıvata türü sınıflandırılmıştır. Önerilen çalışmanın tek-seferde öğrenme aşaması, 3-way(sınıf), 5-way ve 10-way olarak incelenmiş ve performans sonuçları karşılaştırmalı olarak Bulgular ve Tartışma bölümünde detaylı olarak sunulmuştur. Ayrıca sınıf sayısının incelenmesi ile beraber iterasyon sayısı, kayıp fonksiyonu ve optimizasyon algoritmasının da öğrenme sürecine katkısı Bulgular ve Tartışma bölümünde detaylandırılmıştır.

2.5.1. Tek-Seferde Öğrenme

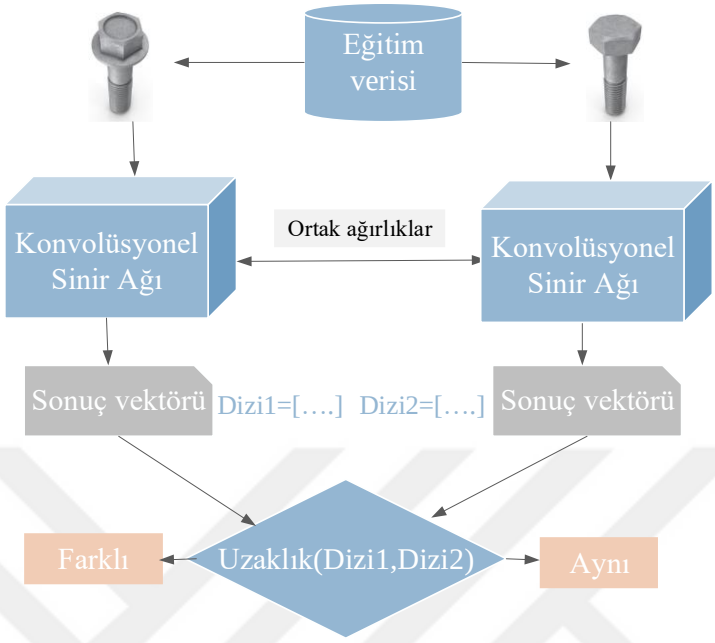
Öğrenme sürecinde, her sınıfta tek verinin bulunması durumudur. Bu duruma örnek olarak bir uçak firmasının yolcuları uçağa alırken yüz tanıma sistemi kullandığı düşünüldüğünde, yolcuların sadece bir tane fotoğrafı veri setinde bulunacaktır ve bu veri seti ile geleneksel derin sinir ağı eğitimi gerçekleştirilemeyecektir. Bu durumda, TSÖ tekniğinin kullanılması sorunu çözmektedir. Büyük veri setlerinin bulunmadığı ve sürekli değişen girdi-çıktı olması durumlarında kullanılmaktadır. Geleneksel KSA ağının kullanımının uygun olmadığı durumlarda TSÖ kullanımı önerilmektedir [113]. Sağlam bir sinir ağını eğitmek için küçük bir eğitim seti yeterli olmamaktadır. Küçük veri setlerinde, geleneksel KSA kullanımı uygun değildir çünkü eğitilmiş özellik vektörleri, gelecekteki görüntü tanıma için kullanılabilecek önemli bilgiler içermemektedir. Eğitilecek sınıfların veya veri kümelerinin sayısı arttığında, ağın yeniden eğitilmesi gerekmektedir, ancak bu işlem çok fazla zaman ve kaynak tükettiği için geleneksel derin sinir ağının kullanmak için uygun olmamaktadır. Bu nedenlerle TSÖ tekniğini kullanmak faydalı olmaktadır. TSÖ tekniğinin uygulanması için meta-öğrenme algoritmaları ile metrik öğrenme tekniklerinden faydalanmak gerekmektedir. Benzer nesnelerin sınıflandırıldığı bu çalışmada, her sınıfa ait tek örneğin bulunmasının sınıflandırma üzerindeki etkisi incelenmek istenmiştir.

2.5.2. Siyam Konvolüsyonel Sinir Ağı

Siyam sinir ağları, iki görüntünün ne kadar farklı olduğunu ölçmek için görüntüler arasındaki benzerliği hesaplamaktadır. Bu ağlar, iki özdeş sinir ağından oluşmaktadır. Ağda kullanılacak parametreler, sinir ağları tarafından ortak olarak paylaşılmaktadır. Yani parametreler her iki ağda da aynı değerdir [114]. Bu özellik, çok önemli bir avantaj sağlamaktadır. Siyam sinir ağları tarafından üretilen tahminler tutarlı olmaktadır. Yani aynı ağırlığa sahip ağlar olduğu için çok benzer görüntüler kullanıldığında, ağ sonucu bu iki görüntünün aynı sınıfa ait olduğunu göstermektedir. Benzer şekilde, farklı sınıflardan görüntüler verildiğinde, fark büyük olacaktır ve dolayısıyla bu iki görüntünün farklı sınıflara ait olduğu sonucuna varılacaktır [115,116].

Siyam sinir ağları, her iki girdinin de ağda üreteceği çıktılar arasındaki farkı dikkate almaktadır. Bu bilgiye dayanarak, bu iki görüntünün aynı veya farklı sınıflara ait olduğuna karar vermektedir. Sinir ağına verilen bir girdi, ağda katmanlardan geçirilerek öznitelikler elde edilerek çıktısı gösterilmektedir. Daha sonra ağa verilen ikinci girdi de aynı şekilde ağda katmanlardan geçirilerek öznitelikler elde edilerek çıktısı gösterilmektedir. Bu iki çıktı arasındaki fark değeri incelenmektedir. Elde edilen fark

değeri, verilen eşikten küçükse bu iki görüntü aynı sınıfa ait olduğu, büyükse bu iki görüntü farklı sınıflara ait kabul edilmektedir. Bu durum Şekil 2.23' te ifade edilmektedir.



Şekil 2.23. Siyam ağının örnek gösterimi

Siyam sinir ağının çalışma prensibi şu şekildedir: ilk olarak ağ iki adet girdi almaktadır ve bu girdiler, öznitelik vektörlerini elde etmek için aynı ağırlık değerleri olan aynı yapıdaki sinir ağından geçirilmektedir. Böylelikle elde edilen öznitelik vektörlerinin arasındaki fark değeri, herhangi bir uzaklık ölçüm fonksiyonu kullanılarak hesaplanmaktadır. Fark değeri belirtilen eşikten küçükse iki giriş benzer olduğu, aksi takdirde iki girdinin birbirinden farklı olduğu düşünülmektedir. Siyam sinir ağı, yapay sinir ağına benzer şekilde, çıktı vektörlerini hesaplamak için iki farklı girdi vektörüyle çalışmak için aynı ağırlıkları kullanmaktadır. Bu ağda, çıktı vektörlerinden biri, diğer çıktı vektörünün karşılaştırılacağı bir temel oluşturmak için önceden hesaplanmaktadır. Siyam sinir ağı, aldığı görüntüleri girdi olarak kodlayarak bir çıktı vektörü elde etmektedir. Benzerlik hesaplamasını bu vektör üzerinden yaptığından, ağ modeli bu çıktı vektörünü elde etmek için görüntüleri nasıl kodlayacağını öğrenmektedir.

Siyam ağlarını kullanmanın bir avantajı, sınıflandırma için her sınıftan yalnızca az miktarda veri kullanarak sınıf içi dengesizliği daha da azaltmasıdır. Diğer bir avantajı, ortalama iki ilişkili denetimli modelden daha iyi sınıflandırma sonuçları sağlayabilmesidir. Ayrıca Siyam ağları, anlamsal benzerliğin öğrenilmesinde de faydalıdır. Siyam ağlarının dezavantajları ise geleneksel ağlardan daha fazla eğitim süresi gerektirmeleri ve eğitimin ikili öğrenmeyi içermesi nedeniyle sınıfların olasılık değerinin nihayetinde üretilememesi, tahminin olasılıklarını vermeyerek her sınıfa olan mesafeyi vermesidir.

2.5.3. L1- Uzaklık Kaybı (L1-Distance Loss)

Tezin bu bölümündeki çalışmada; Siyam sinir ağı, girdi olarak aldığı görüntüleri $f(\theta)$ öznitelik fonksiyonu ile (bizim çalışmamızda, bu işlev KSA ağına karşılık gelmektedir.) kodlamayı ilk olarak öğrenmektedir. Daha sonra elde ettiği öznitelikler arasındaki uzaklık hesaplaması yapılmaktadır. Bizim çalışmamızda, Denklem 2.21 ile gösterilen L1-uzaklık fonksiyonu kullanılmıştır. Hesaplanan mesafe değeri, doğrusal bir ileri besleme katmanı ve sigmoid fonksiyonu tarafından bir olasılığa dönüştürülmektedir. Üretilen bu olasılık değeri Denklem 2.22' de ifade edilmiştir.

$$L1_{mesafe} = \left|f_\theta(x_i) - f_\theta(x_j)\right| \tag{2.21}$$

$$\mathcal{L}(B) = \sum_{(x_i,x_j,y_i,y_j\in B)} 1_{y_i=y_j} \log \mathrm{p}\left(x_i, x_j\right) + (1 - 1_{y_i=y_j})\log(1 - \mathrm{p}\left(x_i, x_j\right)) \tag{2.22}$$

Eğitim setindeki görüntüleri 'E' ve test veri setindeki örnek görüntü 'x' için ait olduğu sınıf Denklem 2.23 ile tahmin edilmektedir.

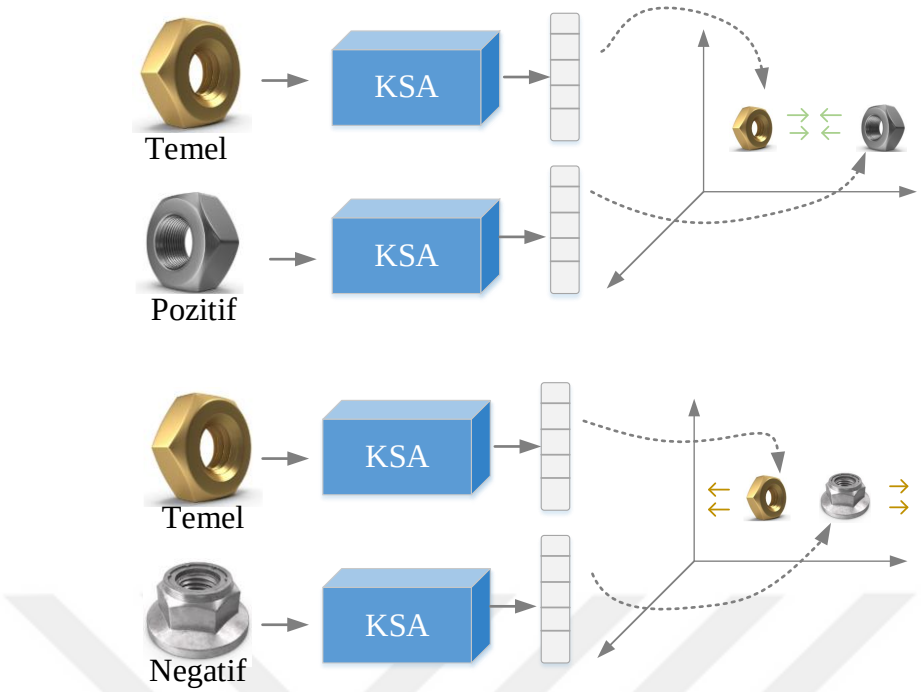
$$\widehat{c_E}(x) = c(\arg\max P(x, x_i)) \, , x_i \in E \tag{2.23}$$

2.5.4. Kontrast Kaybı (Contrastive Loss)

Geleneksel sinir ağlarında kayıp fonksiyonu, hata değerinin ulaşılabilir en küçük değere indirgenmesi için tasarlanmıştır. Tahmine dayalı hata fonksiyonlarının aksine, Kontrast kayıp fonksiyonu, mesafeye dayalı bir kayıp fonksiyonudur. Kontrast kaybı üç tane görüntüyü, üçlü kayıp fonksiyonu şeklinde karşılaştırmaktadır. Ancak kontrast kaybı, üçlü görüntüyü aynı anda karşılaştırmak yerine karşılaştırmaları birbiri ardına çiftler halinde gerçekleştirmektedir.

$$L = \frac{(1-Y)}{2} D_w^2 + \frac{Y}{2} max(0, marg - D_w)^2 \tag{2.24}$$

Girdi olarak verilen görüntüler aynı sınıfa aitse Denklem 2.24' teki Y değişkeni, 0 döndürürken farklıysa 1 döndürmektedir. İki çıktı vektörü arasındaki mesafe, D_w değişkeni ile gösterilir. İki görüntü arasındaki mesafe 0 ise kayıp fonksiyonu da 0 olacaktır ve bu durumda modelin eğitilebilmesi için '$marg$' değeri kayıp fonksiyonuna eklenmektedir. Girdi olarak verdiğimiz görüntüler aynı sınıfa aitse 1, farklı sınıflara aitse 0'dır. Hata fonksiyonları, görüntüler arasındaki mesafeye göre hesaplanır. Bu durum, Şekil 2.24' te temel görüntü, negatif görüntü ve pozitif görüntü olarak ifade edilmektedir.

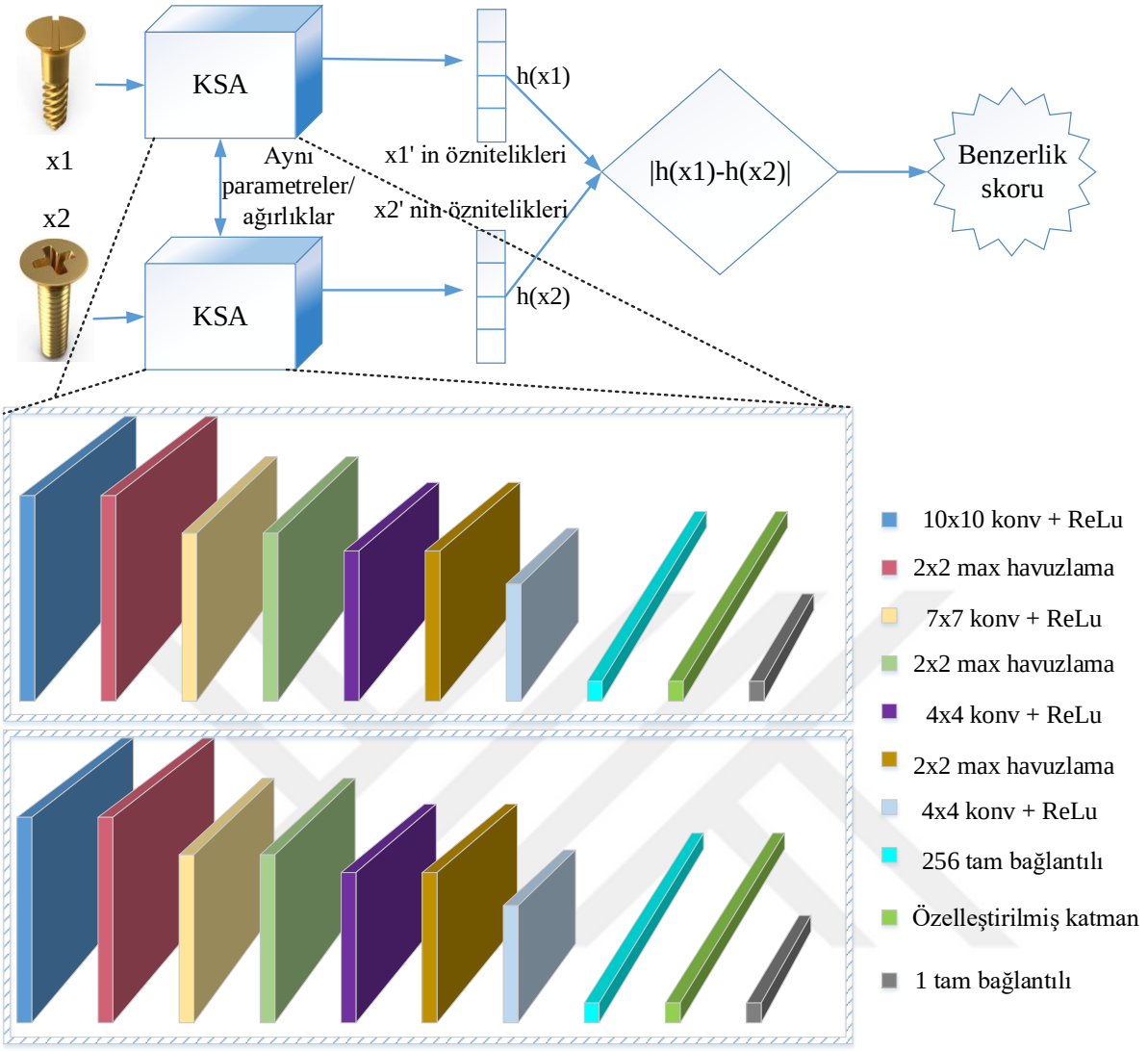


Şekil 2.24. Kontrast kaybı fonksiyonunun işleyişi

2.5.5. Önerilen Yöntem

Önerilen yaklaşımda, bağlantı elemanlarını sınıflandırmak için Siyam ağları, TSÖ ve derin öğrenme teknikleri kullanılmıştır. Önerilen çalışma 3-way, 5-way ve 10-way TSÖ üzerinden denenmiştir. TSÖ tekniği kullanılarak sınıflandırılacak sınıfların sayısı, N-way değişkendeki N-değeri ile tanımlanmaktadır. Sınıflandırılacak her sınıf için eğitimde kullanılan örnek sayısı birdir. Sınıflandırılacak sınıf sayısı arttıkça önerilen yöntemin başarı oranı düşmektedir. Deneysel çalışmalar sonucunda elde edilen doğruluk oranları karşılaştırmalı olarak Bulgular ve Tartışma bölümünde sunulmuştur. Bu çalışmada, Siyam ağının kayıp fonksiyonu olarak L1 mesafesi fonksiyonu ve Contrastive kayıp fonksiyonu kullanılmıştır.

Tezin bu bölümünde oluşturulan Siyam KSA ağında, giriş görüntüleri RGB formatında 105x105 piksel boyutundadır. Ağa aktarılan giriş görüntüsünü, iki boyutlu gri tonlamalı biçime dönüştürmek için birkaç ön işleme adımı kullanılmıştır, bu da ağdaki hesaplama maliyetini azaltmıştır. Önerilen ağın içeriğinde; 64 adet 10x10 konvolüsyon katmanı, 2x2 maksimum havuzlama katmanı, 128 adet 7x7 konvolüsyon katmanı, 2x2 maksimum havuzlama katmanı, 128 adet 4x4 konvolüsyon katmanı, 2x2 maksimum havuzlama katmanı, 256 adet 4x4 konvolüsyon katmanı, düzleştirme katmanı, 256 tam bağlantılı katman ve özelleştirilmiş bir katman bulunmaktadır. Bahsedilen katmanlar Şekil 2.25' te gösterilmiştir.



Şekil 2.25. Önerilen çalışmanın blok şeması

Bu çalışmada oluşturulan Siyam ağının performansı, 1-En Yakın Komşu ve Rastgele Tahmin gibi temel yöntemlerle de karşılaştırılmıştır. K-NN yöntemi ile üretilen sonuç, TSÖ yönteminin sonucu ile eşleştirildiği için k değeri 1 olarak seçilerek 1-En Yakın Komşu (1-NN) yöntemi kullanılmıştır. K-NN yönteminde kayıp fonksiyonunun hesaplanmasında, L2-mesafesi kullanılmıştır. Ağırlıklar, tüm eğitim görüntüleri doğru şekilde sınıflandırılana kadar rastgele tahmin tekniğinde rastgele başlatılmaktadır [117]. Test işlemindeki N-way değeri, rastgele tahmin yaklaşımının doğruluğuna bağlıdır.

Contrastive kayıp fonksiyonu ve L1-uzaklık fonksiyonu da eşit koşullar altında test edilmiştir ve en iyi sonuca L1 mesafe fonksiyonu ile ulaşıldığı fark edilmiştir. Siyam ağları ile geliştirilen KSA mimarisi Sigmoid, Softplus ve Tanh aktivasyon fonksiyonları ile test edilmiştir. 16 parti boyutu, 100 iterasyon, Adam optimizasyon algoritması olan hiper-parametreler ayarlanmıştır.

2.6. Bağlantı Elemanlarının ve Tekstil Makinesi Yedek Parçalarının Birkaç-Seferde Öğrenme Tabanlı İlişki Ağı ile Sınıflandırılması

Bu bölümün amacı, görsel olarak birbirine çok benzeyen nesne gruplarını sınıflandırmaktır. Derin metrik öğrenme (Deep Metric Learning-DMÖ) tekniği, nesne benzerliğinin hesaplanmasına yardımcı olmaktadır. Bu çalışmada, DMÖ ile konvolüsyon tabanlı bir yeni bir İlişki Ağ tasarlanmıştır. Geliştirilen modelin performansı, önerilen yöntem kullanılarak iki farklı gerçek dünya veri setinde değerlendirilmiştir. Çalışmamızda, *VS_Baglanti* ve *VS_Nit2* veri setlerimiz kullanılmıştır. Önerilen yöntem, görüntü özellikleri arasındaki ilişkinin geometrik ortalaması dikkate alınarak hesaplanmıştır. Bu çalışmada, birkaç-seferde öğrenme ile geometrik ortalamaya dayalı yeni bir ilişki ağı (GMRNet) tasarlanmıştır. Literatürdeki standart ilişki ağına, geometrik ortalama modülü eklenerek hem literatüre yeni bir mimari kazandırılmıştır hem de standart ilişki ağına kıyasla daha yüksek başarı elde edilmiştir. Tasarlanan bu ağda, N ve K değerleri sırasıyla sınıf sayısı ve sınıflardaki örnek sayısına karşılık gelmektedir. Bu çalışmada, N=3,5,7,10 ve K=5,10 kombinasyonları ile test edilmiştir.

2.6.1. Birkaç-Seferde Öğrenme

Az sayıda öğrenme olarak ta adlandırılan TSÖ tekniği, eğitim veri kümesinin sınırlı bilgi içerdiği bir tür makine öğrenimi yöntemidir. Makine öğrenimi uygulamaları için yaygın bir uygulama, modelin alabileceği kadar veri beslemesidir. Bunun nedeni, çoğu makine öğrenimi uygulamasında, daha iyi tahmin sağlamak üzere daha fazla veri ile model beslenmektedir. Bununla birlikte, birkaç-seferde öğrenme, daha az eğitim verisi ile doğru makine öğrenimi modelleri oluşturmayı amaçlamaktadır. Girdi verilerinin boyutu, zaman maliyeti ve hesaplama maliyeti gibi kaynak maliyetlerini belirleyen bir faktör olduğundan, son zamanlarda uygulamalarda, birkaç seferde öğrenme kullanılarak makine öğrenmesinin maliyetleri azaltılabilmektedir.

Birkaç-seferde öğrenme, bilgisayarların insanlar gibi birkaç örnekten öğrenmesi beklenmektedir. BSÖ ile makineler nadir durumları öğrenebilmektedirler. Örneğin, hayvanların görüntülerini sınıflandırırken BSÖ teknikleriyle eğitilmiş bir makine öğrenme modeli, az miktarda ön bilgiyi işledikten sonra nadir bir hayvan türünün görüntüsünü doğru şekilde sınıflandırabilmektedir. Ayrıca BSÖ ile bir modeli eğitmek için daha az veri gerektiğinden, veri toplama ve etiketleme ile ilgili yüksek maliyetler ortadan kalkmaktadır. Düşük miktardaki eğitim verisi, eğitim veri setinde hesaplama maliyetlerini önemli ölçüde azaltabilen düşük boyutluluk anlamına gelmektedir. Bilgisayarlı görü, doğal dil işleme, ses işleme, robotik, sağlık ve diğer alanlardaki uygulamalarda da BSÖ kullanılmaktadır.

Sıfır-seferde öğrenme ise eğitim veri setinde sınıflara ait herhangi bir örneğin bulunmadığı bir yöntemdir ve model doğru sınıfı bu şekilde tahmin etmeyi amaçlamaktadır. Tek-seferde öğrenme yönteminde, amaç tek bir eğitim örneğinden nesne kategorileri hakkında bilgi öğrenmektir ve. Akıllı telefonlar tarafından kullanılan yüz tanıma teknolojisi, tek-seferde öğrenme problemine bir örnektir.

Geleneksel denetimli öğrenme modeli ile BSÖ karşılaştırıldığında, geleneksel denetimli öğrenmede model eğitildikten sonra model tahmin yapmak üzere test görüntüsü sisteme verilmektedir. BSÖ' de ise sorgu örneği daha önce model tarafından hiç görülmemiş olup bilinmeyen bir sınıfa aittir. Bu durum iki teknik arasındaki temel farktır.

Makine öğrenmesi modeli, görülmemiş iki sınıfı birbirinden ayırt edebilmek için Siyam ağlarını ve Triplet ağlarını kullanabilmektedir. Eğitim veri kümesinde bulunmayan daha önce görülmemiş çoklu-sınıfların ayırt edilmesinde: Matching(Karşılaştırma) ağı, Prototypical(Prototipik) ağı ve Relation(İlişki) ağı kullanılaktadır.

BSÖ tabanlı sınıflandırmada, $V_{meta-egitim}$ eğitim veri setine $V_{meta-test}$ ise test veri setine karşılık gelmektedir. Her bir veri seti sınıflardan ve örneklerden oluşmaktadır. Ancak BSÖ tabanlı veri setlerinde her sınıfta en az 2 en fazla ise 10 adet görüntü bulunmaktadır. BSÖ tabanlı veri setleri N adet sınıfı ve her sınıfta K adet örneği içermektedir [118-120]. Bu duruma literatürde, N-way K-shot denilmektedir. Eğitim verisi NxK adet örnekle sınırlıdır. Bu sebeple eğitim aşaması zor olmaktadır. Her bir eğitim iterasyonunda $V_{meta-egitim}$' den N sınıf rastgele seçilmektedir. Rastgele olarak seçilen eğitim örnekleri her sınıftan alınmaktadır [121]. Eğitim veri seti Denklem 2.25 ile ifade edilmektedir. Sorgu verisi ise eğitim veri seti için rastgele seçilen örneklerin dışında kalan örnekler olan R grubundan rastgele olarak seçilmektedir [122, 123].

$$T = \{T_a^b; a = 1, \dots, N\ b = 1, \dots, K\} \tag{2.25}$$

$$Q = \{q_m, l_m; m = 1, \dots, R\} \tag{2.26}$$

Denklem 2.25' teki T_a^b değeri, a-sınıfındaki rastgele b. örnektir. Denklem 2.26' da q_m örneğinin etiketleri $l_m \in [1, \dots, N]$ aralığındadır.

2.6.2. Derin Metrik Öğrenme

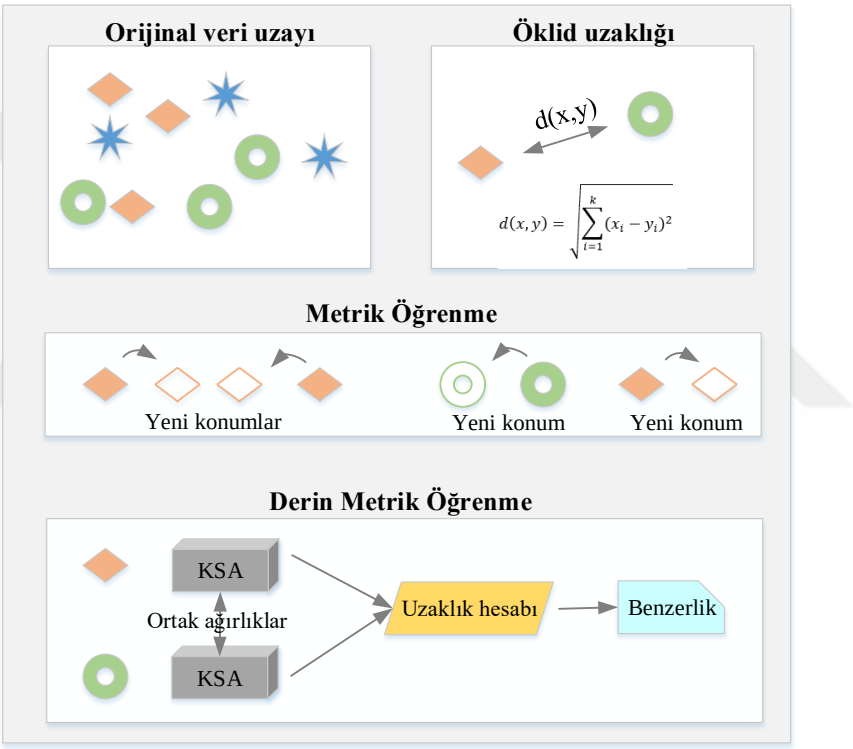
Makine öğrenmesi algoritmaları, verilerdeki hedef girişlere karşılık gelen çıkış etiketlerini bulmak için bir dizi kural veya karmaşık işlev kullanmaktadırlar. Metrik öğrenme ise veriler üzerinden benzerlik fonksiyonunu öğrenmeye çalışmaktadır [124-127]. Metrik öğrenme, aynı sınıfa karşılık gelen özellik vektörleri arasındaki mesafeyi azaltırken, farklı örneklere karşılık gelen özellik vektörleri arasındaki mesafeyi arttırmaktadır. Model, özellik vektörlerini bu şekilde öğrenmeyi amaçlamaktadır. Mesafe metriği, daha anlamlı ve güçlü bir ayırt edici model sağlamak için örnek benzerliğini kullanan yeni bir veri formatıdır.

Geleneksel makine öğrenimi yaklaşımlarının ham verileri işleme yetenekleri sınırlı olmaktadır. Ön-işleme veya özellik çıkarma için derin öğrenme gerekli değildir. Derin öğrenme, kendi kendine yüksek seviyeli özellikler öğrenmektedir. Euclid, Mahalanobis, Matusita, Bhattacharyya ve Kullback-Leibler mesafeleri, veri sınıflandırması için en yaygın kullanılan benzerlik ölçütleridir [128]. DMÖ, ham

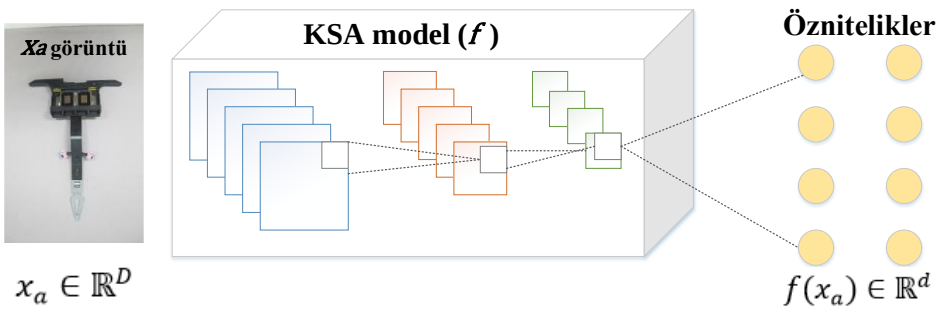
verilerden öğrenmek için derin mimariler kullanmaktadır ve doğrusal olmayan alt uzay öğrenimi yoluyla gömülü özelliklerle benzerlik bulmaktadır [129, 130]. DMÖ, örnekler arasındaki mesafe ile orantılıdır. Farklı nesneler arasındaki mesafeyi büyütürken benzer örnekler arasındaki mesafeyi azaltmaktadır. Bunu gerçekleştirmek için derin öğrenme metrik kayıp fonksiyonları kullanılmaktadır. DMÖ uygulamalarında sıklıkla kullanılan Mahalanobis uzaklık hesabı Denklem 2.27' de ifade edilmiştir.

$$MD = (x - m)^T C^{-1}(x - m) \tag{2.27}$$

Denklem 2.27' deki x veri vektörüdür, m bağımsız değişkenlerin ortalama değer vektörü, C^{-1} bağımsız değişkenlerin kovaryans matrisinin tersi ve T transpoze işlemini temsil etmektedir. DMÖ temel mantığı Şekil 2.26' da ve DMÖ tabanlı KSA yaklaşımı ise Şekil 2.27' de gösterilmiştir.



Şekil 2.26. Derin metrik öğrenme [128]



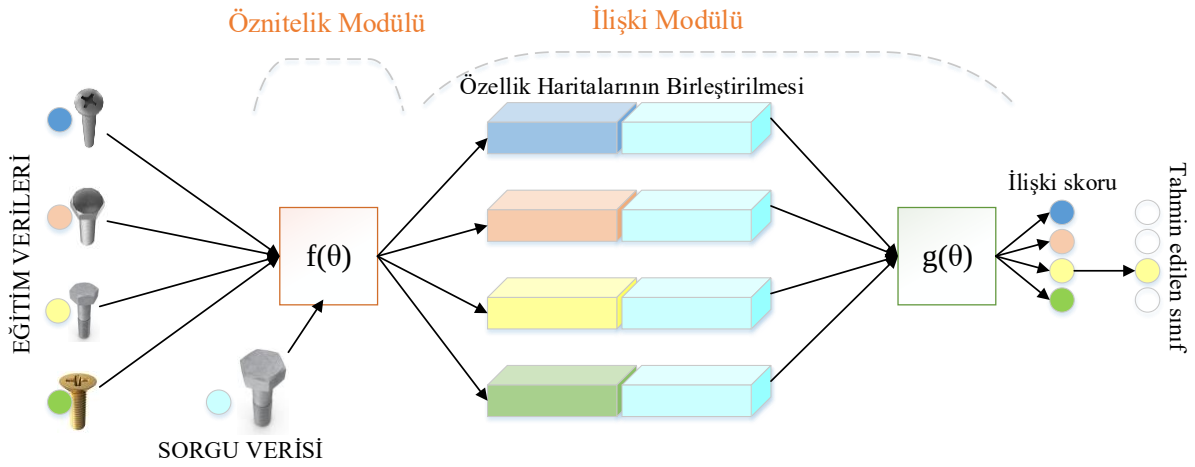
Şekil 2.27. DMÖ ve KSA tabanlı bir eğitim yaklaşımı

DMÖ, metrik kayıp fonksiyonuna odaklansa da, bilgilendirilmiş örnek seçimi de sınıflandırmaya önemli ölçüde katkıda bulunmaktadır. Ağın örneklerine katkıda bulunmak için ağa farklı eğitim örnekleri sağlanmalıdır. Sonuç olarak, ağın başarısını artırmak için DMÖ uygulanmadan önce ön işleme ve örnek seçimi yapılmalıdır.

2.6.3. İlişki Ağı

Geleneksel denetimli makine öğrenimi sisteminde, modelin parametrelerini eğitmek için büyük miktarda etiketlenmiş veri ve birçok eğitim yinelemesi gerekmektedir. Meta-öğrenme sürecinde, giriş görüntüleri arasındaki farkı ölçmek için derin metriğini öğrenme kullanılmaktadır. Görüntüler ve sınıf etiketleri arasındaki ilişkiyi karşılaştırmaktadır. Model eğitildikten sonra, İlişki ağı daha fazla güncellemeden, sorgu örneğinden gelen girdi ile eğitim kümesinden gelen girdi arasındaki ilişki/benzerlik puanını hesaplayarak yeni sınıfların görüntülerini sınıflandırma yeteneğindedir.

Şekil 2.28' de gösterildiği gibi, İlişki ağının genel yapısının içeriğinde: öznitelik modülü, ilişki modülü ve ilişki değerini hesaplayan modülü bulunmaktadır. İlişki modülü, her sınıf prototipi ile bir sorgu görüntüsünün özelliklerinin birleştirilmesiyle beslenmektedir ve her bir çift için bir ilişki puanı verilmektedir. İlişki puanlarına bir Softmax katmanı uygulayarak bir tahmin değeri elde edilmektedir. Uzaklık fonksiyonu, diğer meta-öğrenme algoritmalarından farklı olarak tanımlanmamaktadır; ancak algoritma tarafından bu uzaklık fonksiyonu öğrenilmektedir. İlişki ağının bunu yapan kendi ilişki modülü bulunmaktadır.

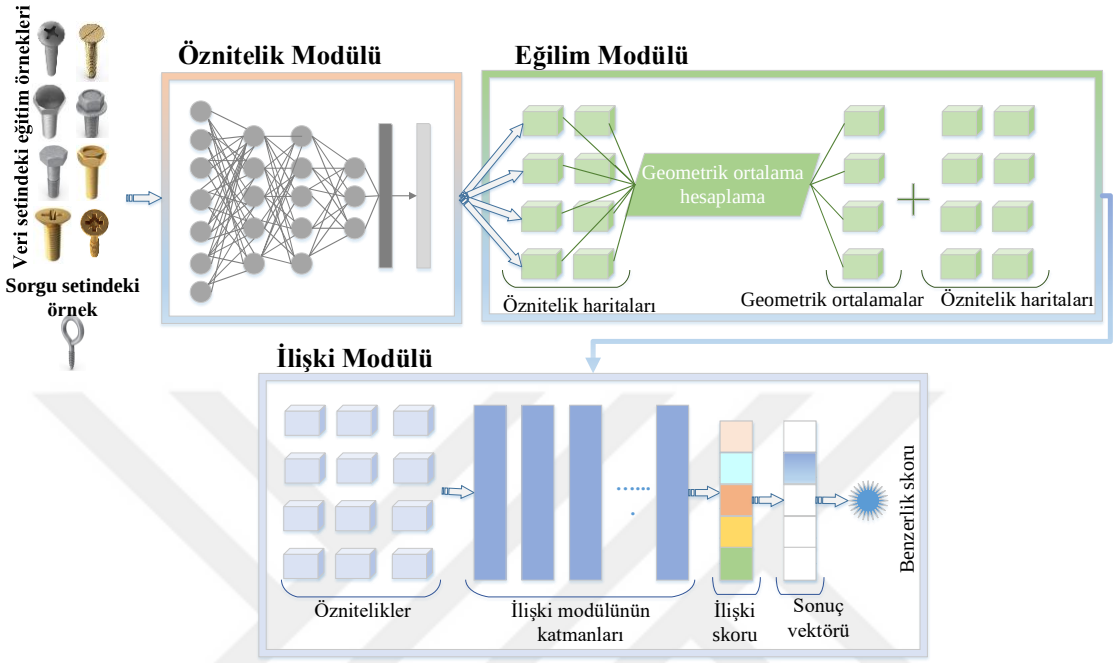


Şekil 2.28. Literatürdeki standart İlişki ağının genel yapısı

2.6.4. Önerilen Yöntem

Tezin bu bölümündeki çalışmanın amacı, iki farklı gerçek veri setini sınıflandırmaktır. BSÖ ve DMÖ ile yeni bir İlişki ağı (GMRNet) tasarlanmıştır. GMRNet, bir öznitelik modülü, bir eğilim modülü ve bir ilişki modülünden oluşmaktadır. Bu çalışmada, veri kümesindeki sınıf sayısı ve örnek sayısının

farklı seçeneklerle birleştirerek ağ performansına etkisi olduğu tespit edilmiştir. Elde edilen sonuçlar detaylı olarak Bulgular ve Tartışma bölümünde ifade edilmiştir. Öznitelik uzayındaki görüntü çiftleri arasındaki ilişki, KSA sınıflandırıcısı ile hesaplamıştır. GMRNet genel yapısı Şekil 2.29' da verilmiştir.



Şekil 2.29. Önerilen İlişki ağının genel yapısı

Tablo 2.10. GMRNet algoritmasının sözde kodu

Girdi: $E, \gamma, b, lr, M, f_\varphi(x), w_g, w_f, r_w, x, m$

Çıktı: $\acute{w_g}, \acute{w_d}, \acute{r_w}$

for $e = 0, 1, 2, \ldots, E-1$ **do**

  for $i = 0, 1, 2, \ldots, (\gamma/m) - 1$ **do**

    $f_\varphi(x) \leftarrow w_f \times x$ # öznitelik fonksiyonu

    $M_\varphi(x) \leftarrow \prod_i^m f_\varphi(x)^{1/m}$ # eğilim fonksiyonu

    $g_\varphi(x) \leftarrow w_g \times f_\varphi(x)$ # ilişki fonksiyonu

    Birleştirme $g_\varphi(x)$ ve $M_\varphi(x)$

    $S_{g,m} \leftarrow r_w(g_\varphi(x), M_\varphi(x))$ # benzerlik skoru

    $\mathcal{L} = CCE(S_{g,m}, x)$ # kategorik çapraz entropi kayıp fonksiyonu **hesaplama**

    $\acute{w_g}, \acute{w_d}, \acute{r_w} = Adam(\nabla w_f, w_d, r_w[\mathcal{L}], lr)$

  End for

End for

Tablo 2.10'da gösterildiği gibi çalışmamızda üretilen GMRNet algoritmasının sözde kodundaki E değeri eğitim iterasyonunu, γ değeri eğitim boyutunu, b değeri parti boyutunu, lr öğrenme oranını, M geometrik ortalamayı, $f_\varphi(x)$ özellik haritasını, w_g ve w_f tasarlanan ağ parametrelerini, r_w ilişki

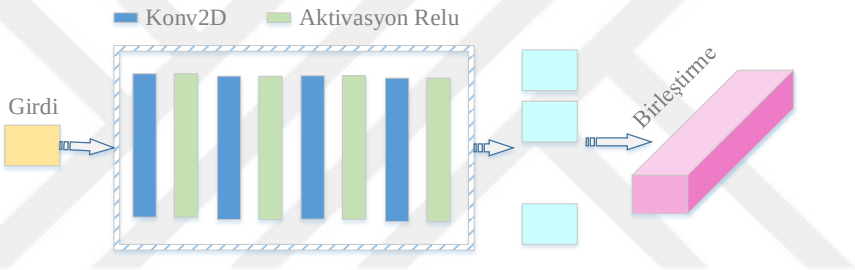
modülünü, x giriş görüntüsünü, m değeri ise x görüntüsünün eğitim örneğini oluşturmaktadır. Ayrıca $\acute{w_g}, \acute{w_d}, \acute{r_w}$ sırasıyla öznitelik, eğilim ve ilişki modüllerindeki optimize edilmiş eğitim parametrelerini temsil etmektedir.

Öznitelik Modülü

Bu modülün amacı, giriş verilerinden ayırt edici özellikleri çıkarmaktır [131, 132]. Bu modülün içeriği, konvolüsyon ve aktivasyon katmanlarından oluşmaktadır. Şekil 2.30' da öznitelik modülünün katman bilgileri verilmiştir. Verilen bir x görüntüsünün öznitelik haritasını, $f_\varphi(x)$ fonksiyonu üretmektedir. Tasarlanan ağdan alınan görüntüler işlenirken $g_\varphi(x)$ elde edilmektedir.

$$f_\varphi(x) = w_f \times x \tag{2.28}$$

$$g_\varphi(x) = w_g \times f_\varphi(x) \tag{2.29}$$



Şekil 2.30. GMRNet' teki öznitelik modülü

Eğilim Modülü

Bu modül, literatürdeki standart İlişki ağlarında bulunmamakla beraber bu çalışmanın özgün yönünü ifade etmektedir. Bu modülün eklenmesi, öznitelikler arasındaki ilişkinin eğilim yönünü belirlemektir. Q sorgu kümesinden x_q ve T eğitim kümesinden x_{ij} örnekleri, $f_\varphi(x)$ öznitelik çıkarma fonksiyonu ile elde edilmiştir. Elde edilen x_q ve x_{ij}' yi ilişki modülüne geçirmeden önce geometrik ortalama değerleri Denklem 2.30' daki gibi hesaplanmaktadır ve x_m değeri elde edilmektedir. Geometrik ortalamalar, girdi olarak kullanılan noktalar kümesindeki merkezi eğilimi temsil etmektedir. Sonuç olarak, iki girdi arasındaki ilişki belirlenmektedir. $g_\varphi(x)$ ilişki modülü x_m, x_q ve x_{ij} özniteliklerini giriş olarak almaktadır. Girişlere bağlı olarak 0-1 aralığında bir benzerlik skaler değeri üretilmektedir.

$$m_\varphi(x) = \prod_{i=1}^{k} f_\varphi(x)^{1/k},\ k = \text{görüntü boyutu} \tag{2.30}$$

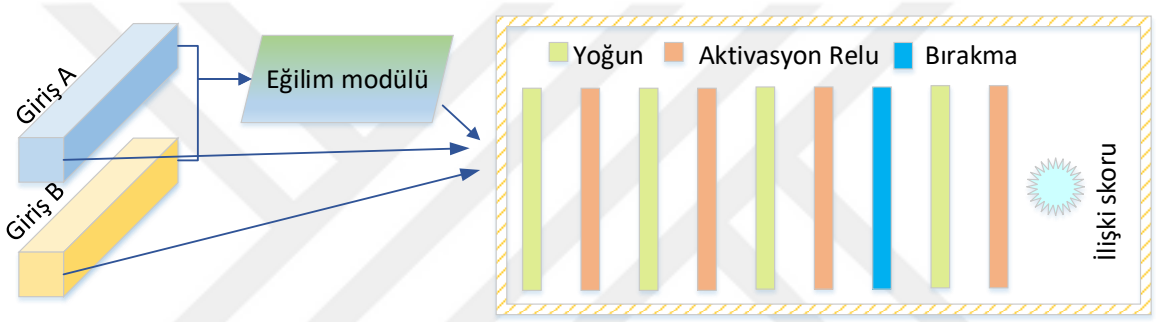
İlişki Modülü

Bu modülün amacı, örnekler arasındaki benzerlik derecesini hesaplamak ve bir korelasyon sonucu oluşturmaktır [133]. Bu modülde Yoğun (dense), Aktivasyon Relu ve Bırakma katmanları

kullanılmıştır. Bu katman bilgileri Şekil 2.31' de gösterilmiştir. Görsel öznitelikler ile geometrik ortalama özniteliği arasındaki benzerliği hesaplarken ilişki modülü, $r_w$değerini üretmektedir. Benzerlik puanı ne kadar yüksekse karşılaştırılan nesneler o kadar benzerdir; aksi takdirde, nesneler farklıdır. Öznitelik modülündeki $g_\varphi(x)$ ve geometrik ortalama sonucundaki $m_\varphi(x)$ birleştirildikten sonra $s_{g,m}$ benzerlik puanı oluşturulmaktadır. Denklem 2.31' deki $C(,)$ değeri, birleştirme fonksiyonunu simgelemektedir. Çalışmamızda kullandığımız kayıp fonksiyonu olarak kategorik çapraz entropi kullanılmıştır ve bu fonksiyon Denklem 2.32' de gösterilmektedir.

$$s_{g,m} = r_w(C(g_\varphi(x_t), m_\varphi(x_t))), t = 1,2,\dots,k \tag{2.31}$$

$$CCE = \sum_i^c t_i \, log\left(s_{g,m}(x)_i\right), t_i = \text{hedef vektör} \tag{2.32}$$



Şekil 2.31. GMRNet' teki ilşiki modülü

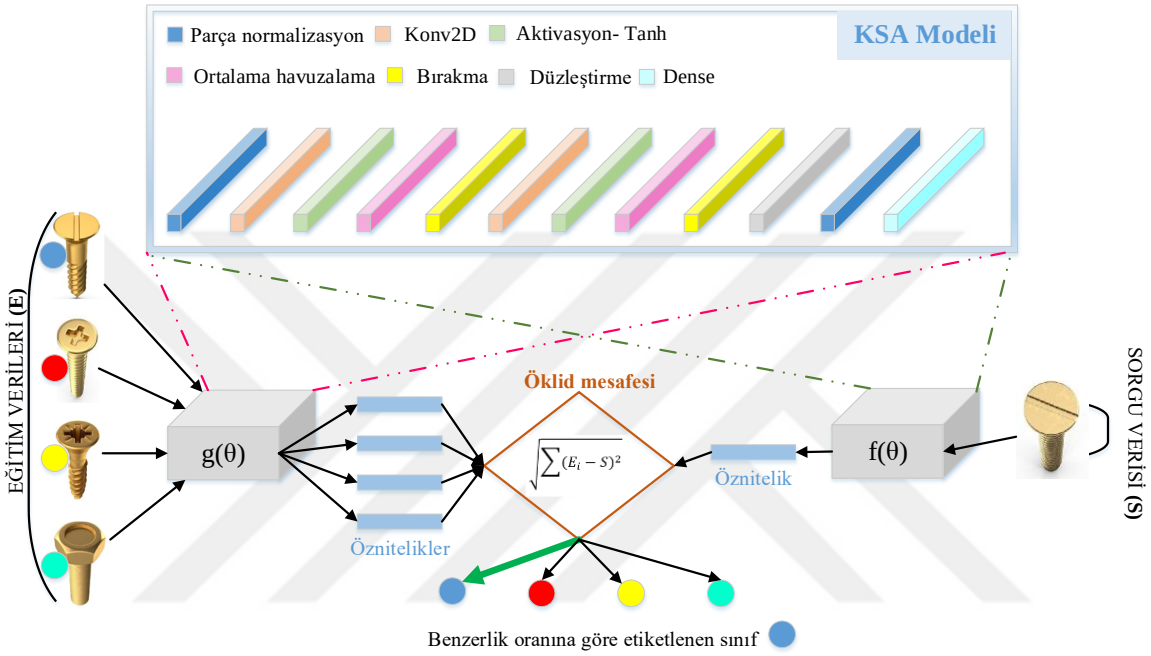
2.7. Bağlantı Elemanlarının Karşılaştırma Ağı ve Prototipik Ağı ile Birkaç-Seferde Öğrenme Tabanlı Sınıflandırılması

Benzerlik ölçütü tabanlı görüntü sınıflandırması yapıldığında, meta-öğrenme ve metrik öğrenme algoritmaları kullanılmaktadır. İyi bir makine öğrenmesi modeli, genellikle çok sayıda örnekle eğitim gerektirmektedir. İnsanlar ise aksine yeni kavramları ve becerileri çok daha hızlı ve verimli bir şekilde öğrenmektedir. Birkaç eğitim örneğiyle yeni kavramları ve becerileri hızlı bir şekilde öğrenerek benzer özelliklere sahip bir makine öğrenimi geliştirmek üzere meta-öğrenme algoritmaları tasarlanmaktadır. Az sayıda eğitim örneği ile görüntü sınıflandırması yapılırken kullanılan algoritmalar: Model-Agnostik Meta-Öğrenme, Karşılaştırma ağı, Prototipik ağı ve İlişki ağıdır.

Tezin bu bölümündeki çalışmada, az sayıda örnek bulunduran bağlantı elemanı veri setinin sınıflandırılması için Karşılaştırma ağı ve Prototipik ağı tasarlanmıştır. Sınıf sayısı N=3,5,7 ve 10 her sınıftaki örnek sayısı K=5 ve 10 farklı kombinasyonlarla olarak denenmiştir. Her iki ağın, benzer nesne sınıflandırma üzerindeki performansı ve sınıf sayısı ile sınıflardaki örnek sayısının performansa etkisi incelenmiştir.

2.7.1. Önerilen Karşılaştırma Ağ Modeli

Karşılaştırma ağının, herhangi bir ön-sınıf bilgisi olmadan örnekleri sınıflandırma yeteneği bulunmaktadır. Eğitim setindeki sınıflarda bulunan her örnekle sorgu görüntüsü karşılaştırılmaktadır. Sınıflar arasında ayrım yapabilmek için görüntülerin özellikleri hesaplanmaktadır. Ağ, her sınıfa ait bu özellikleri öğrenmektedir. Az sayıda bağlantı elemanı sınıflandırması için bu çalışmada tasarlanan Karşılaştırma konvolüsyonel sinir ağının mimarisi Şekil 2.32' de anlatılmıştır.



Şekil 2.32. Önerilen Karşılaştırma ağı

Eğitim veri seti E, girdi x ve etiketlenen örnekler l olduğu durumda: yeni örnek $\acute{x}$' in hangi sınıfa ait olduğu olasılık değeri $P(\acute{y} \mid \acute{x}, E)$ olmaktadır. Girdi $\acute{x}$ iken Karşılaştırma ağı çıktı etiketini tahmin etmek üzere Denklem 2.33 kullanılmaktadır.

$$\acute{y} = \sum_{i=1}^{l} \propto (\acute{x}, x_i) y_i \tag{2.33}$$

Denklem 2.33' teki $\propto$ değeri, $\acute{x}$ girdisinin x' e ne kadar benzediğini hesaplamaktadır. Benzerlik hesabında, kosinüs uzaklığı temel alınmıştır. Denklem 2.34' te kosinüs uzaklığı ile Softmax değerinin hesabı formül edilmiştir.

$$\propto (\acute{x}, x_i) = e^{cosine(f(\acute{x}), g(x_i))} \Big/ \sum_{j=1}^{l} e^{cosine(f(\acute{x}), g(x_j))} \tag{2.34}$$

Denklem 2.34' teki g fonksiyonu, E veri setinden öznitelik örnekler için E' nin tüm örnekleri hesaba dâhil etmektedir. Karşılaştırma ağında, eğitim setindeki her örnek sorgu görüntüsü ile bir çift

haline getirilerek karşılaştırılmaktadır. Tüm örnekler ile sorgu görüntüsü karşılaştırıldıktan sonra sorgu görüntüsünün hangi sınıfa ait olduğuna karar verilmektedir.

2.7.2. Önerilen Prototipik Ağ Modeli

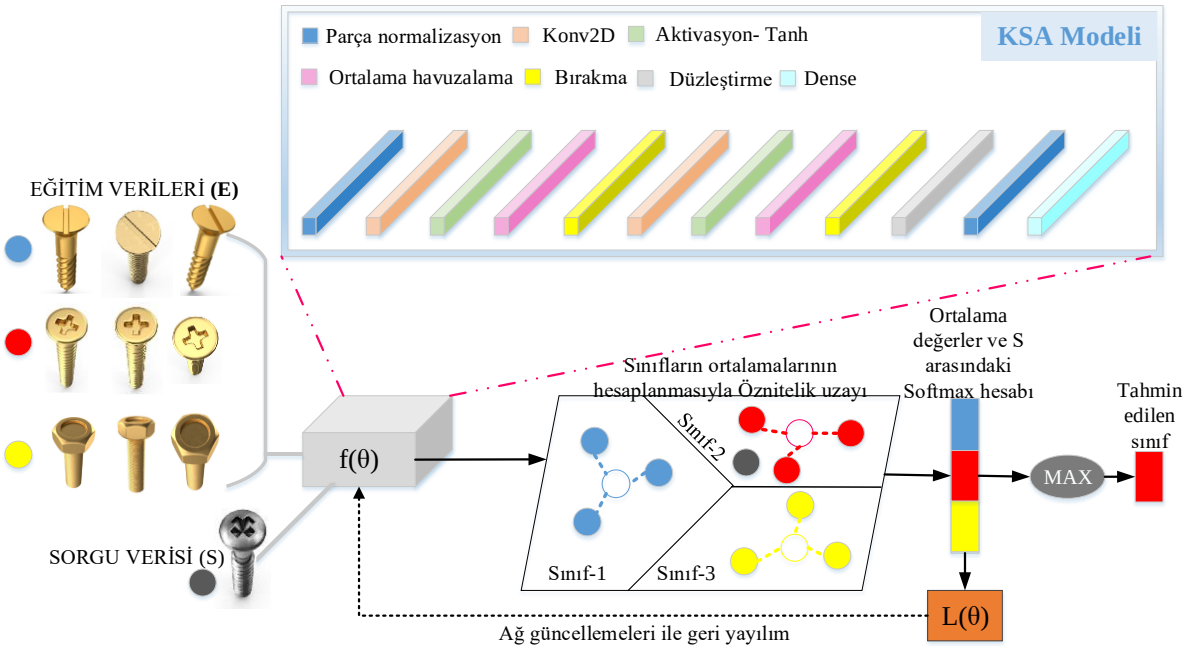
Prototipik ağlar, Karşılaştırma ağlara benzemektedir; ancak Prototipik ağ, sorgu görüntüsünü eğitim veri setinden gelen tüm görüntülerle karşılaştırmaz, sadece sınıf prototipleri oluşturularak bunlarla karşılaştırılırlar. Prototipik ağlarda sınıf prototipi, sınıftaki görüntülerin özelliklerinin ortalama değeridir. Prototipik ağlar, Karşılaştırma ağlardan farklı olarak kosinüs mesafesi yerine öklid mesafesini kullanmaktadır. Önerilen Prototipik ağ modeli Şekil 2.33' te gösterilmiştir.

D boyutlu özellik uzayında, her bir girdiyi kodlayarak özniteliklerini çıkaran fonksiyon $f(\theta)$' dır. Her c sınıfı için oluşturulan prototip, bu sınıftaki örneklerin ortalama vektörüdür.

$$v_c = \frac{1}{|E_c|} \sum_{(x_i, y_i) \in E_c} f_\theta(x_i) \tag{2.35}$$

Verilen bir sorgu görüntüsü q için prototip vektörler arasındaki Softmax değeri, Denklem 2.36' daki gibi hesaplanmaktadır.

$$P(y = c \mid x) = \exp(-d_\vartheta(f_\theta(x); v_c)) \Big/ \sum_{\acute{c} \in C} \exp(-d_\vartheta(f_\theta(x); v_{\acute{c}})) \tag{2.35}$$



Şekil 2.33. Önerilen Prototipik ağı

3. BULGULAR VE TARTIŞMA

3.1. Performans Ölçüm Metrikleri

Tez kapsamında, temelde KSA kullanılarak altı farklı çalışma gerçekleştirilmiştir. Bu çalışmaların performans çıktıları incelenirken birkaç metrik üzerinde durulmuştur. Bunlar; modelin doğruluğu, kesinliği, hassasiyeti, f1-ölçüm, karmaşıklık matrisi ve roc eğrisi altında kalan alan (Area Under Curve-AUC) değeridir. Bahsedilen metrikler, nesne sınıflandırma konusunda literatürde de sıklıkla esas alınan ölçütlerdir.

3.1.1. Karmaşıklık Matrisi

Gerçek değerleri bilinen bir modelin ürettiği tahmin çıktıları ile bu gerçek değerlerin karşılaştırılmasına dayanan bir performans ölçüm tekniğidir. Tahmin edilen yanlış veri sayısına karşılık veriyi doğru tahmin etme sayısının incelenmesidir. Literatürde, sınıflandırma tabanlı model performansının incelenmesinde en sık kullanılan yöntemdir. Sınıf sayısının karesi kadar bir boyutta olması beklenmektedir. Her sınıfa ait doğru tahmin ve yanlış tahmin değerleri, bu matriste ifade edilmektedir. Karmaşıklık matrisinde hesaplanan oran değerleri Şekil 3.1' de görselleştirilmiştir.



Şekil 3.1. Karmaşıklık matrisinin genel bilgisi

3.1.2. Doğruluk

Doğruluk, belirlenen makine öğrenmesi yaklaşımının bütün veri seti içerisinde kaç adet doğru tahmin yapıldığına ilişkin yüzdelik şekilde üretilen bir değerdir. Modelin performansının ölçümü için kullanılan temel bir ölçüm yöntemidir. Ancak sınıflandırma performansı incelenirken sadece doğruluk değerinin esas alınması yeterli değildir. Bu sebeple diğer ölçüm metrikleri de performans incelenirken göz önünde bulundurulmalıdır. Doğruluk hesaplamasında uygulanan işlem, Denklem 5.1' de ifade edilmektedir.

$$Dogruluk = \sum(TP + TN) \Big/ \sum(TP + TN + FP + FN) \tag{5.1}$$

3.1.3. Hassasiyet

Modelin pozitif olarak tahmin edilmesi gereken elemanlardan kaç tanesinin gerçekten doğru olarak tahmin etmesinin incelenmesidir. Tüm sınıflardan yapılan tahminlerin kaç tanesinin doğru olduğunun ölçülmesini ifade etmektedir. Bu değerin olabildiğince yüksek olması istenmektedir. Hassasiyet hesaplamalarında kullanılan formül Denklem 5.2' de verilmiştir.

$$Hassasiyet = {\sum(TP)}\Big/{\sum(TP + FN)} \tag{5.2}$$

3.1.4. Kesinlik

Modelin pozitif olarak tahmin ettiklerinin doğruluk performansını ölçmektedir. 'Pozitif tahmin edici' olarak ta ifade edilmektedir. Modelin pozitif olarak tahmin ettikleri arasında hepsinin pozitif olduğundan emin olma işlemidir. Modelin pozitif olarak tahmin ettiği bir durumdaki başarıyı hesaplamaktır. Modelin bir veriyi yanlış pozitif olarak etiketlemesinin, maliyeti yükselteceği durumlarda kesinlik değeri oldukça önemli olmaktadır.

$$Kesinlik = {\sum(TP)}\Big/{\sum(TP + FP)} \tag{5.3}$$

3.1.5. F1-Ölçüm

Modelin hassasiyetinin ve kesinliğinin tek bir ölçüm metriğinde birleştirilmesidir. Kesinlik ve hassasiyetin harmonik ortalamasının hesaplanması ile elde edilmektedir. F1-ölçüm hesabı yapılırken basit bir ortalama yerine harmonik ortalama kullanılmasının nedeni: uç durumların göz-ardı edilmesini engellemektir. Eğer basit bir ortalama ile bu ölçüm değeri hesaplansaydı; kesinlik değeri 1 ve hassasiyet değeri 0 olan bir modelin F1-ölçüm 0.5 olarak elde edileceğinden ve bu durum yanıltıcı sonuçlara ulaştırabilmektedir. F1-ölçüm değerinin kullanıldığı en önemli durum, eşit dağılmayan veri setlerindeki hatalı bir model seçimi yapmamaktır. Ayrıca sadece yanlış negatifi ya da yanlış pozitifi değil tüm hata maliyetlerini de içerecek bir ölçme metriğine ihtiyaç duyulması durumlarında, F1-ölçüm önemli bir metrik olmaktadır.

$$F1_\ddot{o}l\text{c}\ddot{u}m = 2 * {(kesinlik * hassasiyet)}\Big/{(kesinlik + hassasiyet)} \tag{5.4}$$

3.1.6. ROC Eğrisi ve Eğri Altında Kalan Alan(AUC)

Sınıflandırma problemlerinde oldukça sık kullanılan, önemli bir performans ölçümüdür. Olasılık eğrisi olmasının yanı sıra eğri altında kalan alan miktarı arttıkça sınıflandırıcının performansı artmaktadır. Yani eğri altında kalan alan miktarı ile sınıflar arası ayrıt etme performansı doğru

orantılıdır. ROC eğrisinde, y-ekseni doğru pozitif oranını temsil edip x-ekseni yanlış pozitif oranını temsil etmektedir.

$$TP_Orani = {TP}/{TP + FN} \tag{5.5}$$

$$FP_Orani = {FP}/{FP + TN} \tag{5.5}$$

AUC değeri, sınıfların ne kadar başarılı ayırt edilebildiğini göstermektedir. Dengeli dağılım göstermeyen veri kümelerinde modelin başarısı sadece doğruluk metriği ile ölçmek yetersiz kalmaktadır ve kesinlik ile hassasiyet metriklerini de göz önünde bulundurmak faydalı olmaktadır. Ancak kesinlik değeri arttıkça hassasiyet değerinin azaldığı fark edilmektedir. Bu yüzden F1-ölçüm metriği sınıflandırma modelimizin gerçek başarısını göstermektedir. ROC eğrisi de F1 değeri arttıkça eğri altında kalan alanın arttığını göstermiştir.

3.2. Pantograf Yüzeyindeki Arızaların Derin Öğrenme Tabanlı Sınıflandırılmayla Elde Edilen Sonuçlar

Tez kapsamında gerçekleştirilen ilk çalışmada, pantograf yüzeyindeki kömür arızalarının sınıflandırılması yapılmıştır. Kullanılan pantograf yüzey aşınma tipleri 4 tiptir: sağlıklı, hafif aşınma, yüksek aşınma ve oluk şeklinde aşınmadır. Bu aşınma tipleri çalışmamızda sırasıyla Sınıf1, Sınıf2, Sınıf3 ve Sınıf4 olarak adlandırılmıştır. '*VS_Pant*' veri setinde veri arttırma tekniği uygulanmadan önce Sınıf1, Sınıf2, Sınıf3 ve Sınıf4' teki görüntü sayısı sırasıyla 131, 155, 329, 156'dır. Veri arttırma tekniği uygulandıktan sonra veri setinde Sınıf1, Sınıf2, Sınıf3 ve Sınıf4 sınıflarına ait görüntü sayıları sırasıyla 1282, 1156, 1330 ve 1307 olmuştur. Bu çalışmada geliştirilen sınıflandırıcının performansını artırmak için veri artttırma tekniği ile toplam aşınma görüntüsü sayısı 701 iken 5375 yapılmıştır. Ham veri seti, sadece görüntü ön-işleme uygulanmış veri seti, sadece veri arttırma uygulanmış veri seti ve hem görüntü ön-işleme hem de veri arttırma uygulanan veri seti bu çalışmada sırasıyla Durum1, Durum2, Durum3 ve Durum4 olarak anılmaktadır. Bahsedilen durumların her biri, geliştirilen KSA ağında eğitilerek elde edilen performans çıktıları karşılaştırmalı olarak sunulmuştur. Bu çalışmada geliştirilen KSA mimarisine ait parametre bilgileri Tablo 3.1' de gösterilmiştir. Bu çalışmanın ana katkıları şu şekildedir:

- İlk katkı, önerilen çalışma literatür ile karşılaştırıldığında, [30]' daki çalışmada ve bizim çalışmamızda görüntülere ön-işleme adımları uygulanmasına rağmen bizim çalışmamızın başarı oranı %4 daha yüksektir. Çalışmamızda diğer çalışmalardan farklı olarak adaptif histogram eşitleme uygulanarak görüntülerin farklı ışıklandırma koşulları ortadan kaldırılarak performansı arttırılmıştır.
- İkinci olarak, [134]' teki çalışmada, görüntülere kenar çıkarma ve Hough dönüşümü uygulanarak pantograf kayan şerit tespit edilmiştir. Daha sonra geliştirilen KSA modeli ile eğitim tamamlanmıştır. [134]' teki veri setinde 771 tane eğitim görüntüsü ve 138 tane de test

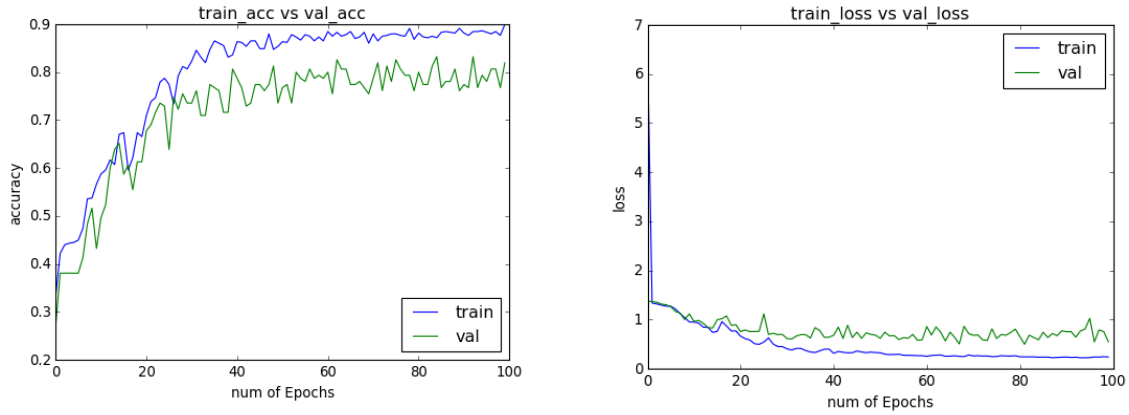
görüntüsü bulunmaktadır. Eğitim ve test görüntüsü sayısı açısından bizim çalışmamız daha öndedir: sırasıyla 4300 ve 1075'tir. Bunlara bağlı olarak [134]' ün başarı oranı %70,14'tür. Çalışmamızın başarı oranı %94,19' dur. Görüntü ön-işleme tekniklerine ek olarak veri iyileştirme tekniklerinin uygulanmasının da performansa katkı sağladığı görülmektedir.

- Ayrıca [135]' te ve bizim çalışmamızda derin öğrenme teknolojisi ile sınıflandırma aşamasında parametre optimizasyonu yapılmaktadır. Ancak bizim çalışmamızdaki doğruluk oranı %94,19 iken [135]' teki çalışmanın doğruluk oranı %90,62'dir.
- Çalışmamızın katkılarından biri de farklı senaryolar altında geliştirilen uygulama denenerek optimum sonucu veren durumun belirlenmiş olmasıdır. Uyarlanabilir bir uygulama olması sayesinde çalışmamız istenilen diğer veri setleri üzerinde uygulanabilir.

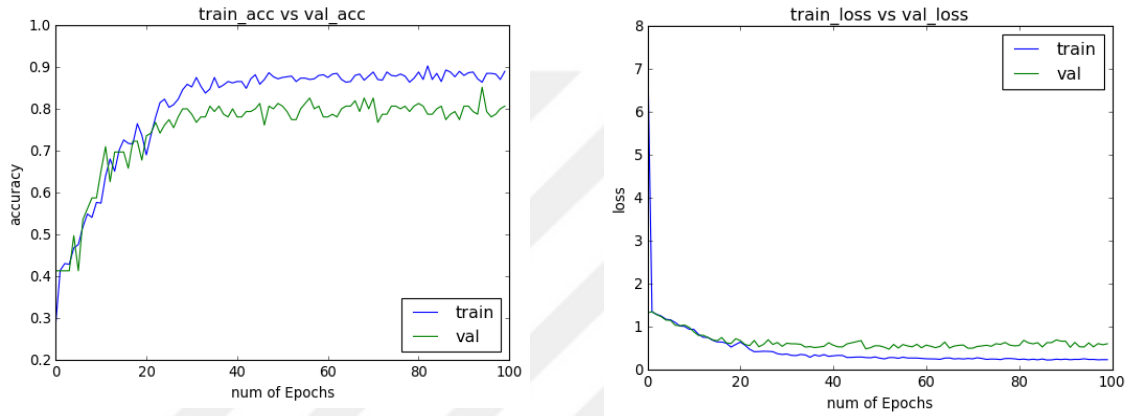
Her duruma ait doğruluk oranı sonuçları Şekil 3.2 ile Şekil 3.5 arasında ifade edilirken performans raporları Tablo 3.2 ile Tablo 3.6 arasında ifade edilmiştir.

Tablo 3.1. Geliştirilen KSA mimarisinin parametreleri

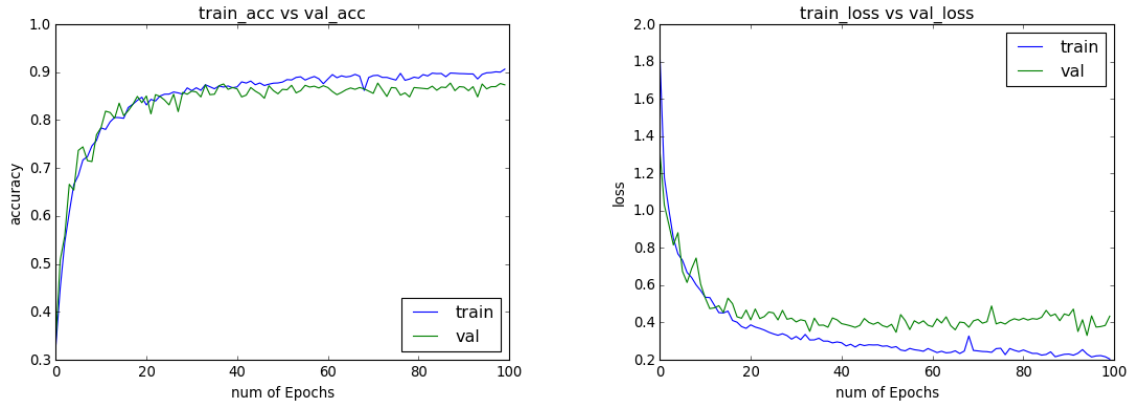
Katman	**Çıktı boyutu**
Giriş görütüsü	150, 150
Konvolüsyon-1 (32 filtre, 3x3 boyutlu)	148, 148, 32
Max Havuzlama-1 (32 filtre, 2x2 boyutlu)	74, 74, 32
Konvolüsyon -2 (64 filtre, 3x3 boyutlu)	72, 72, 64
Max Havuzlama -2 (64 filtre, 2x2 boyutlu)	36, 36, 64
Konvolüsyon -3 (64 filtre, 3x3 boyutlu)	34, 34, 64
Max Havuzlama -3 (64 filtre, 2x2 boyutlu)	17, 17, 64
Bırakma	0.25
Konvolüsyon -4 (128 filters of 3x3 size)	15, 15, 128
Max Havuzlama -4 (128 filters of 2x2 size)	7, 7, 128
Bırakma	0.25
Konvolüsyon -5 (256 filters of 3x3 size)	5, 5, 256
Max Havuzlama -5 (256 filters of 2x2 size)	2, 2, 256
Bırakma	0.25
Tam bağlantılı-1	1x262400
Tam bağlantılı -2	1x65792
Tam bağlantılı -3	1028
Softmax	num=4
İterasyon	100
Parti boyutu	64
Optimizasyon algoritması	Adam



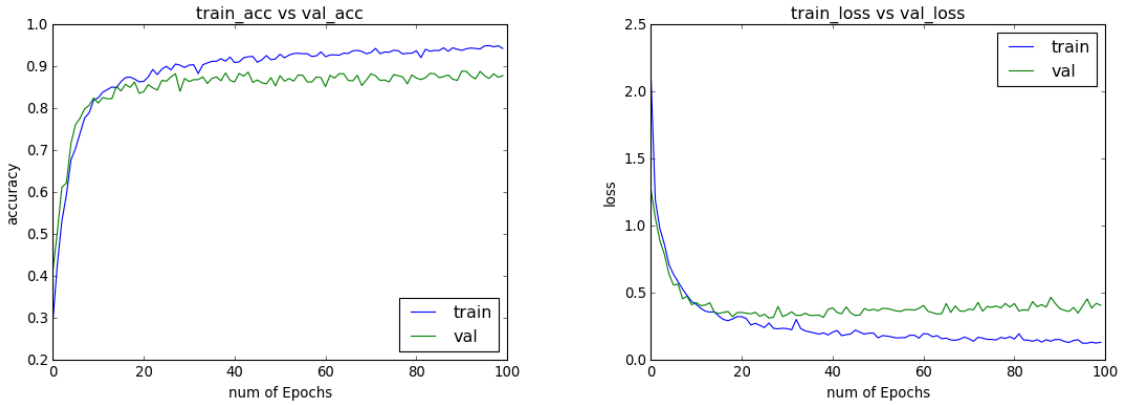
Şekil 3.2. Durum1' de eğitim ve doğrulama sonucunda elde edilen doğruluk ve hata oranları



Şekil 3.3. Durum2' de eğitim ve doğrulama sonucunda elde edilen doğruluk ve hata oranları



Şekil 3.4. Durum3' te eğitim ve doğrulama sonucunda elde edilen doğruluk ve hata oranları



Şekil 3.5. Durum4' 4e eğitim ve doğrulama sonucunda elde edilen doğruluk ve hata oranları

Tablo 3.2. Bu çalışmadaki Durum1' in performans raporu

Etiket	Kesinlik	Hassasiyet	F1-Ölçüm	Örnek sayısı
Sınıf1	0.65	0.88	0.75	25
Sınıf2	0.94	1.00	0.97	29
Sınıf3	0.86	0.81	0.83	59
Sınıf4	0.82	0.67	0.74	42

Tablo 3.3. Bu çalışmadaki Durum2' nin performans raporu

Etiket	Kesinlik	Hassasiyet	F1-Ölçüm	Örnek sayısı
Sınıf1	0.71	0.92	0.80	26
Sınıf2	0.97	0.97	0.97	37
Sınıf3	0.90	0.69	0.78	64
Sınıf4	0.60	0.75	0.67	28

Tablo 3.4. Bu çalışmadaki Durum3' ün performans raporu

Etiket	Kesinlik	Hassasiyet	F1-Ölçüm	Örnek sayısı
Sınıf1	0.84	0.84	0.84	276
Sınıf2	0.95	1.00	0.97	274
Sınıf3	0.97	0.77	0.86	279
Sınıf4	0.76	0.88	0.82	246

Tablo 3.5. Bu çalışmadaki Durum4' ün performans raporu

Etiket	Kesinlik	Hassasiyet	F1-Ölçüm	Örnek sayısı
Sınıf1	0.82	0.91	0.86	257
Sınıf2	0.96	0.99	0.97	254
Sınıf3	0.92	0.79	0.85	306
Sınıf4	0.81	0.83	0.82	258

Tablo 3.6. Bu çalışmadaki tüm durumların performans raporu

Veri seti	Eğitim doğruluğu	Eğitim kaybı	Doğrulama doğruluğu	Doğrulama kaybı
Durum1	0.89	0.23	0.81	0.54
Durum2	0.88	0.23	0.80	0.60
Durum3	0.90	0.20	0.83	0.38
Durum4	**0.94**	0.13	**0.87**	0.40

Bu çalışma, KSA tabanlı bir model ve görüntü ön-işleme teknikleri kullanılarak pantograf temas noktasındaki arızaların sınıflandırılmasına odaklanmaktadır. Çalışmada kullanılan pantograf görüntülerine, gri tonlama dönüşümü uygulanmıştır. Böylece çalışmada geliştirilen derin öğrenme modelindeki parametre sayısı azaltılmıştır. Elde edilen görüntülere, normalizasyon uygulanarak görüntüler belirli bir boyuta ölçeklendirilmiştir. Daha sonra, görüntülerin farklı pozlanma ve aydınlatma koşullarını yok etmek için görüntülere histogram eşitleme yöntemi uygulanmıştır. Bu sürece ek olarak, veri arttırma tekniğinin uygulanması ile önerilen sınıflandırıcının performansı arttırılmıştır. Gerçekleştirilen deneysel sonuçlar incelendiğinde, görüntü ön-işleme yöntemlerinin ve veri arttırmanın sınıflandırıcı performansına olumlu katkı sağladığı sonucuna varılmıştır. Önerilen KSA tabanlı modelin doğruluk oranları: ham veri seti üzerinde %89, sadece görüntü ön-işleme tekniği uygulandığında %88, sadece veri arttırma uygulandığında %90 ve son olarak hem ön-işleme hem de veri arttırma uygulandığında %94 olarak elde edilmiştir. Doğruluk oranlarına bakıldığında, sınıflandırıcının en iyi performansı hem-ön işleme hem de veri artırmanın uygulandığı veri setinden, ikinci en iyi performansı sadece veri arttırma uygulanan veri setinden elde etmiştir.

3.3. Bağlantı Elemanlarının Transfer Öğrenme Tabanlı İnce-Ayar ve Özellik Çıkarım ile Sınıflandırma Sonuçları

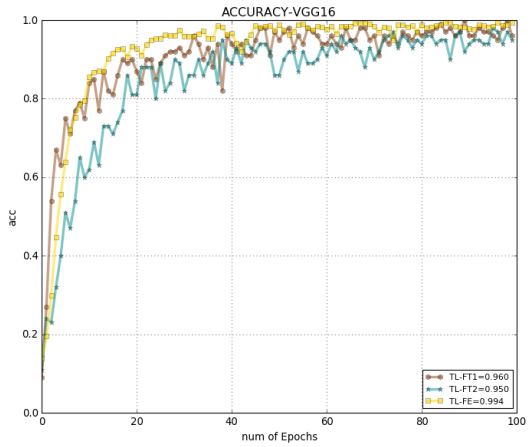
Tez kapsamında gerçekleştirilen ikinci çalışmada, 18 çeşit bağlantı elemanını sınıflandırmak için önceden-eğitilmiş ağları kullanan farklı senaryolar önerilmiştir. Bu çalışmada, VGGNet, DenseNet, EfficientNet, MobileNet, InceptionNet, ResNet ve Xception tabanlı 18 farklı önceden-eğitilmiş ağ kullanılarak her bir senaryonun performans çıktıları ayrıntılı olarak incelenmiştir. Sınıflandırma için kullanılan veri seti cıvataları, vidaları ve somunları içeren *VS_Baglanti* veri setidir. Çalışmamızda

önerilen her bir yaklaşımın performansı ayrıntılı olarak incelenmiş ve çalışmamız literatürdeki diğer çalışmalarla karşılaştırılmıştır. Kullanılan modelin hiper-parametre ayarlaması Tablo 3.7' deki gibidir.

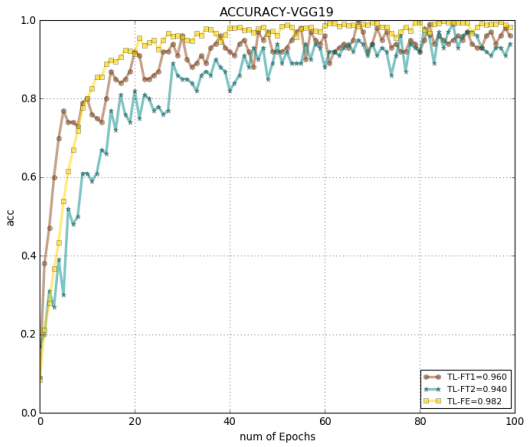
Tablo 3.7. Kullanılan KSA modelindeki hiper-parametreler

Hiper-parametre	**Değeri**
Optimizsayon algoritması	Adam
İterasyon	100
Parti-boyutu	32
Öğrenme oranı	0.001
Giriş görüntü boyutu	150x150

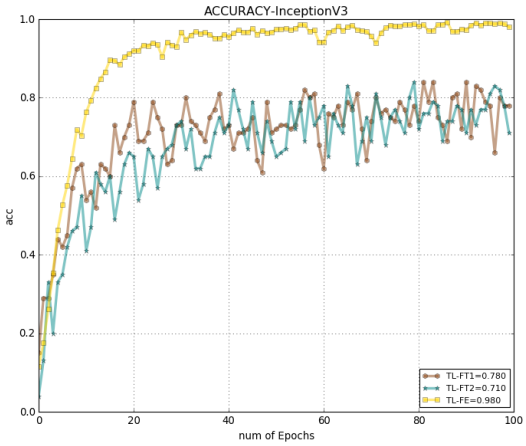
Çalışmamızdaki ilk senaryoda, ön-eğitimli ağ mimarisinin üzerinde sadece Softmax katmanındaki sınıf sayısı, güncellenerek ince-ayar yapılmıştır. İlk senaryoya '*TL-FT1*' adı verilmiştir. İkinci senaryoda, ön-eğitimli mimarinin son katmanlarına yeni katmanlar eklenmesiyle ince-ayar uygulanmıştır. İkinci senaryoya '*TL-FT2*' adı verilmiştir. Son senaryoda ise ön-eğitimli modelin sadece öznitelik çıkarım özelliği kullanılmış ve ardından çıkarılan yerel özellikler oluşturulan yeni bir KSA modelinde eğitilmiştir. Üçüncü senaryoya ise '*TL-FE*' adı verilmiştir. Her üç senaryoya ait performans metrikleri aşağıdadır.



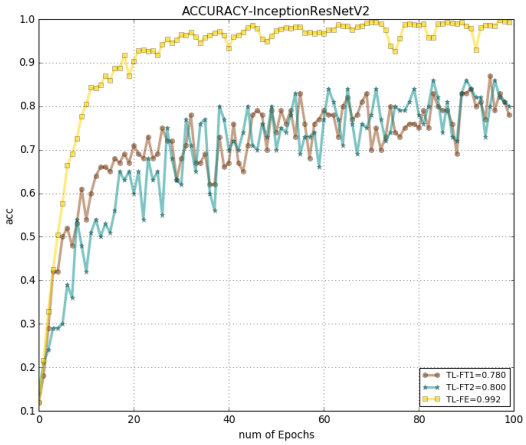
Şekil 3.6. VGG16 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



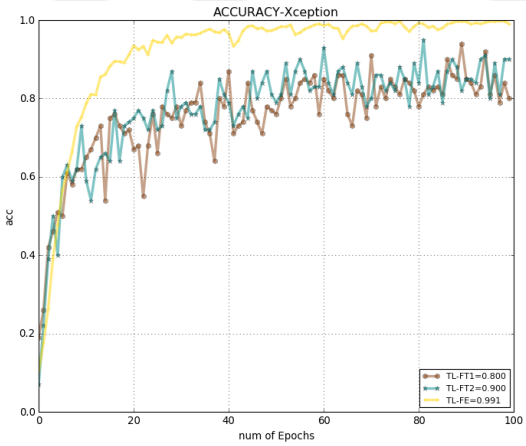
Şekil 3.7. VGG19 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



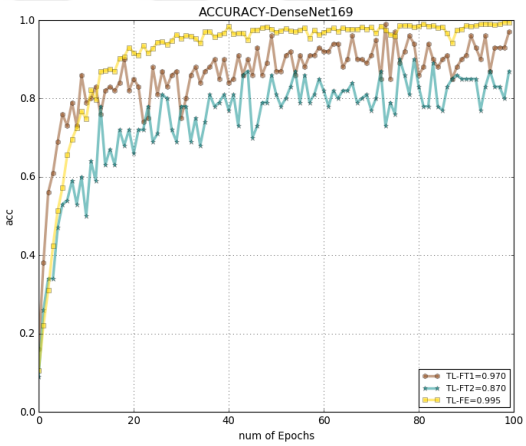
Şekil 3.8. InceptionV3 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



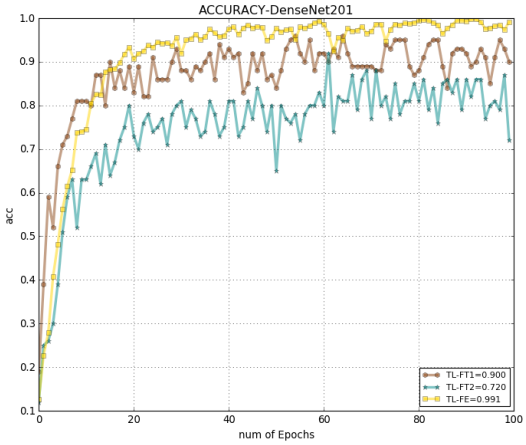
Şekil 3.9. InceptionResNetV2 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



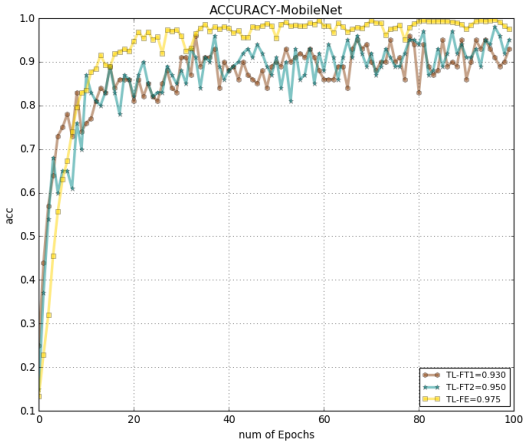
Şekil 3.10. Xception mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



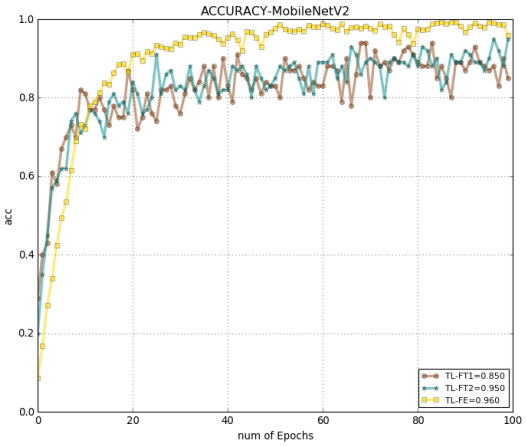
Şekil 3.11. DenseNet169 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



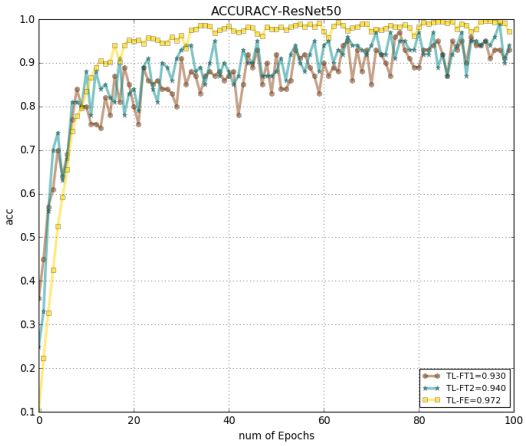
Şekil 3.12. DenseNet201 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



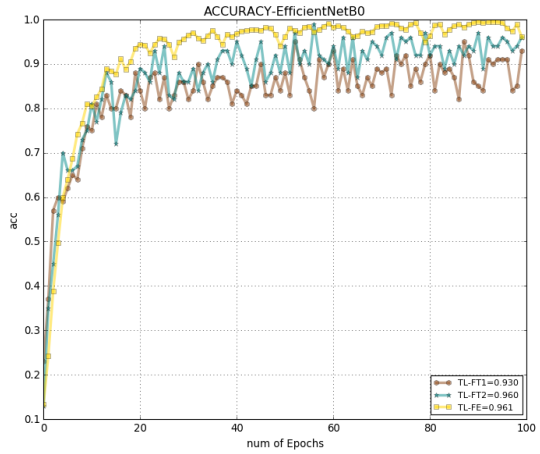
Şekil 3.13. MobileNet mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



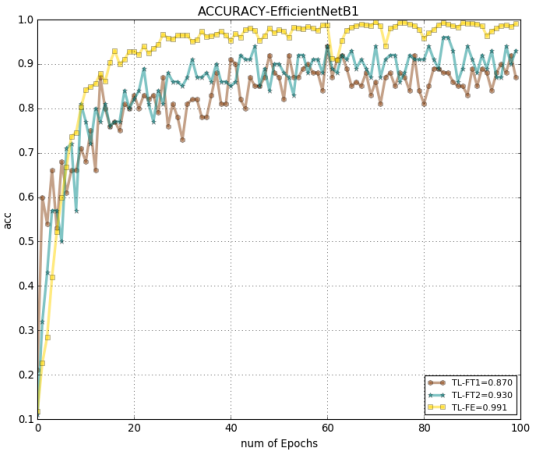
Şekil 3.14. MobileNetV2 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



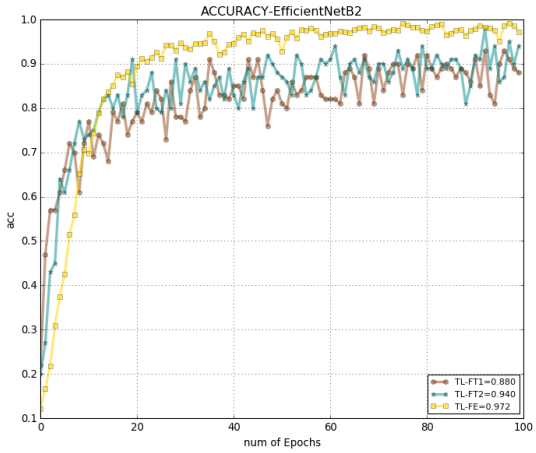
Şekil 3.15. ResNet50 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



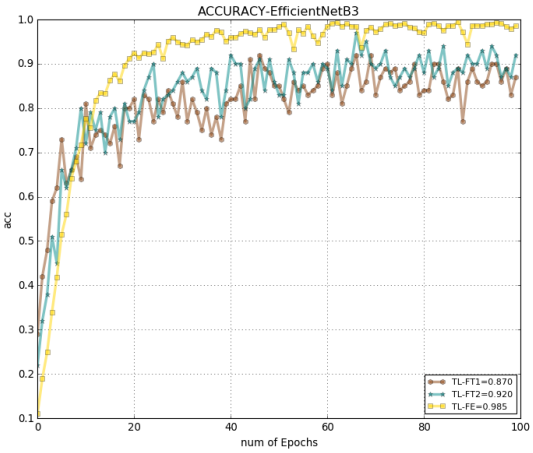
Şekil 3.16. EfficientNetB0 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



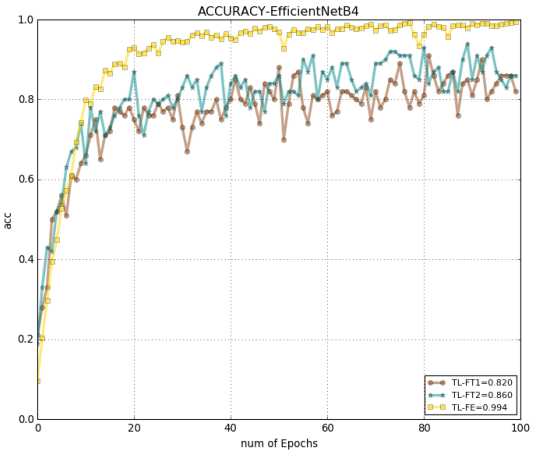
Şekil 3.17. EfficientNetB1 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



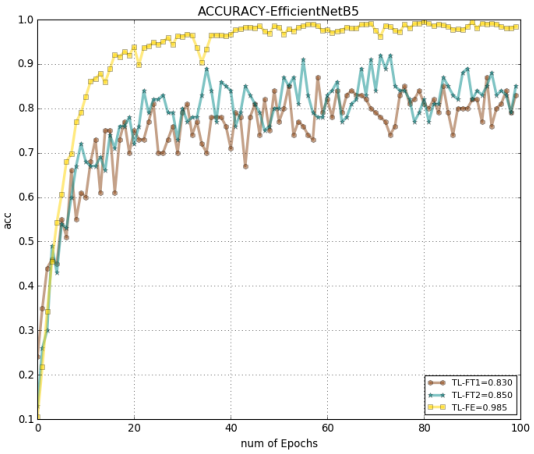
Şekil 3.18. EfficientNetB2 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



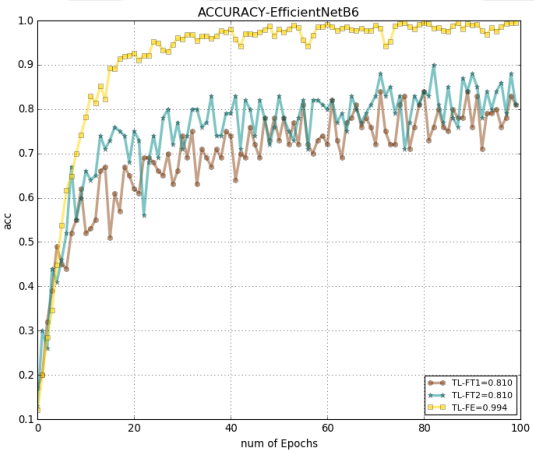
Şekil 3.19. EfficientNetB3 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



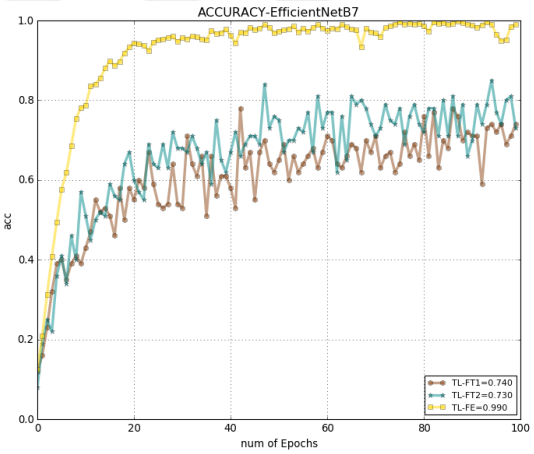
Şekil 3.20. EfficientNetB4 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



Şekil 3.21. EfficientNetB5 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı

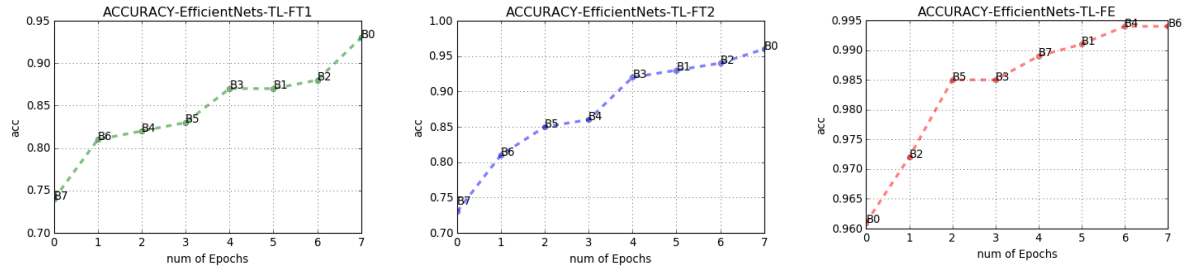


Şekil 3.22. EfficientNetB6 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı



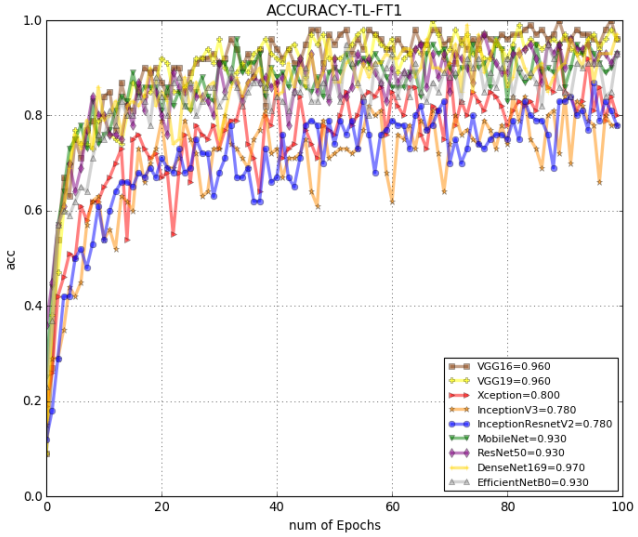
Şekil 3.23. EfficientNetB7 mimarisinin TL-FT1, TL-FT2 ve TL-FE senaryolarına göre doğruluk oranı

TL-FT1, TL-FT2 ve TL-FE senaryolarının üçünü her bir model için doğruluk oranları, Şekil 3.6 ile Şekil 3.23 arasında gösterilmiştir. TL-FT1, TL-FT2 ve TL-FE senaryoların TL-FT1, TL-FT2 ve TL-FE senaryoların doğruluk oranları, tüm mimarilerde incelendiğinde en yüksek doğruluk oranı TL-FE senaryosundan elde edilmiştir. Ancak her üç senaryo da tüm mimarilerde en düşük doğruluk oranı %70 seviyelerindedir. Aşağıda Şekil 3.24' te ise EfficientNet mimarilerinin, her bir senaryoya göre doğruluk oranı sıralaması görselleştirilmiştir. TL-FT1 ve TL-FT2 ince ayarlı transfer öğrenme yaklaşımında, en iyi EfficientNet mimarisi 'B0' ile elde edilmiştir. TL-FE senaryosunda ise en iyi EfficientNet mimarisi 'B6' olmuştur. EfficientNet mimarisinin her üyesi için senaryoların her biri üzerinde başarı sıralaması Şekil 3.24' te gösterilmiştir. Her senaryonun tüm modeller bazında karşılaştırmalı doğruluk oranları Şekil 3.25, Şekil 3.26 ve Şekil 3.27'de ifade edilmiştir.

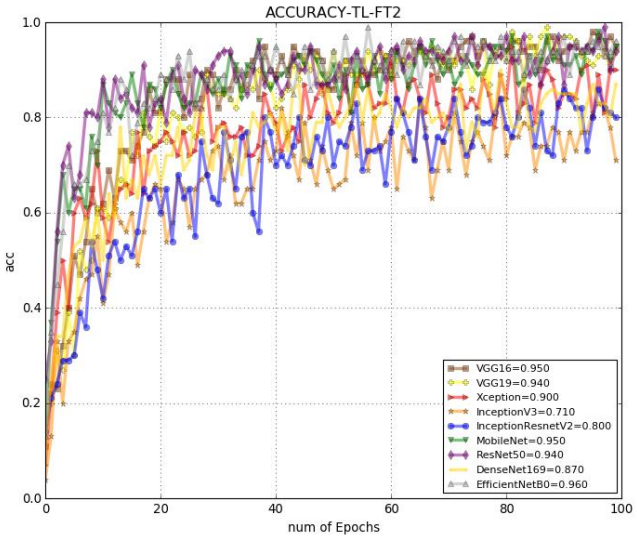


(a) TL-FT1 (b) TL-FT2 (c) TL-FE

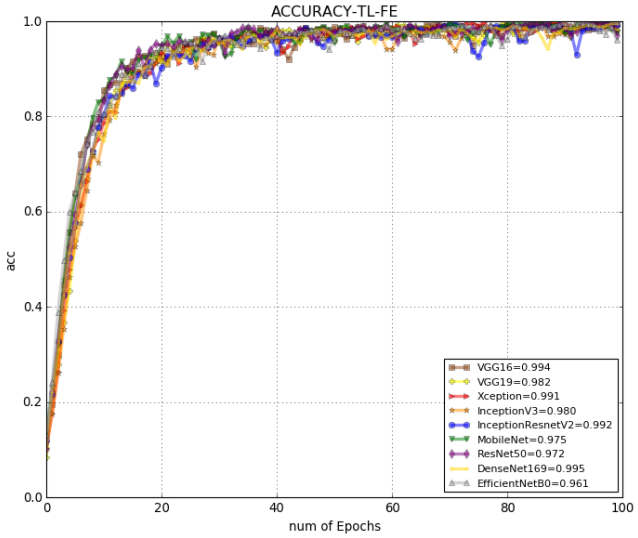
Şekil 3.24. EfficientNet mimarisinin üç senaryoya göre doğruluk oranı



Şekil 3.25. TL-FT1 senaryosuna göre tüm modellerin doğruluk oranı



Şekil 3.26. TL-FT2 senaryosuna göre tüm modellerin doğruluk oranı



Şekil 3.27. TL-FE senaryosuna göre tüm modellerin doğruluk oranı

Tablo 3.8. Tüm modellerin üç senaryoya göre doğruluk oranı karşılaştırması

Model	TL-FT1	TL-FT2	TL-FE
VGG16	0.960	0.950	0.9943
VGG19	0.960	0.940	0.982
InceptionV3	0.780	0.710	0.980
InceptionResNetV2	0.780	0.800	0.992
Xception	0.800	0.900	0.991
MobileNet	0.930	0.950	0.975
MobileNetV2	0.850	0.950	0.960
ResNet50	0.930	0.940	0.972
DenseNet169	**0.970**	0.870	**0.995**
DenseNet201	0.900	0.710	0.991
EfficientNetB0	0.930	**0.960**	0.961
EfficientNetB1	0.870	0.930	0.991
EfficientNetB2	0.880	0.940	0.972
EfficientNetB3	0.870	0.920	0.985
EfficientNetB4	0.820	0.860	0.994
EfficientNetB5	0.830	0.850	0.985
EfficientNetB6	0.810	0.810	0.994
EfficientNetB7	0.740	0.730	0.989

Tablo 3.8, TL-FE senaryosunun tüm modeller arasında en iyi sonuçları verdiğini göstermektedir. Bu durum, önerilen hibrit model TL-FE' nin başarılı olduğu anlamına gelmektedir. Tablo 3.8' de gösterildiği gibi, en iyi sonuçlara sahip modeller TL-FT1 senaryosunda 0,97 doğrulukla DenseNet169,

TL-FT2' de 0,96 ile EfficientNetB0 ve TL-FE' de 0,995 ile DenseNet169 olmuştur. Her modelin, senaryoların her birinden elde ettiği performans metrik çıktıları detaylı olarak tablolarda ifade edilmiştir. Performans metrik tablolarındaki A, A1, B, B1, C, C1, D ve D1 değişkenleri sırasıyla doğruluk, doğrulama doğruluğu, AUC, doğrulama AUC, kesinlik, doğrulama kesinliği, hassasiyet ve doğrulama hassasiyetine karşılık gelmektedir. TL-FT1, TL-FT2 ve TL-FE senaryolarının performans metrikleri sırasıyla Tablo 3.9, Tablo 3.10 ve Tablo 3.11' de gösterilmiştir.

Tablo 3.9. TL-FT1 senaryosuna göre performans metrikleri

Model	A	A1	B	B1	C	C1	D	D1
VGG16	0.960	0.860	0.964	0.964	0.899	0.899	0.820	0.821
VGG19	0.960	0.880	0.964	0.964	0.897	0.898	0.814	0.814
InceptionV3	0.780	0.510	0.819	0.819	0.563	0.564	0.537	0.537
InceptionResNetV2	0.870	0.270	0.781	0.781	0.494	0.494	0.465	0.465
Xception	0.800	0.480	0.826	0.825	0.605	0.605	0.597	0.597
MobileNet	0.930	0.700	0.896	0.896	0.747	0.747	0.742	0.743
MobileNetV2	0.850	0.510	0.848	0.848	0.659	0.659	0.655	0.655
ResNet50	0.930	0.030	0.741	0.741	0.498	0.498	0.456	0.456
DenseNet169	**0.970**	0.460	**0.995**	0.753	**0.970**	0.460	**0.970**	0.460
DenseNet201	0.900	0.430	0.979	0.721	0.900	0.430	0.900	0.430
EfficientNetB0	0.930	0.500	0.984	0.792	0.930	0.505	0.930	0.500
EfficientNetB1	0.870	0.420	0.962	0.731	0.870	0.420	0.870	0.420
EfficientNetB2	0.880	0.310	0.962	0.676	0.880	0.310	0.880	0.310
EfficientNetB3	0.870	0.320	0.957	0.671	0.869	0.327	0.860	0.320
EfficientNetB4	0.820	0.370	0.926	0.706	0.820	0.374	0.820	0.320
EfficientNetB5	0.830	0.420	0.951	0.723	0.838	0.420	0.830	0.420
EfficientNetB6	0.810	0.210	0.925	0.595	0.816	0.210	0.800	0.210
EfficientNetB7	0.740	0.240	0.892	0.598	0.740	0.232	0.740	0.230

Tablo 3.10. TL-FT2 senaryosuna göre performans metrikleri

Model	A	A1	B	B1	C	C1	D	D1
VGG16	0.950	0.860	0.960	0.960	0.890	0.890	0.737	0.738
VGG19	0.940	0.840	0.959	0.959	0.890	0.890	0.713	0.714
InceptionV3	0.710	0.510	0.870	0.870	0.606	0.606	0.487	0.488
InceptionResNetV2	0.800	0.320	0.803	0.803	0.498	0.498	0.400	0.400
Xception	0.900	0.570	0.886	0.886	0.659	0.660	0.563	0.564
MobileNet	0.950	0.660	0.941	0.941	0.783	0.783	0.716	0.716

MobileNetV2	0.950	0.480	0.906	0.906	0.702	0.702	0.626	0.627
ResNet50	0.940	0.070	0.858	0.858	0.897	0.897	0.426	0.426
DenseNet169	0.870	0.420	0.995	0.812	0.884	0.603	0.840	0.350
DenseNet201	0.720	0.410	0.971	0.830	0.758	0.479	0.690	0.230
EfficientNetB0	**0.960**	0.540	**1.000**	0.885	**0.970**	0.587	**0.960**	0.540
EfficientNetB1	0.930	0.430	0.999	0.851	0.948	0.442	0.920	0.420
EfficientNetB2	0.940	0.460	**1.000**	0.795	0.959	0.469	0.940	0.380
EfficientNetB3	0.920	0.550	0.993	0.875	0.935	0.538	0.870	0.420
EfficientNetB4	0.860	0.410	0.990	0.814	0.903	0.463	0.840	0.380
EfficientNetB5	0.850	0.300	0.993	0.720	0.878	0.304	0.790	0.280
EfficientNetB6	0.810	0.350	0.991	0.823	0.839	0.491	0.780	0.280
EfficientNetB7	0.730	0.330	0.966	0.840	0.788	0.340	0.670	0.320

Tablo 3.11. TL-FE senaryosuna göre performans metrikleri

Model	**A**	**A1**	**B**	**B1**	**C**	**C1**	**D**	**D1**
VGG16	0.994	0.972	**1.000**	0.994	**0.995**	0.972	0.994	0.972
VGG19	0.982	0.977	0.999	0.997	0.983	0.977	0.980	0.972
InceptionV3	0.980	0.977	0.999	0.994	0.981	0.983	0.978	0.977
InceptionResNetV2	0.992	0.972	**1.000**	0.994	0.993	0.972	0.992	0.972
Xception	0.991	0.943	0.999	0.988	0.991	0.943	0.991	0.938
MobileNet	0.975	0.943	0.999	0.996	0.977	0.949	0.974	0.943
MobileNetV2	0.960	0.938	0.997	0.990	0.961	0.938	0.959	0.938
ResNet50	0.972	0.909	0.997	0.981	0.973	0.909	0.972	0.909
DenseNet169	**0.995**	0.989	**1.000**	1.000	**0.995**	1.000	**0.995**	0.989
DenseNet201	0.991	0.989	**1.000**	0.997	0.992	0.989	0.989	0.989
EfficientNetB0	0.961	0.926	0.998	0.985	0.967	0.931	0.960	0.926
EfficientNetB1	0.991	0.972	**1.000**	0.994	0.991	0.972	0.989	0.972
EfficientNetB2	0.972	0.966	0.998	0.997	0.973	0.966	0.970	0.966
EfficientNetB3	0.985	0.989	**1.000**	1.000	0.987	0.989	0.984	0.989
EfficientNetB4	0.994	0.972	**1.000**	0.997	0.994	0.972	0.993	0.972
EfficientNetB5	0.985	0.926	**1.000**	0.973	0.987	0.926	0.985	0.926
EfficientNetB6	0.994	0.960	**1.000**	0.988	0.994	0.960	0.993	0.960
EfficientNetB7	0.990	0.938	**1.000**	0.985	0.991	0.938	0.990	0.938

Tablo 3.12. Literatürdeki Transfer Öğrenme metotlarıyla önerilen yöntem karşılaştırılması

Çalışma	Sınıf sayısı	Veri seti boyutu	Model	Metot	Doğruluk oranı
[31]	2	4986	DenseNet201	TF-FT2	0.9818
			DenseNet121		0.9577
			VGG19		0.9637
			VGG16		0.9437
			InceptionResNetV2		0.9457
			Xception		0.9195
[32]	2	460	VGG19	TF-FT2	0.8511
			DenseNet121		0.8533
			ResNet50		0.8206
			MobileNet		0.8642
[33]	15	20.639	ResNet50	TF-FT2	0.9926
[34]	2	2198	VGG16	TF-FT2	0.9146
[35]	2	40 sınıflı veri seti	AlexNet	TF-FT2	0.9675
		24 sınıflı veri seti	VGGNet		0.9747
		40 sınıflı veri seti	GoogLeNet		0.9597
[36]	2	613	ResNet-34	TF-FT1	1.000
[38]	4	236	AlexNet	TF-FT2	0.9153
			VGG19		0.9587
			ResNet50		0.8983
			ResNet150V2		0.874
[40]	7	38.000	Xception	TF-FT2	0.874
			DenseNet201		0.891
[41]	25	9339	VGG16	TF-FT1	0.9839
			VGG19		0.9871
			InceptionV3		0.9625
			ResNet50		0.9871
[42]	5	3520	VGG16	TF-FT1	0.9767
[43]	21	20.574	VGG16	TF-FT2	0.9847
			VGG19		0.9859
[44]	3	1596	Xception	TF-FT2	0.7830
			VGG16		0.7212
			VGG19		0.7006
			InceptionV3		0.7430
			InceptionResNet2		0.7840
			MobileNet		0.7237
[45]	3	11.862	InceptionV3	TF-FT1	0.9800
[46]	3	3924	VGG16	TF-FT1	0.9000
			VGG19		0.9200

[47]	20	10.000	VGG16 VGG19 InceptionV3 ResNet	TF-FT1	0.7890 0.7820 0. 8790 0.6991	
[48]	7	5648	EfficientNetB4 DenseNet201 EfficientNetB0 MobileNetV3	TF-FT1	0.9552 0.9405 0.9298 0.9640	
					TL-FT1	TL-FT2
Bizim çalışma	18	1760	VGG16 VGG19 InceptionV3 InceptionResNetV2 Xception MobileNet ResNet50 Dense201	TF-FT1 TF-FT2	0.960 0.960 0.780 0.780 0.800 0.930 0.930 0.900	0.950 0.940 0.710 0.800 0.900 0.950 0.940 0.710

Tablo 3.12 incelendiğinde, TL-FT1 senaryosu uygulandığında [46]' da VGG16' nın doğruluk oranı 0.90 ve VGG19'un doğruluk oranı 0.92'dir. TL-FT1 senaryosu uygulandığında [47]' de VGG16' nın değeri 0.789 ve VGG19' un değeri 0.782' dir. Çalışmamızda VGG16' nın doğruluğu 0.960, VGG19' un doğruluğu 0.960'dır. [31]' de TL-FT2 senaryosunun uygulanmasında, VGG16' nın doğruluğu 0.9437 ve VGG19' un doğruluğu 0.9637' dir. [32]' de TL-FT2 senaryosunun uygulanmasında ResNet' in doğruluğu 0.8206, MobileNet' in doğruluğu 0.8642 ve VGG19' un doğruluğu 0.8511' dir. [34]' te TL-FT2 senaryosunun uygulanmasında VGG16' nın doğruluğu 0.9146' dır. [38]' de TL-FT2 senaryosunun uygulanmasında ResNet50' nin doğruluğu 0.8983' tür. [40]' ta TL-FT2 senaryosunun uygulanmasında, Xception' ın doğruluğu 0.874' tür. [44]' te TL-FT2 senaryosunun uygulanmasında, VGG16' nın doğruluğu 0.7212, VGG19' un doğruluğu 0.7006, Xception' ın doğruluğu 0.783, InceptionV3' ün doğruluğu 0.743, InceptionResNetV2' nin doğruluğu 0.784 ve MobileNet' in doğruluğu 0.7237' dir.

Çalışmamızda VGG16'nın doğruluğu 0.950, VGG19'un doğruluğu 0.940, MobileNet'in doğruluğu 0.950, Xception'ın doğruluğu 0.90, InceptionV3'ün doğruluğu 0.710 ve InceptionResNetV2'nin doğruluğu 0.80' dir. Sonuçlardan da görüldüğü üzere birçok modelde, literatürdekilere kıyasla önerdiğimiz yaklaşımlar oldukça başarılı sonuçlar üretmiştir.

Bu çalışmanın ana katkıları aşağıda listelenmiştir:

- Transfer öğrenme tabanlı bağlantı elemanlarını sınıflandırmak için üç farklı yöntem önermektedir. Bunlardan iki tanesi ince-ayar yaklaşımını kullanırken, üçüncüsü özellik çıkarma yaklaşımını benimsemektedir. Literatürdeki çalışmalar, bağlantı elemanlarını ince-ayar yöntemlerinden sadece birini kullanarak sınıflandırırken bizim çalışmamız, üç farklı senaryoyu

uygulayarak detaylı olarak karşılaştırmaktadır. Bu açıdan bakıldığında, literatürdeki çalışmalara göre oldukça kapsamlı ve faydalı bir transfer öğrenme kaynağıdır.

- Önerilen sınıflandırma senaryolarından üçüncüsü olan FE ile sınıflandırma hibrit bir yöntemdir. KSA mimarisi ile önceden eğitilmiş bir TL ağını birleştiren üçüncü senaryoda, modele giriş olarak ham görüntü verilmektense TL ağı tarafından çıkarılan özellikler KSA ağına girdi olarak verilerek ağın başarısını artırmaktadır.
- Literatürde FT kullanan çalışmalar incelendiğinde FT' nin iki şekilde yapılabileceği görülmektedir. Çalışmamız, her iki FT yaklaşımını ayrı ayrı uygulayarak performans çıktılarını karşılaştırmaktadır. Bu durum, literatürdeki çoğu çalışmadan daha başarılı sonuçlara ulaşılmasına yol açmıştır.
- FT içeren mevcut çalışmalara bakıldığında, en fazla 6 türde önceden eğitilmiş TL modeli kullanılırken çalışmamız literatürde sınıflandırma için kullanılan TL modellerinin hemen hemen tamamını (18 adet) kullanarak en iyi sonuçları veren modeli seçmeyi amaçlamaktadır. Dolayısıyla bu çalışma literatürdeki en kapsamlı TL çalışmasından birisi olmuştur.
- Ayrıca derin KSA ağı, TL-FT ve TL-FE kullanarak az sayıda görüntü içeren bir veri seti üzerinde eğitilmesi ile yüksek performanslı sınıflandırma sonuçları elde edilmiştir.

3.4. Tekstil Makinesi Yedek Parçalarının Diferansiyel Gizlilik ve KSA ile Sınıflandırılması Sonuçları

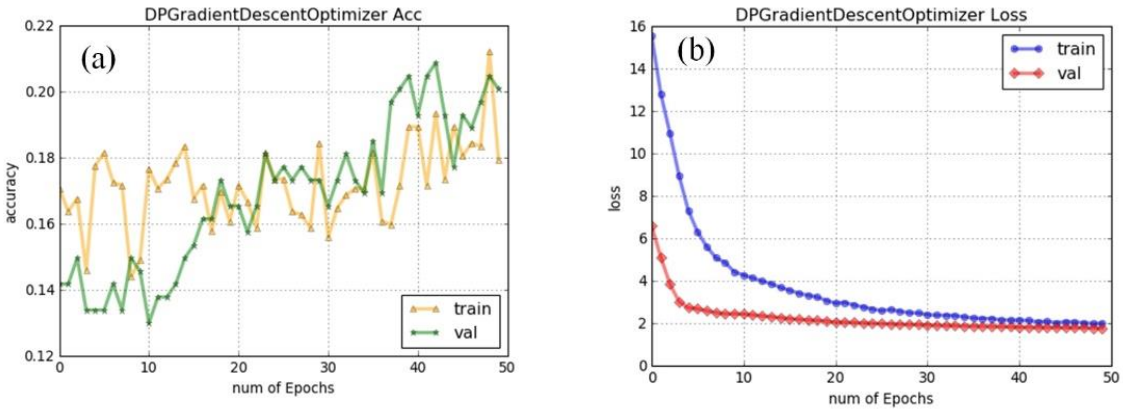
Diferansiyel gizlilik ve KSA yaklaşımı kullanılarak tasarlanan mimaride, 6 farklı yedek parça sınıflandırılması yapılmıştır. Geliştirilen model, VS_Nit1 veri seti üzerinde eğitim ve doğrulama adımlarını uygulamıştır. Modelin eğitimi bitirmesi 298 saniye sürmüştür. Diferansiyel gizlilik tekniği farklı optimizasyon algoritmaları üzerinde uygulanmıştır. Bu algoritmalardan elde edilen performans kriterleri Tablo 3.13’ te gösterilmektedir. Her bir algoritma tarafından üretilen karışıklık matrisleri de verilmiştir. Modelin eğitim süreci sonucunda, her bir algoritmanın elde ettiği doğruluk, kayıp, doğrulama doğruluğu ve doğrulama kaybı oranları Şekil 3.28 ile Şekil 3.35 arasında gösterilmiştir.

DG, gizliliğini sağlamak için sekiz farklı optimizasyon algoritması üzerinde test edilmiştir. Bu algoritmalar: DG-GradientDescent, DG-GradientDescentGaussian, DG-RMSProp, DG-RMSPropGaussian, DG-Adagrad, DG-AdagradGaussian, DG-Adam, DG-AdamGaussian' dır. Farklı optimizasyon algoritmalarına göre karmaşıklık matrisi Şekil 3.36’da verilmiştir. Her algoritmanın performans rapor çıktıları Tablo 3.14 ile Tablo 3.22 arasında sunulmuştur.

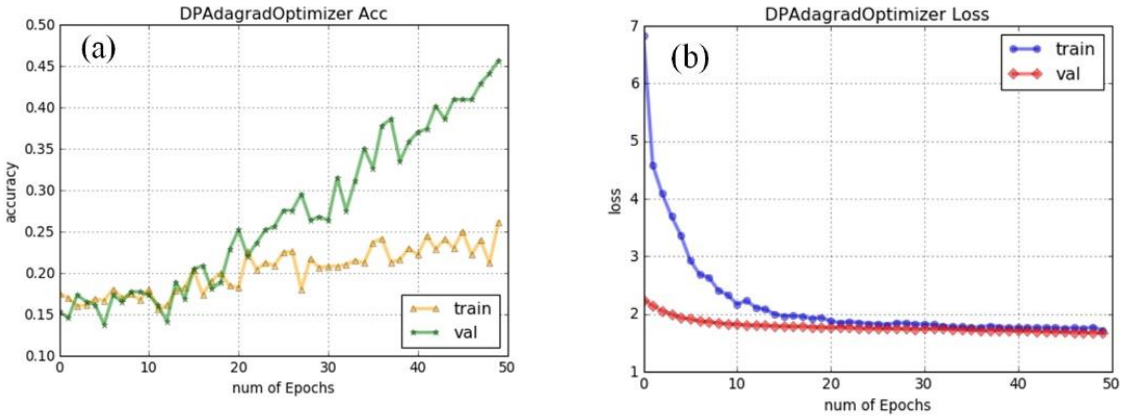
DG’ nin ε-değeri, veri setindeki gizlilik maliyet değeridir. Bir ε-değeri, bir veri setindeki bir veri çıktısının, komşu bir veri setindeki aynı modelin çıktısından ne kadar farklı olmasına neden olduğunu araştırmaktadır. Bu değişken, gizliliğin korunup korunmadığını göstermektedir. ε-değeri, ne kadar küçükse gizlilik o kadar büyük olmaktadır. Tahmin sayısı ne kadar yüksek olursa gizlilik maliyet değeri de o kadar yüksek olmaktadır.

Tablo 3.13. Geliştirilen KSA mimarisinin parametreleri

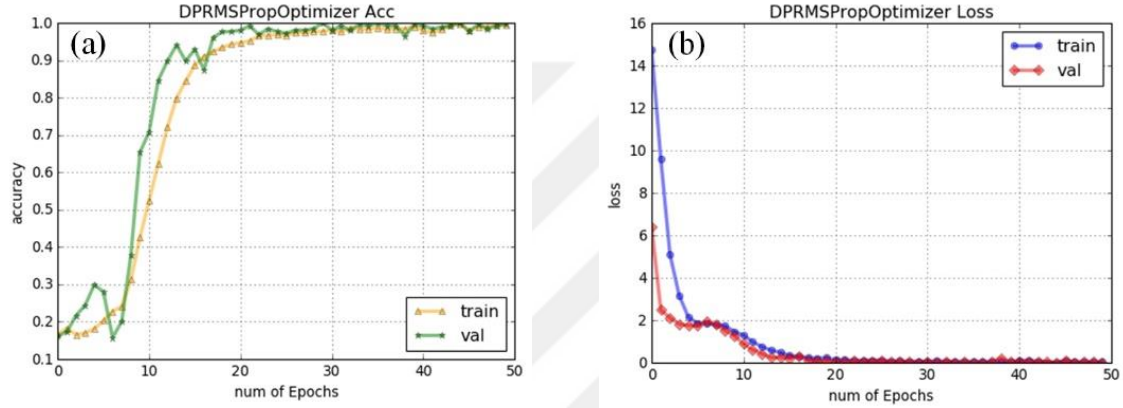
Katman	**Çıktı değeri**
Giriş görüntüsü	150x150
Konvolüsyon-1 (3x3 boyutunda 32 filtre)	148, 148, 32
Max Havuzlama-1 (2x2 boyutunda 32 filtre)	74, 74, 32
Konvolüsyon -2 (3x3 boyutunda 64 filtre)	72, 72, 64
Max Havuzlama -2 (2x2 boyutunda 64 filtre)	36, 36, 64
Konvolüsyon -3 (3x3 boyutunda 64 filtre)	34, 34, 64
Max Havuzlama -3 (2x2 boyutunda 64 filtre)	17, 17, 64
Bırakma	0.25
Konvolüsyon -4 (3x3 boyutunda 128 filtre)	15, 15, 128
Max Havuzlama -4 (2x2 boyutunda 128 filtre)	7, 7, 128
Bırakma	0.25
Konvolüsyon -5 (3x3 boyutunda 256 filtre)	5, 5, 256
Max Havuzlama -5 (2x2 boyutunda 256 filtre)	2, 2, 256
Bırakma	0.25
Tam bağlantı 1	1x262400
Tam bağlantı 2	1x65792
Tam bağlantı 3	1028
Softmax	6 sınıflı
İterasyon	50
Parti boyutu	128



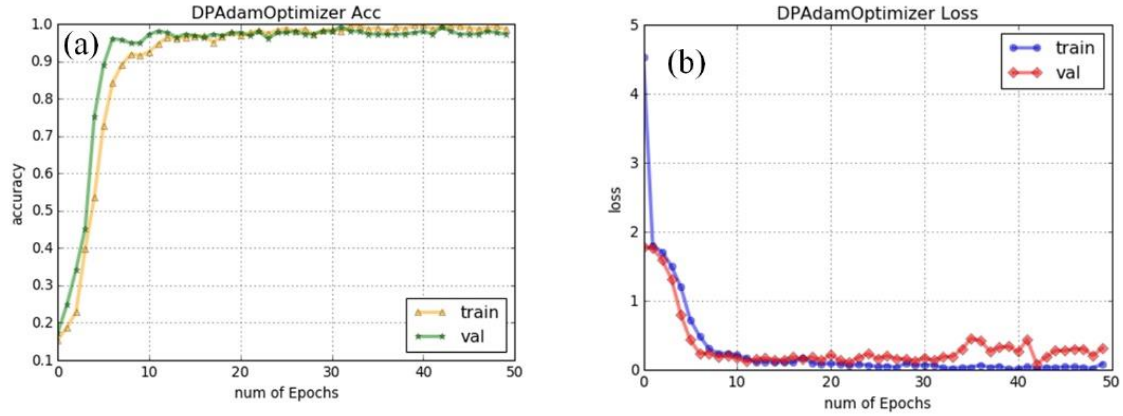
Şekil 3.28. DG-GradientDescent' in eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı



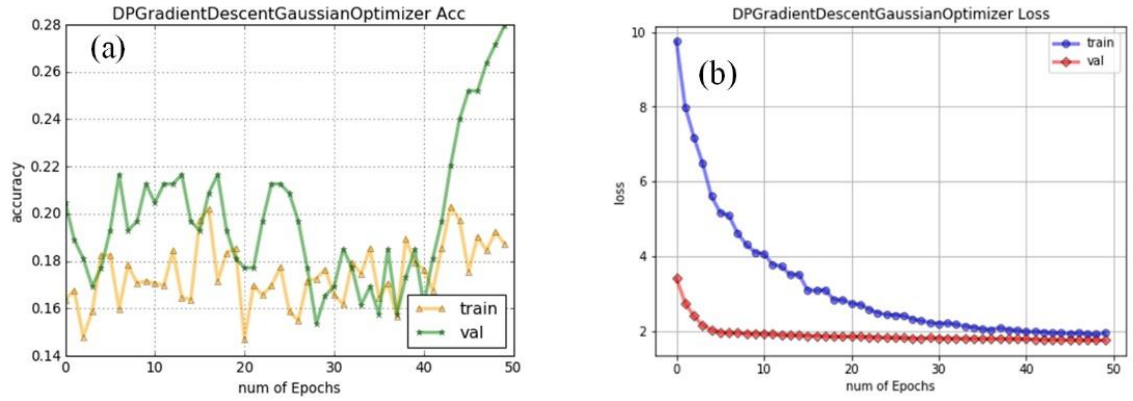
Şekil 3.29. DG-Adagrad' ın eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı



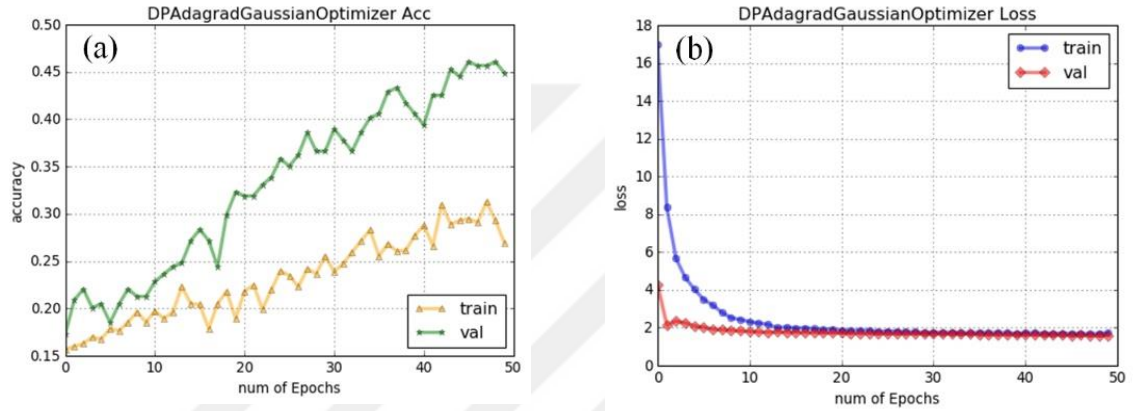
Şekil 3.30. DG-RMSProp' ın eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı



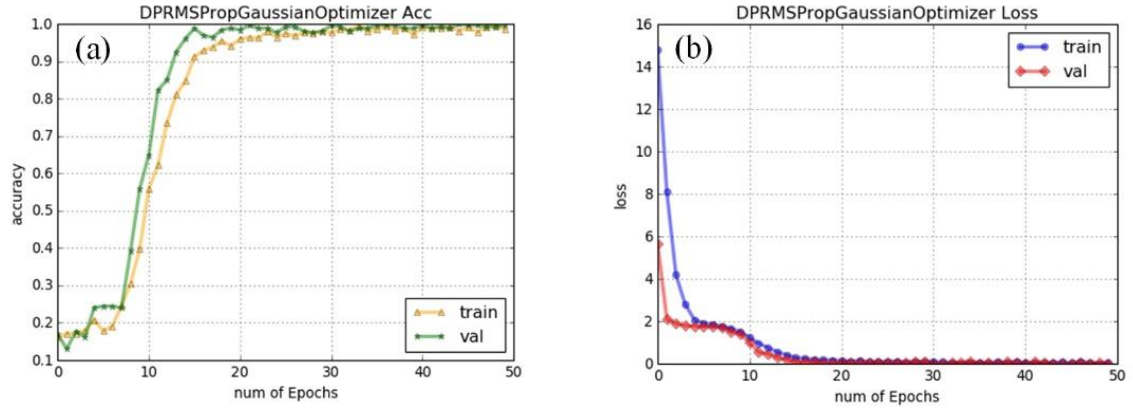
Şekil 3.31. DG-Adam' ın eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı



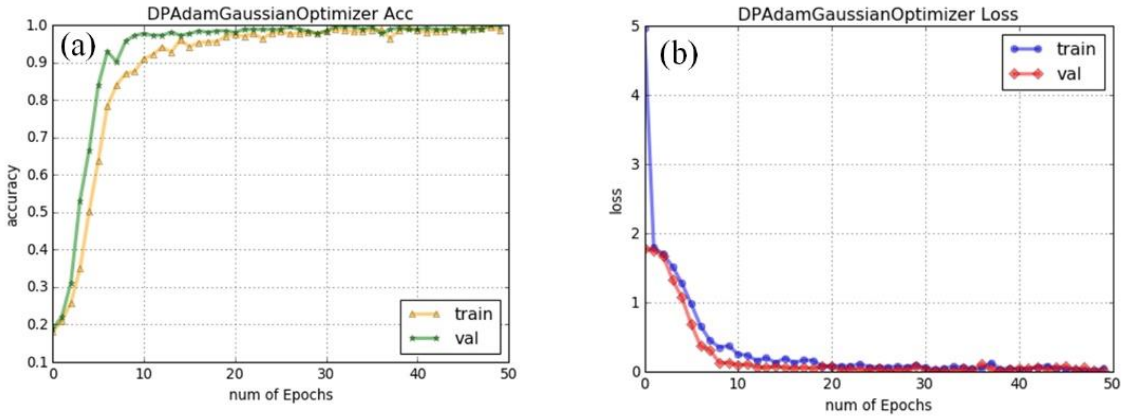
Şekil 3.32. DG-GradientDescentGaussian' ın eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı



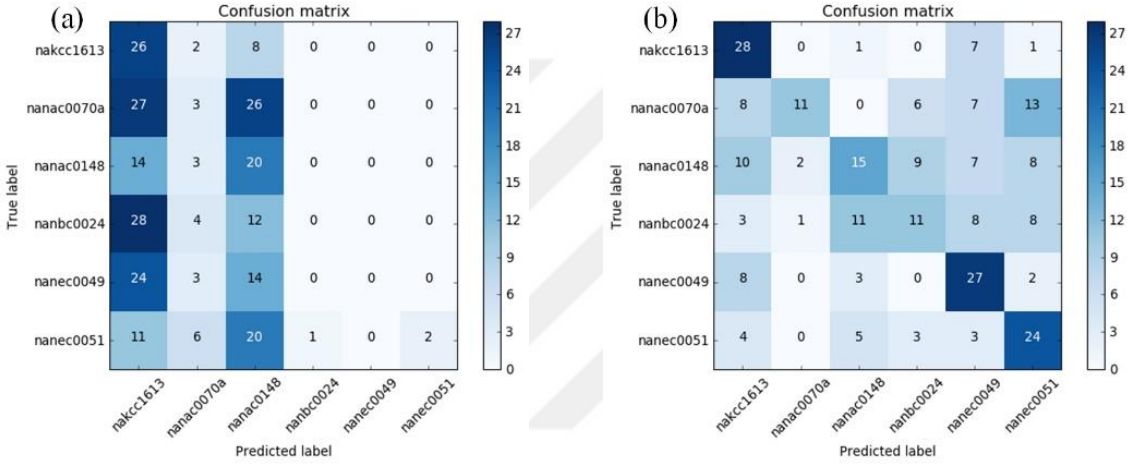
Şekil 3.33. DG-AdagradGaussian' ın eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı



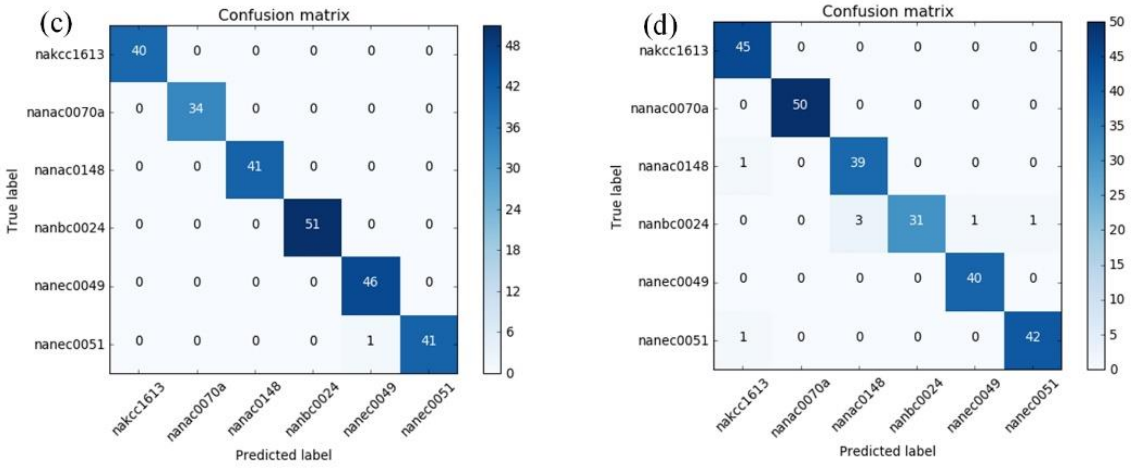
Şekil 3.34. DG-RMSPropGaussian' ın eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı



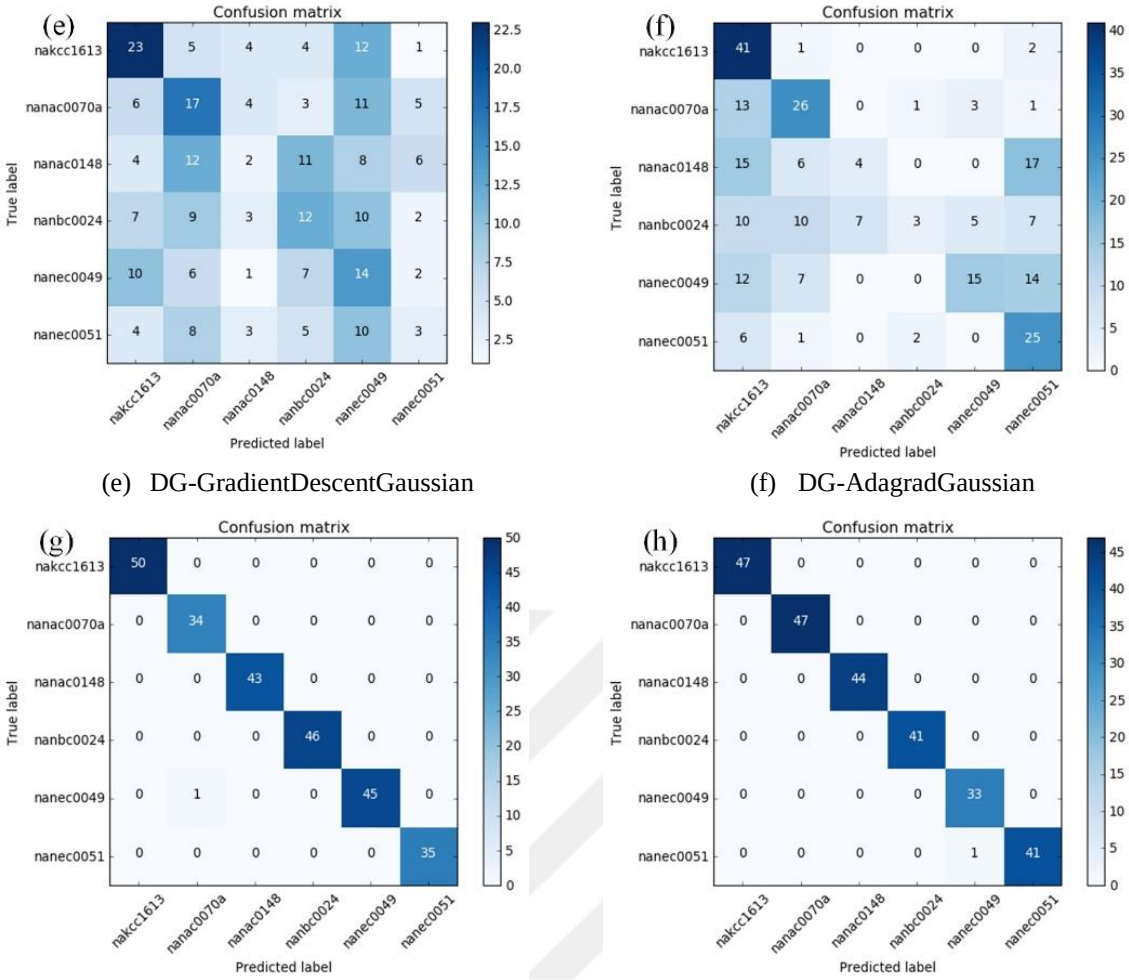
Şekil 3.35. DG-AdamGaussian' ın eğitim ve doğrulama sonucunda (a) doğruluk ve (b) kayıp oranı



(a) DG-GradientDescent (b) DG-Adagrad



(c) DG-RMSProp (d) DG-Adam



(e) DG-GradientDescentGaussian (f) DG-AdagradGaussian

(g) DG-RMSPropGausian (h) DG-AdamGaussian

Şekil 3.36. Önerilen modelin farklı optimizasyon algoritmalarına göre karmaşıklık matrisi

Tablo 3.14. DG-GradientDescent' in sınıflandırma raporu

Etiket	Kesinlik	Hassasiyet	F1-Ölçüm	Örnek sayısı
Nakcc1613	0.20	0.72	0.31	36
Nanac0070a	0.14	0.05	0.08	56
Nanac0148	0.20	0.54	0.29	37
Nanbc0024	0.00	0.00	0.00	44
Nanec0049	0.00	0.00	0.00	41
Nanec0051	1.00	0.05	0.10	40

Tablo 3.15. DG-Adagrad' ın sınıflandırma raporu

Etiket	Kesinlik	Hassasiyet	F1-Ölçüm	Örnek sayısı
Nakcc1613	0.46	0.76	0.57	37
Nanac0070a	0.79	0.24	0.37	45
Nanac0148	0.43	0.29	0.35	51
Nanbc0024	0.38	0.26	0.31	42
Nanec0049	0.46	0.68	0.55	40
Nanec0051	0.43	0.62	0.51	39

Tablo 3.16. DG-RMSProp' ın sınıflandırma raporu

Etiket	Kesinlik	Hassasiyet	F1-Ölçüm	Örnek sayısı
Nakcc1613	1.00	1.00	1.00	40
Nanac0070a	1.00	1.00	1.00	34
Nanac0148	1.00	1.00	1.00	41
Nanbc0024	1.00	1.00	1.00	51
Nanec0049	0.98	1.00	0.99	46
Nanec0051	1.00	0.98	0.99	42

Tablo 3.17. DG-Adam' ın sınıflandırma raporu

Etiket	Kesinlik	Hassasiyet	F1-Ölçüm	Örnek sayısı
Nakcc1613	0.96	1.00	0.98	45
Nanac0070a	1.00	1.00	1.00	50
Nanac0148	0.93	0.97	0.95	40
Nanbc0024	1.00	0.86	0.96	36
Nanec0049	0.98	1.00	0.99	40
Nanec0051	0.98	0.98	0.98	43

Tablo 3.18. DG- GradientDescentGaussian' ın sınıflandırma raporu

Etiket	Kesinlik	Hassasiyet	F1-Ölçüm	Örnek sayısı
Nakcc1613	0.43	0.47	0.45	49
Nanac0070a	0.30	0.37	0.33	46
Nanac0148	0.12	0.05	0.07	43
Nanbc0024	0.29	0.28	0.28	43
Nanec0049	0.22	0.35	0.27	40
Nanec0051	0.16	0.09	0.12	33

Tablo 3.19. DG- AdagradGaussian' ın sınıflandırma raporu

Etiket	Kesinlik	Hassasiyet	F1-Ölçüm	Örnek sayısı
Nakcc1613	0.42	0.93	0.58	44
Nanac0070a	0.51	0.59	0.55	44
Nanac0148	0.36	0.10	0.15	42
Nanbc0024	0.50	0.07	0.12	42
Nanec0049	0.65	0.31	0.42	48
Nanec0051	0.38	0.74	0.50	34

Tablo 3.20. DG- RMSPropGaussian' ın sınıflandırma raporu

Etiket	Kesinlik	Hassasiyet	F1-Ölçüm	Örnek sayısı
Nakcc1613	1.00	1.00	1.00	50
Nanac0070a	0.97	1.00	0.99	34
Nanac0148	1.00	1.00	1.00	43
Nanbc0024	1.00	1.00	1.00	46
Nanec0049	1.00	0.98	0.99	46
Nanec0051	1.00	1.00	1.00	35

Tablo 3.21. DG- AdamGaussian' ın sınıflandırma raporu

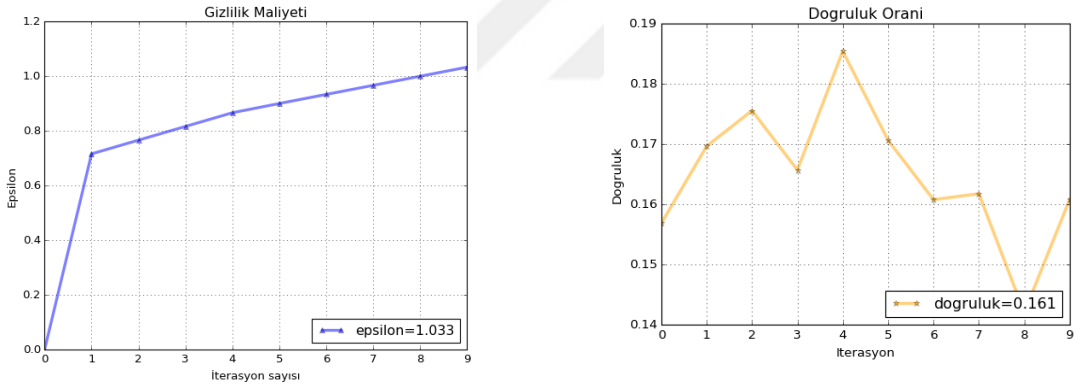
Etiket	Kesinlik	Hassasiyet	F1-Ölçüm	Örnek sayısı
Nakcc1613	1.00	1.00	1.00	47
Nanac0070a	1.00	1.00	1.00	47
Nanac0148	1.00	1.00	1.00	44
Nanbc0024	1.00	1.00	1.00	41
Nanec0049	0.97	1.00	0.99	33
Nanec0051	1.00	0.98	0.99	42

Tablo 3.22. Bu çalışmada kzullanılan optimizasyon algoritmalarının performans metrikleri

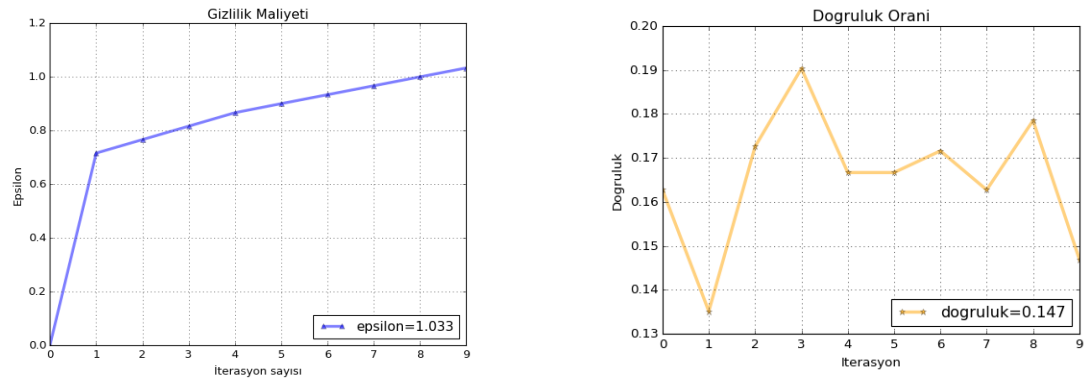
Algoritma	Doğruluk	Hata	Doğrulama doğruluğu	Doğrulama hatası
DG-GradientDescent	0.1795	1.9967	0.2008	1.7690
DG-Adagrad	0.2613	1.7178	0.4567	1.6719
DG-RMSProp	**0.9941**	0.0414	0.9921	0.0311
DG-Adam	0.9872	0.0821	0.9724	0.3082
DG-GradientDescentGaussian	0.1874	1.9414	0.2795	1.7474
DG-AdagradGaussian	0.2692	1.7232	0.4488	1.5906

DG-RMSPropGaussian	0.9862	0.0645	0.9961	0.0108
DG-AdamGaussian	0.9852	0.0209	1.0000	0.0131

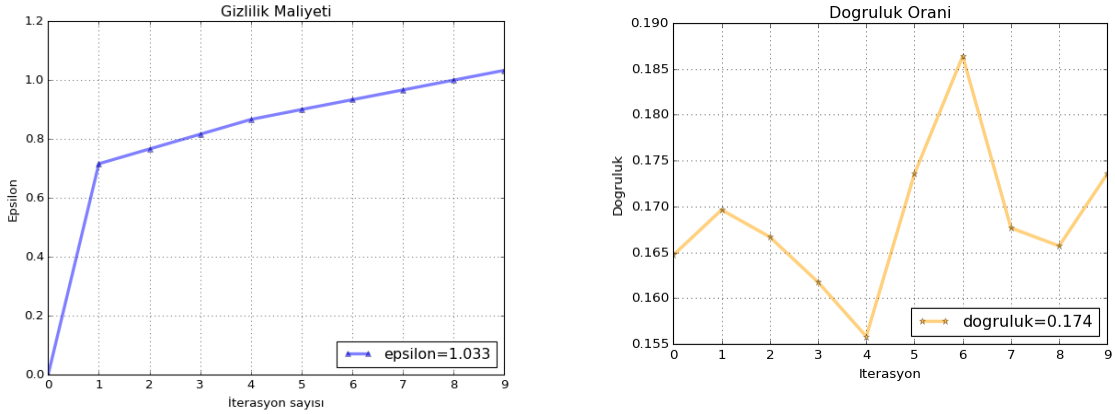
DG hiper-parametrelerinden öğrenme oranı (α) ne kadar yüksek olursa her güncelleme o kadar önemli olmaktadır. Gürültü çarpanı (ϑ), eğitim sırasında veriye eklenen gürültü miktarıdır. Bu değer ne kadar fazla olursa modelin gizliliği de o kadar yüksek olmaktadır. Çarpan değerinin, kırpma değerine göre büyük olduğu durumda, eğitim işleminin yakınsaması için öğrenme oranı düşük tutulmalıdır. Çarpan değeri, kırpma değeri ve öğrenme oranı parametrelerinin gizlilik maliyeti ve doğruluk oranı ile ilişkisi Şekil 3.37 ve Şekil 3.43 arasında sunulmuştur. Genel bir karşılaştırma tablo halinde Tablo 3.23'te verilmiştir. Mikro-parça boyutu, eğitim setindeki örneklerin parçalara bölüneceği miktarı belirtmektedir. Bu değeri arttırmak eğitim süresini uzatmaktadır ve eğitimi yavaşlatmasına rağmen eğitim performansı için olumlu sonuçlar vermektedir. L2 kırpma değeri (γ), her mikro-partiden gelen tüm ağ parametreleri arasındaki kümülatif eğim değeridir. Eğim değerinin en fazla olması gereken değere göre kırpılmasıdır. Her bir eğim değerine, L2-norm (Öklid) uygulamaktadır. DG uygulamalarında, gizlilik maliyeti (ε) ile doğruluk oranı arasındaki ilişkinin anlaşılması için birkaç senaryo gösterilmiştir. Bu işlem için parça-boyutu, mikro-parça boyutu, iterasyon ve optimizasyon algoritması değerleri sırasıyla 4, 256, 10 ve Gaussian_RMSProp ortak kullanılan hiper-parametrelerdir.



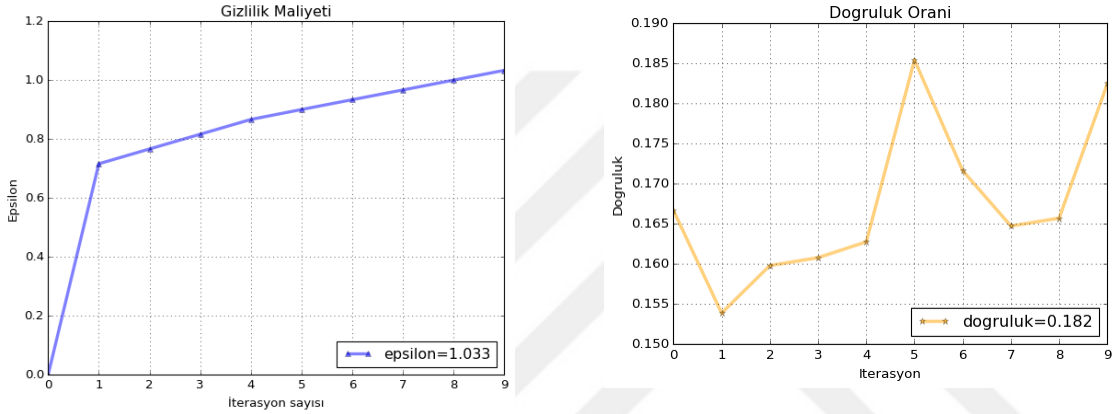
Şekil 3.37. γ=1.0, ϑ=1.1 ve α=0.25 koşulları altında gizlilik maliyeti ve doğruluk oranı



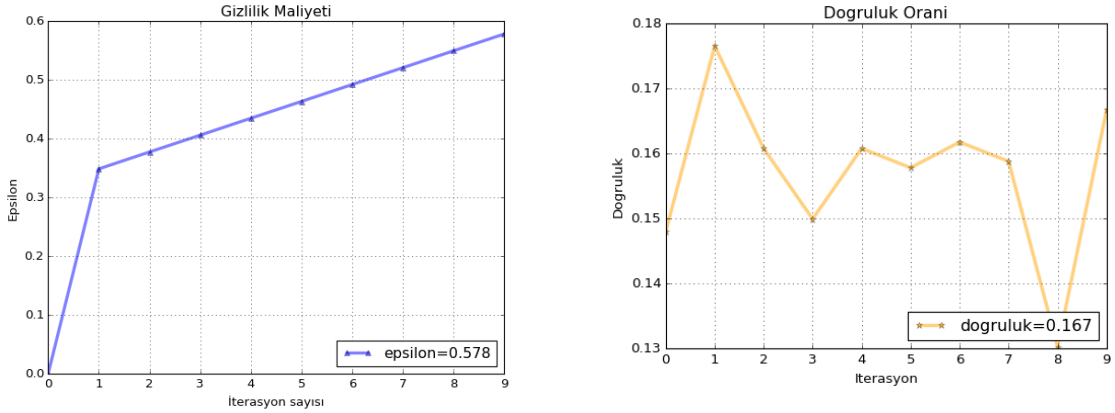
Şekil 3.38. γ=1.5, ϑ=1.1 ve α=0.15 koşulları altında gizlilik maliyeti ve doğruluk oranı



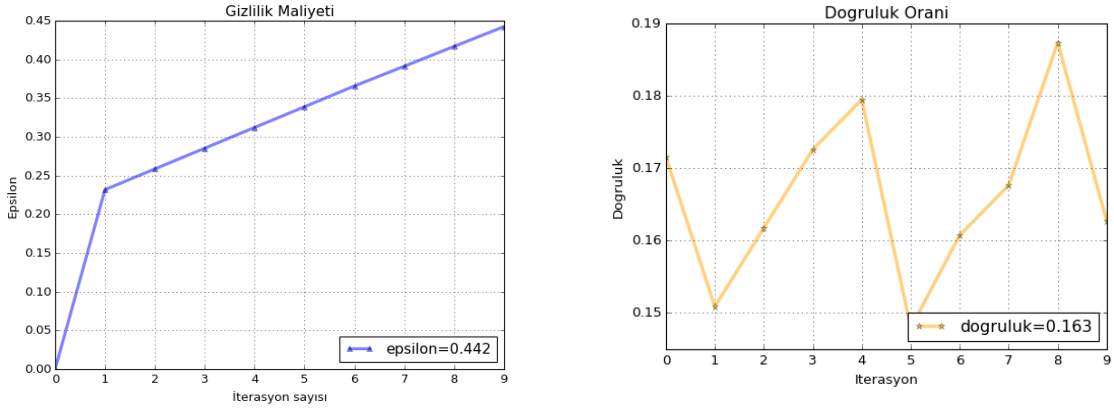
Şekil 3.39. γ=1.0, ϑ=1.1 ve α=0.15 koşulları altında gizlilik maliyeti ve doğruluk oranı



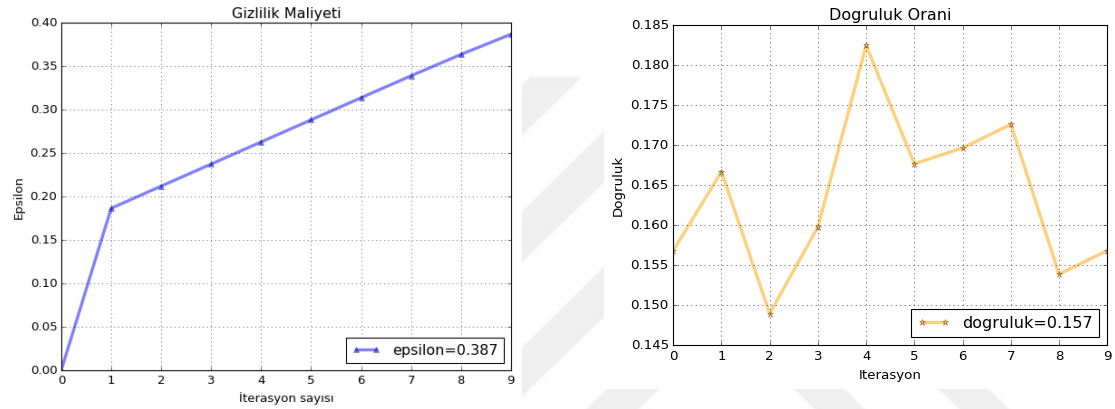
Şekil 3.40. γ=0.5, ϑ=1.1 ve α=0.15 koşulları altında gizlilik maliyeti ve doğruluk oranı



Şekil 3.41. γ=1.0, ϑ=1.5 ve α=0.15 koşulları altında gizlilik maliyeti ve doğruluk oranı



Şekil 3.42. γ=1.0, ϑ=1.8 ve α=0.15 koşulları altında gizlilik maliyeti ve doğruluk oranı



Şekil 3.43. γ=1.0, ϑ=2.0 ve α=0.15 koşulları altında gizlilik maliyeti ve doğruluk oranı

Tablo 3.23. DG algoritmasındaki örnek senaryolar üzerinden gizlilik maliyeti ve doğruluk oranı ilişkisi

#	Öğrenme oranı (α)	Gürültü çarpanı (ϑ)	L2 kırpma değeri (γ)	Gizlilik maliyeti (ε)	Doğruluk oranı (%)
1	0.25	1.1	1.0	1.033	0.161
2	0.15	1.1	1.5	1.033	0.147
3	0.15	1.1	1.0	1.033	0.174
4	0.15	1.1	0.5	1.033	0.182
5	0.15	1.5	1.0	0.578	0.167
6	0.15	1.8	1.0	0.442	0.163
7	0.15	2.0	1.0	0.387	0.157

Tablo 3.23 incelendiğinde, öğrenme oranının gizlilik maliyeti üzerinde bir etkisi olmadığı sadece doğruluk oranını etkilediği gözlemlenmektedir. Öğrenme oranı ne kadar az olursa doğruluk oranı da o kadar yüksek olmaktadır. DG parametrelerinden gürültü çarpanı, sistem gizliliğinin yüksek olmasını sağlamaktadır. Veri setindeki görüntülere eklenen gürültü çarpanı ile gizlilik artmaktadır; ancak görüntülerin tanınırlığı düştüğünde doğruluk oranı da düşmektedir. Bir diğer DG parametresi olan L2 kırpma değeri, doğruluk oranı ile ters bir ilişkidedir. DG uygulamalarında, gizlilik ile gizlilik maliyeti arasında da ters bir ilişki vardır. Gizlilik maliyeti ne kadar azsa elde edilen sistem daha güçlü bir

gizlilikte olduğu söylenebilmektedir. Gizlilik maliyetinin genellikte 1' den küçük olması beklenmektedir. Böylece sistemin güçlü bir gizlilikte olduğu düşünülür. Buradan da anlaşılacağı üzere gürültü çarpanı parametresi ile gizlilik maliyeti arasında ters bir ilişki bulunmaktadır.

Tezin bu bölümde önerilen yöntemin katkıları aşağıda özetlenmiştir:

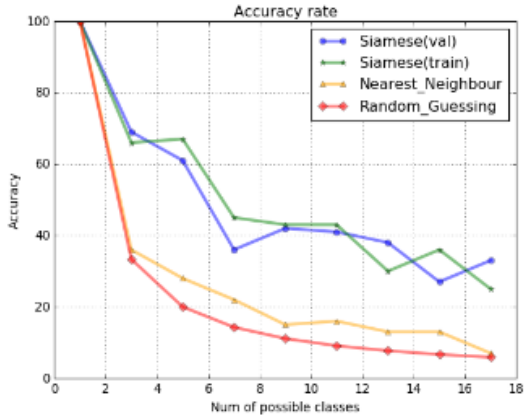
- Literatürde, yedek parça sınıflandırmasına yönelik makine öğrenimine dayalı bir yaklaşım bulunmamaktadır. Bunun yerine literatürde, genellikle dilsel sınıflandırma süreci kullanılmıştır. Bu açıdan çalışmamız, literatüre katkı sağlamaktadır.
- Derin sinir ağları kullanılarak bir KSA modeli geliştirilmiştir. Bu model, diğer veri setleri için de uyarlanabilir haldedir.
- Geliştirilen modelin güvenliğini artırmak için DG tabanlı yöntemler uygulanarak doğruluk oranları karşılaştırılmıştır. Çalışmamızda, sekiz adet DG-KSA yaklaşımı uygulanmıştır ve hangisinin en iyi sonucu verdiği deneysel olarak belirlenmiştir.
- [136]' da iki farklı veri seti üzerinde çalışılmıştır ve başarı oranları %98.1 ve %81.5' dir. [137]' de yerel gizlilik, MNIST ve CIFAR veri setlerinde test edilmiştir ve sırasıyla %96 ve %91 oranlarında başarı elde edilmiştir. Çalışmamızda ise oluşturduğumuz veri setinden elde ettiğimiz doğruluk oranı %99.41' dir.

3.5. Bağlantı Elemanlarının Siyam KSA ve TSÖ ile Sınıflandırılması Sonuçları

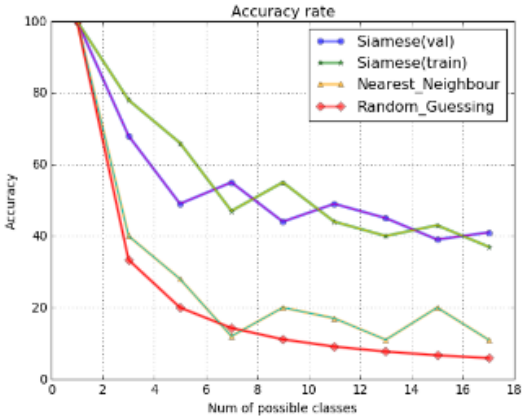
Siyam KSA mimarisi ve TSÖ ile *VS_Baglanti* veri setinin sınıflandırılmasında uygulanmıştır. Bu veri setinde, 18 kategori bulunmaktadır. Kategorilerde yer alan görseller arasındaki benzerlik oldukça fazla olduğu için onları sınıflandırmak zorlaşmaktadır. Bu çalışmada, 3-way, 5-way ve 10-way TSÖ ile geliştirilen Siyam KSA ağı test edilmiştir. Elde edilen performans sonuçları sunulmuştur. Önerilen ağda, aktivasyon fonksiyonu olarak Sigmoid, Softplus ve Tanh kullanılarak her birine ait performans sonuçları verilmiştir. 1-NN algoritması ve rastgele tahmin algoritması *VS_Baglanti* veri seti üzerinde çalıştırılmıştır. Her iki algoritmadan elde edilen sonuçlar, önerilen yöntemle karşılaştırmalı olarak sunulmuştur. Tablo 3.24 önerilen çalışmanın hiper-parametre ayarlamalarını içermektedir.

Tablo 3.24. Bu çalışmada kullanılan hiper-parametre değerleri

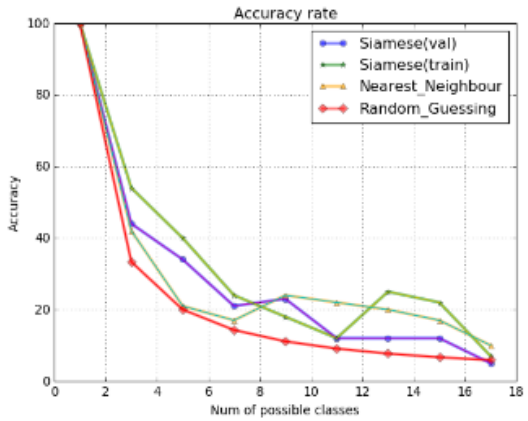
Hiper-parametre	Hiper-parametre değerlerinin açıklaması	Değer
Parti boyutu	Her yinelemedeki görüntü çiftlerinin sayısı	16
İterasyon	Ağın eğitim adımlarının sayısı	100
n_val	Doğrulanacak tek seferlik işlem sayısı	50
n_way	Tek seferlik işlemleri test etmek için sınıf sayısı	3, 5, 10
evaluate_every	Tek seferlik işlemleri değerlendirmek için aralık değeri	1
loss_every	İterasyonlarda yazdırma kaybı için aralık değeri	50
loss_function	Kayıp değerini hesaplamak için kullanılan ağın işlevi	L1_Distance



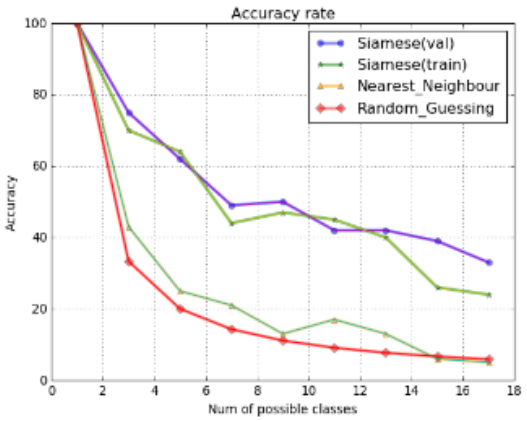
Şekil 3.44. Sigmoid aktivasyon fonksiyonu, L1-distance ve 3-way TSÖ ile olası sınıf sayısına göre doğruluk oranı



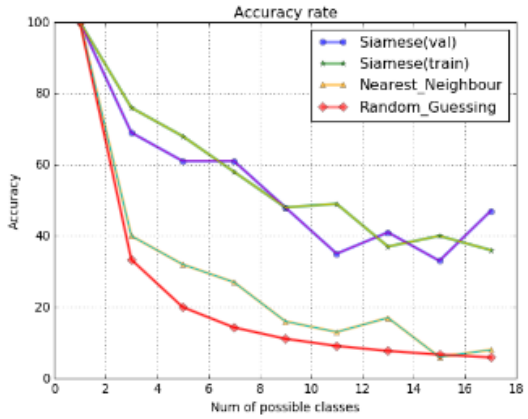
Şekil 3.45. Softplus aktivasyon fonksiyonu, L1-distance ve 3-way TSÖ ile olası sınıf sayısına göre doğruluk oranı



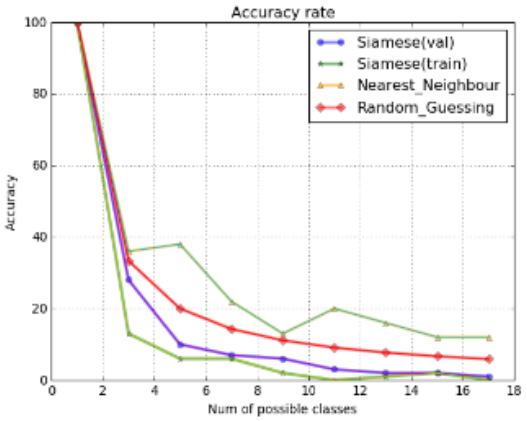
Şekil 3.46. Tanh aktivasyon fonksiyonu, L1-distance ve 3-way TSÖ ile olası sınıf sayısına göre doğruluk oranı



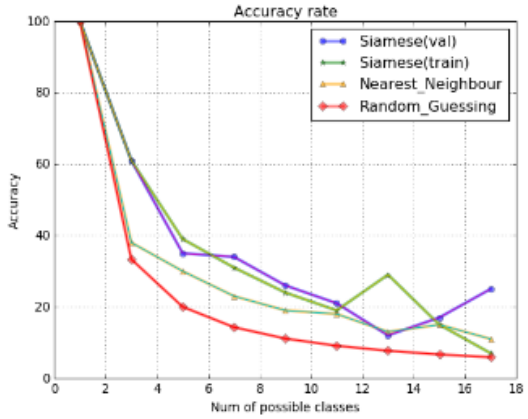
Şekil 3.47. Sigmoid aktivasyon fonksiyonu, L1-distance ve 5-way TSÖ ile olası sınıf sayısına göre doğruluk oranı



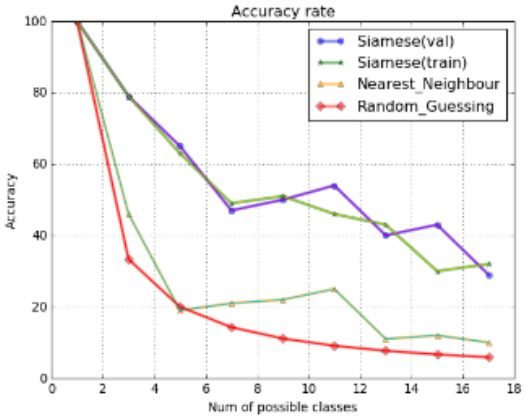
Şekil 3.48. Softplus aktivasyon fonksiyonu, L1-distance ve 5-way TSÖ ile olası sınıf sayısına göre doğruluk oranı



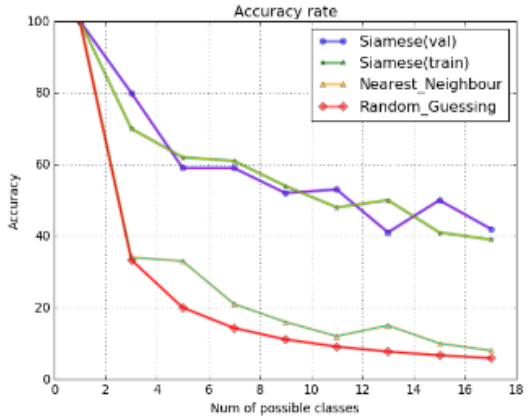
Şekil 3.49. Softplus aktivasyon fonksiyonu, Constrastive fonksiyon ve 5-way TSÖ ile olası sınıf sayısına göre doğruluk oranı



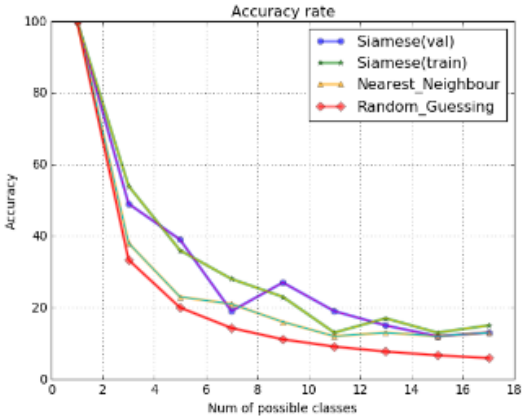
Şekil 3.503. Tanh aktivasyon fonksiyonu, L1-distance ve 5-way TSÖ ile olası sınıf sayısına göre doğruluk oranı



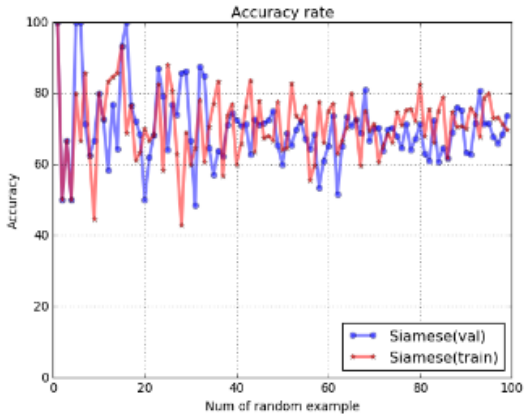
Şekil 3.51. Sigmoid aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile olası sınıf sayısına göre doğruluk oranı



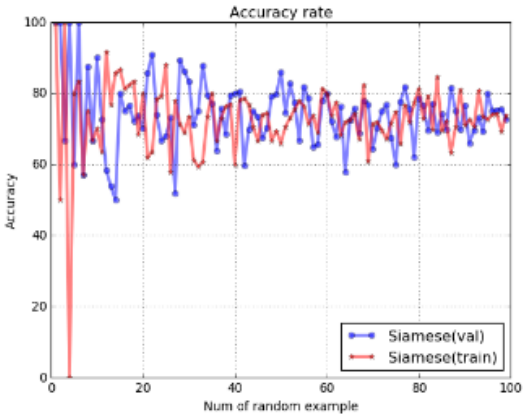
Şekil 3.52. Softplus aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile olası sınıf sayısına göre doğruluk oranı



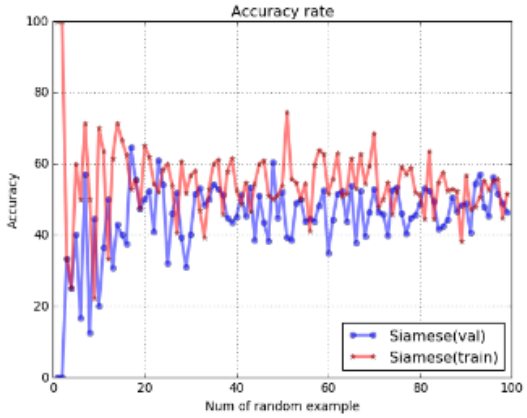
Şekil 3.53. Tanh aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile olası sınıf sayısına göre doğruluk oranı



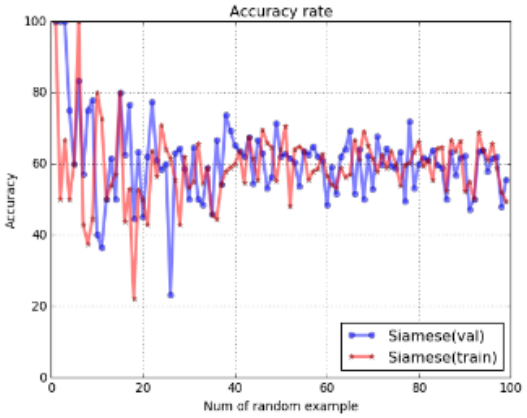
Şekil 3.54. Sigmoid aktivasyon fonksiyonu, L1-distance ve 3-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı



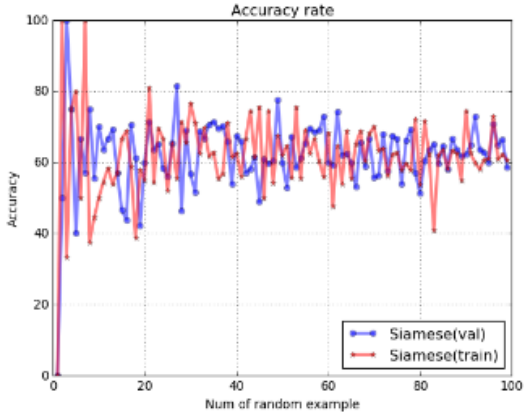
Şekil 3.55. Softplus aktivasyon fonksiyonu, L1-distance ve 3-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı



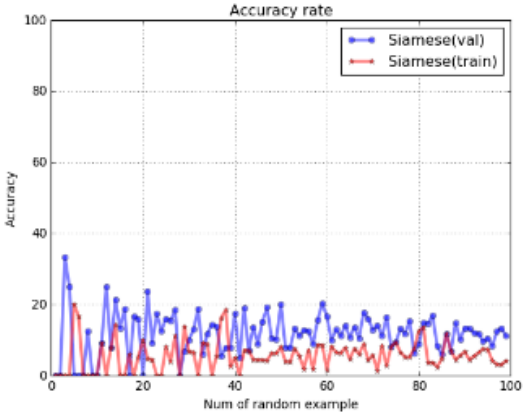
Şekil 3.56. Tanh aktivasyon fonksiyonu, L1-distance ve 3-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı



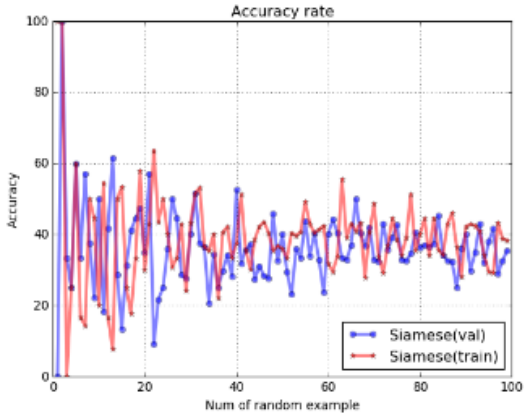
Şekil 3.57. Sigmoid aktivasyon fonksiyonu, L1-distance ve 5-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı



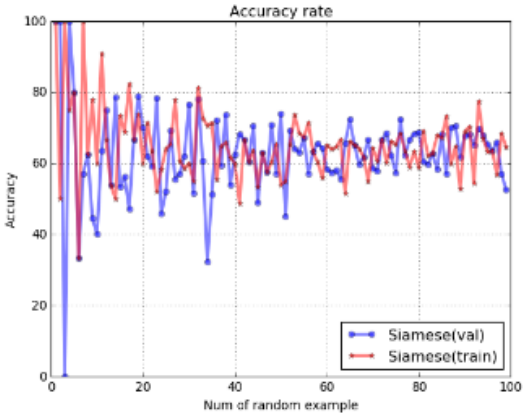
Şekil 3.58. Softplus aktivasyon fonksiyonu, L1-distance ve 5-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı



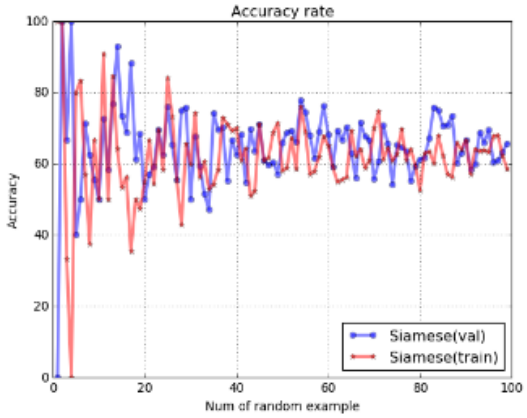
Şekil 3.59. Softplus aktivasyon fonksiyonu, Contrastive fonkisyon ve 5-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı



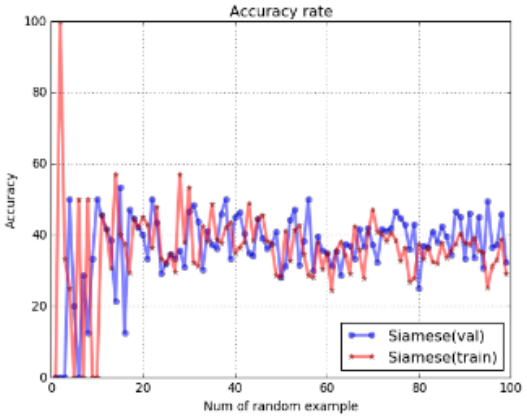
Şekil 3.60. Tanh aktivasyon fonksiyonu, L1-distance ve 5-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı



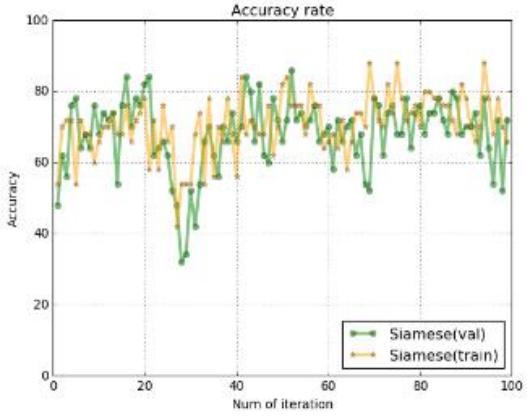
Şekil 3.61. Sigmoid aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı



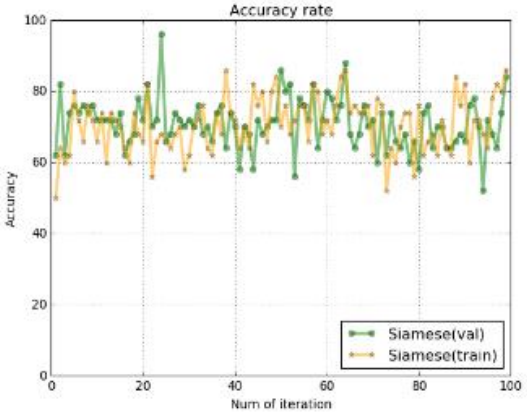
Şekil 3.62. Softplus aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı



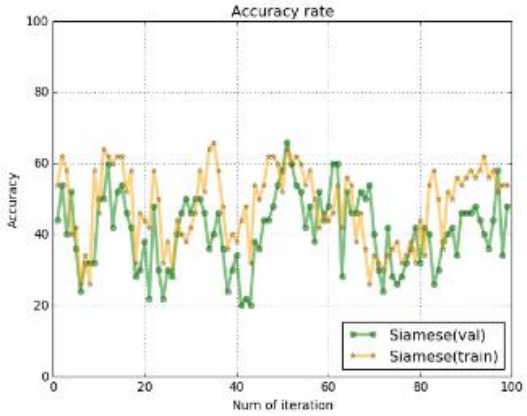
Şekil 3.63. Tanh aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile rastgele örnek sayısına göre doğruluk oranı



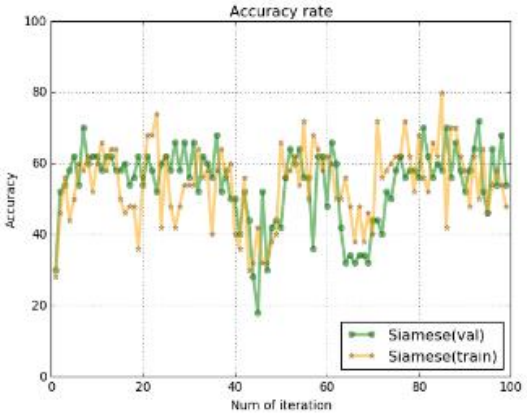
Şekil 3.64. Sigmoid aktivasyon fonksiyonu, L1-distance ve 3.way TSÖ ile iterasyon sayısına göre doğruluk oranı



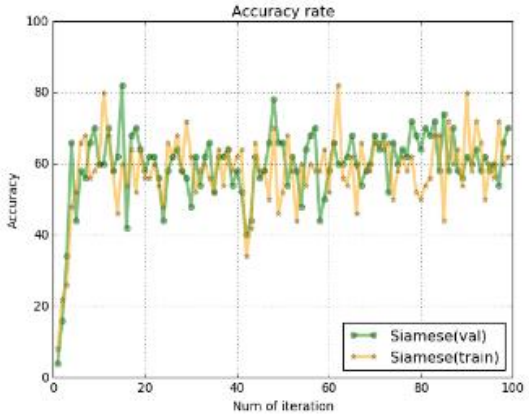
Şekil 3.65. Softplus aktivasyon fonksiyonu, L1-distance ve 3-way TSÖ ile iterasyon sayısına göre doğruluk oranı



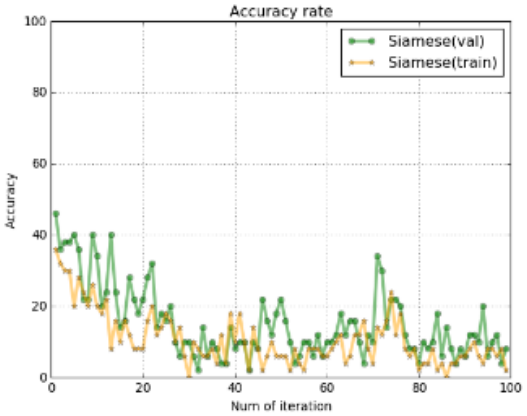
Şekil 3.66. Tanh aktivasyon fonksiyonu, L1-distance ve 3.way TSÖ ile iterasyon sayısına göre doğruluk oranı



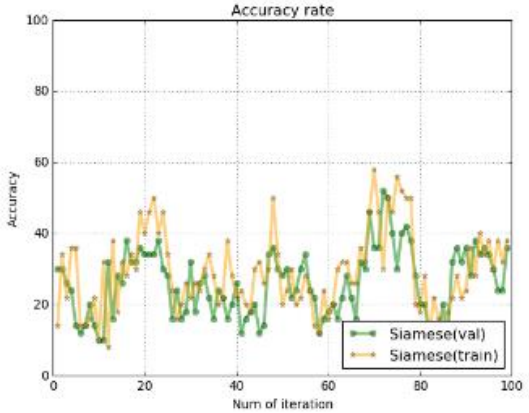
Şekil 3.67. Sigmoid aktivasyon fonksiyonu, L1-distance ve 5-way TSÖ ile iterasyon sayısına göre doğruluk oranı



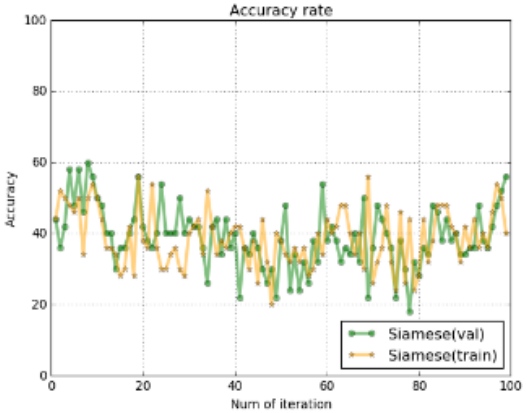
Şekil 3.68. Softplus aktivasyon fonksiyonu, L1-distance ve 5.way TSÖ ile iterasyon sayısına göre doğruluk oranı



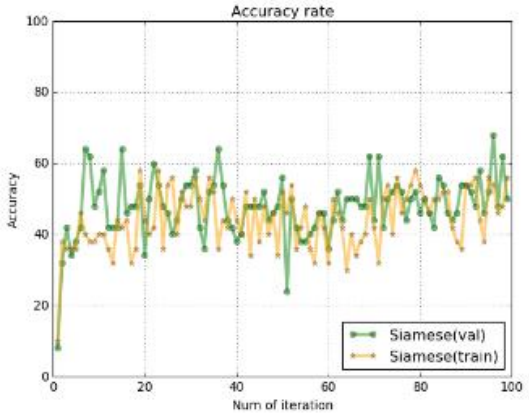
Şekil 3.69. Softplus aktivasyon fonksiyonu, Contrastive fonksiyon ve 5-way TSÖ ile iterasyon sayısına göre doğruluk oranı



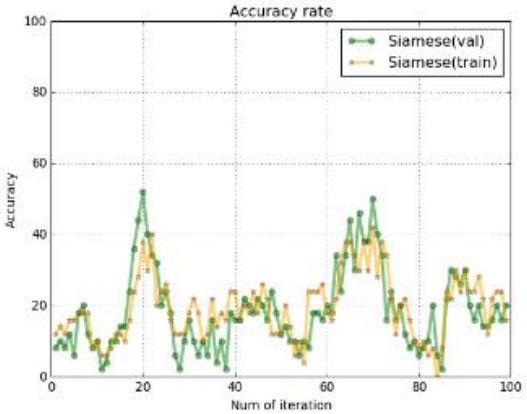
Şekil 3.70. Tanh aktivasyon fonksiyonu, L1-distance ve 5.way TSÖ ile iterasyon sayısına göre doğruluk oranı



Şekil 3.71. Sigmoid aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile iterasyon sayısına göre doğruluk oranı



Şekil 3.72. Softplus aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile iterasyon sayısına göre doğruluk oranı



Şekil 3.73. Tanh aktivasyon fonksiyonu, L1-distance ve 10-way TSÖ ile iterasyon sayısına göre doğruluk oranı

Çalışmamızda, TSÖ ve Siyam KSA ağlarının kullanımı sırasında birçok parametrenin performansa etkisi değerlendirilmiştir. Bunun için TSÖ tekniği 3-way, 5-way ve 10-way olarak uygulanmıştır. Ayrıca kayıp fonksiyonu ve aktivasyon fonksiyonunun performans üzerindeki etkisi de araştırılmıştır. Çalışmamızda, L1-Distance ve Contrastive kayıp fonksiyonları uygulanmıştır. Her iki fonksiyonun da çalışmaya katkısı belirlenmiştir. Son olarak Sigmoid, Tanh ve Softplus aktivasyon fonksiyonlarının performansa olan etkisi araştırılmıştır. Tüm bunlara ek olarak olası sınıf sayısı, rastgele örnek sayısı ve iterasyon sayısı gibi parametrelerin sınıflandırma üzerindeki önemi incelenmiştir. Sınıf sayısına göre Şekil 3.44 ile Şekil 3.53 arasında, rastgele örnek sayısına göre Şekil 3.54 ile Şekil 3.63 arasında, iterasyon sayısına göre ise Şekil 3.64 ile Şekil 3.73 arasında önerilen çalışmanın doğruluk oranı sunulmuştur. Sınıflandırmanın olası sınıf sayısı üzerindeki etkisini incelendiğinde; N-way, kayıp fonksiyon ve aktivasyon fonksiyon parametrelerini dikkate alırsak, ağda sınıflandırılacak sınıf sayısındaki artış ağın doğruluğunda bir azalmaya neden olmaktadır. Sınıflandırılacak sınıf sayısındaki artış ile ağın birbirine oldukça benzeyen bu sınıfları yanlış etiketleyebildiği gözlemlenmiştir. Rastgele örnek sayısının sınıflandırmaya etkisini incelendiğinde, N-way, kayıp fonksiyonu ve aktivasyon fonksiyonu parametrelerini dikkate alındığında örnek görüntü sayısındaki artış ise performansı başlangıçta arttırmakla beraber belli bir eşik değerinden sonra örnek sayısı ne kadar artarsa artsın ağ performansı belli bir aralıkta sabit kalmaktadır. Son olarak iterasyon sayısına göre doğruluk oranları incelendiğinde, bu artışın performans üzerinde belirgin bir farklılık oluşturmadığı anlaşılmıştır.

Bu çalışmanın ana katkıları aşağıda özetlenmiştir:

- Birbirine çok benzeyen nesneler TSÖ, Siyam ağı ve derin öğrenme kullanılarak sınıflandırılmıştır. TSÖ, Siyam ağı ve derin öğrenme kullanan çalışmalardan [53], 5 nesne grubunu sınıflandırmıştır, [49]' da 5 nesne grubu sınıflandırmıştır ve [55] ise 4 nesne grubu sınıflandırılmıştır. Bunlara karşın çalışmamızda 18 nesne grubu sınıflandırılmıştır.
- [53], [49] ve [55] çalışmalarında, sınıflandırma aşamasında sırasıyla sadece Triplet kayıp, sadece L2-mesafe ve sadece Öklid kaybı ile kullanılmıştır. Çalışmamızda ise L1-mesafe ve Contrastive kayıp fonksiyonları test edilmiş ve elde edilen deneysel sonuçlar karşılaştırmalı olarak sunulmuştur.
- [50], [55] ve [117] çalışmalarında, N-way TSÖ' deki N parametresini çoklu değerler üzerinde denemiştir. Çalışmamızda ise N- parametresine ek olarak rastgele örnek sayısı ve iterasyon sayısı gibi parametreler de dikkate alınarak optimum sonuçlar elde edilmesi amaçlanmıştır.
- Çalışmamızda diğer çalışmalardan farklı olarak deneysel çalışmalarımızda optimum sonuçları elde etmek için Sigmoid, Softplus ve Tanh aktivasyon fonksiyonları ile L1-mesafe ve Contrastive kayıp fonksiyonları uygulanmıştır. Ayrıca çalışmamızda sınıflandırma performansı incelenirken olası sınıf sayısı, rastgele örnek sayısı ve iterasyon sayısının doğruluk oranına etkisi de araştırılmıştır.

3.6. Bağlantı Elemanlarının ve Tekstil Makinesi Yedek Parçalarının BSÖ Tabanlı İlişki Ağı ile Sınıflandırılması Sonuçları

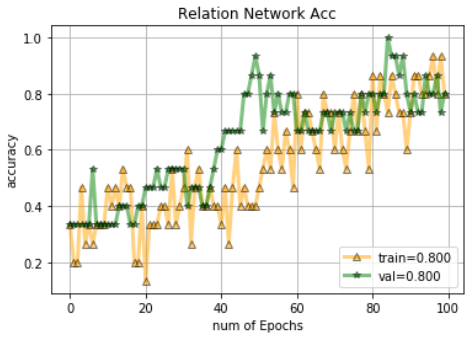
Tezin bu bölümünde, derin metrik öğrenme ve BSÖ tabanlı yeni bir ilişki ağı oluşturulmuştur. Önerilen ağ modeli, *VS_Baglanti* ve *VS_Nit2* adındaki kendi oluşturduğumuz iki farklı veri seti üzerinde test edilmiştir. Bu veri setlerinin temel özelliği, nesnelerin birbirine olan görsel benzerliğidir. Oluşturulan ağ, nesneler arasındaki benzerlik puanını toplayarak bu nesnelerin sınıflandırılmasını kolaylaştırmıştır. Önerdiğimiz yeni ilişki ağ modeli ile literatürdeki geleneksel ilişki ağ modelinin performansı arasında detaylı olarak karşılaştırmalar yapılmıştır.

Çalışmamızda, başlangıçta her sınıf için çok fazla veri olmadığından BSÖ kullanılmıştır. N-way K-shot sınıflandırmasında, N ve K parametreleri çeşitli şekillerde değiştirilmiştir ve bu değişikliklerin, modelin performansı üzerindeki etkileri araştırılmıştır. Bu çalışmadaki hiper-parametreler Tablo 3.25'deki gibi ayarlanmıştır.

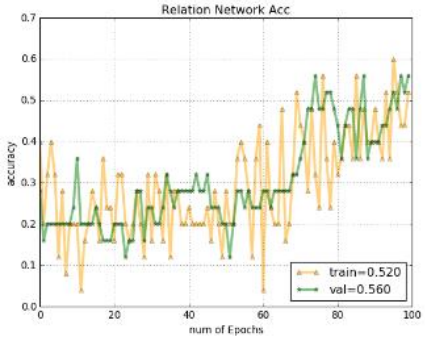
Tablo 3.25. İlişki ağı modelindeki hiper-parametre değerleri

Hiper-parametre	**Değeri**
Optimizsayon algoritması	Adam
İterasyon	100
Parti-boyutu	32
Öğrenme oranı	0.001
Giriş görüntü boyutu	150x150

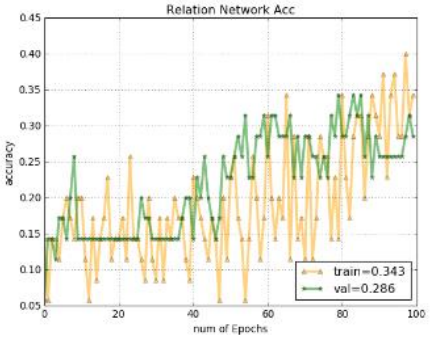
Bu çalışmada, öncelikle BSÖ tabanlı veri setleri ile önerilen ilişki ağ modeli sınıflandırılması yapılmıştır. N-way K-shot sınıflandırılmasında, N-sınıf sayısı ve K-sınıflardaki örnek sayısı değerlerinin N=3, 5, 7 ve 10 ile K=5 ve 10 arasında farklı kombinasyonlar uygulanmıştır. Sınıf sayısı ve her sınıfa ait örnek sayısı değerlerindeki değişimin performansa etkisi saptanmıştır. Önerilen GMRNet modelinin N-way K-shot sınıflandırma sonuçları, *VS_Baglanti* verisetinde Şekil 3.74' te ve *VS_Nit2* verisetinde Şekil 3.75' te sunulmuştur. Önerilen GMRNet modelinin N-way K-shot sınıflandırma sonuçları Tablo 3.26' da gösterilmiştir. GMRNet modelinin veri setlerindeki N-way K-shot sınıflandırma yüzdelik oranları Şekil 3.76' da gösteriliştir.



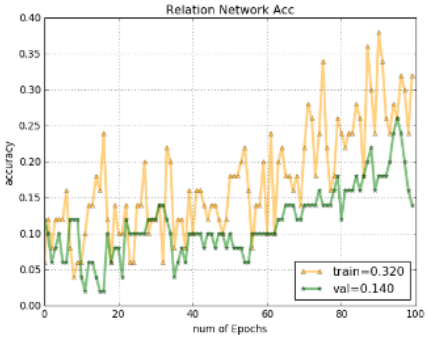
(a) 3-way 5-shot



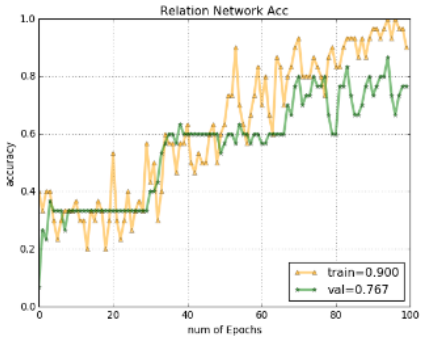
(b) 5-way 5-shot



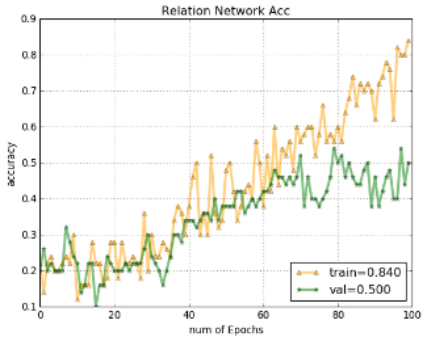
(c) 7-way 5-shot



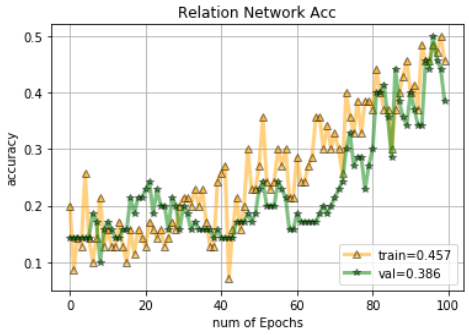
(d) 10-way 5-shot



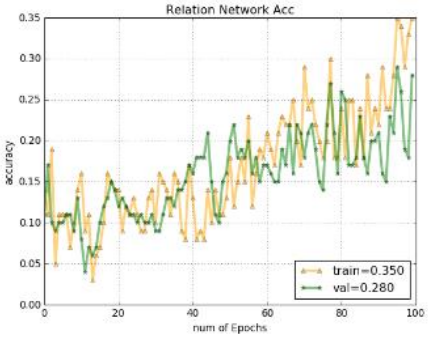
(e) 3-way 10-shot



(f) 5-way 10-shot

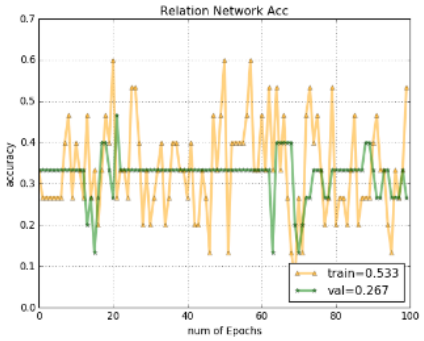


(g) 7-way 10-shot

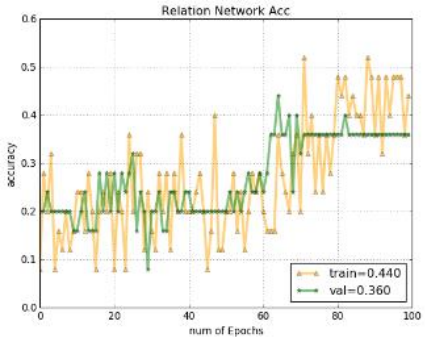


(h) 10-way 10-shot

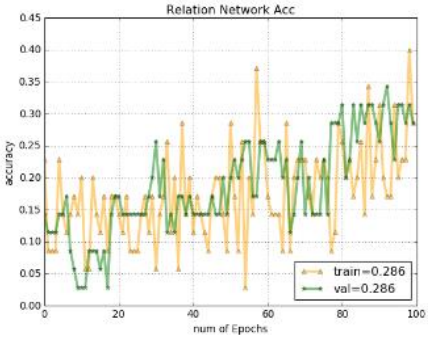
Şekil 3.74. Önerilen GMRNet modelinin *VS_Baglanti* verisetinde N-way K-shot sınıflandırma sonuçları



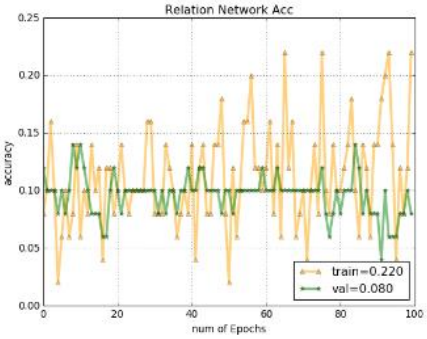
(a) 3-way 5-shot



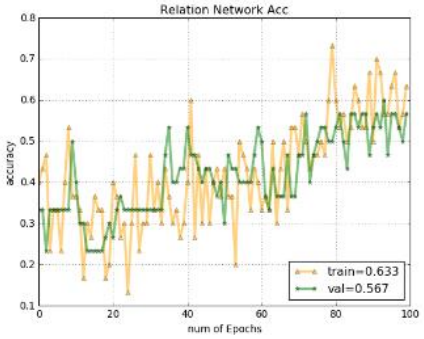
(b) 5-way 5-shot



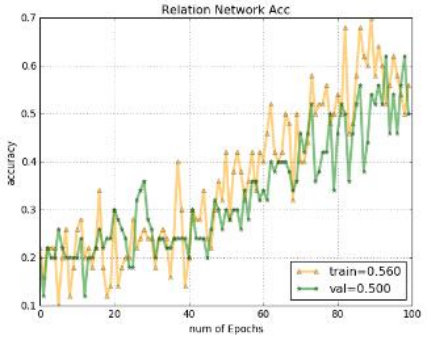
(c) 7-way 5-shot



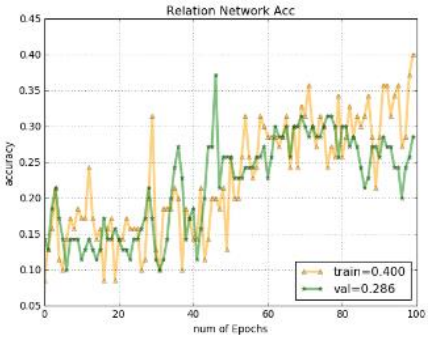
(d) 10-way 5-shot



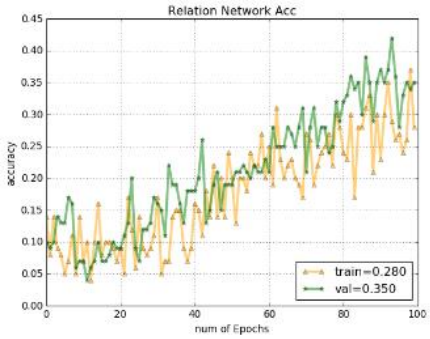
(e) 3-way 10-shot



(f) 5-way 10-shot



(g) 7-way 10-shot

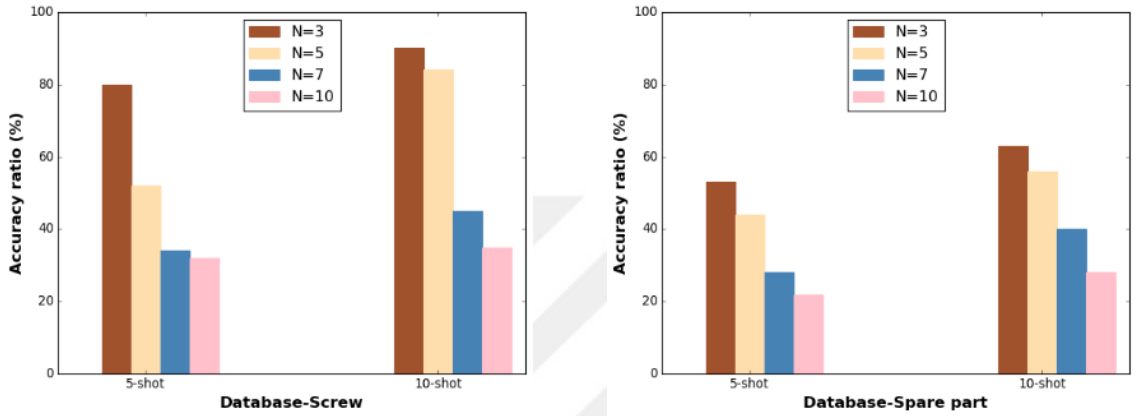


(h) 10-way 10-shot

Şekil 3.75. Önerilen GMRNet modelinin *VS_Nit2* verisetinde N-way K-shot sınıflandırma sonuçları

Tablo 3.26. Önerilen GMRNet modelinin N-way K-shot sınıflandırma sonuçları

N	VS_Baglanti Doğruluk oranı (%)		VS_Nit2 Doğruluk oranı (%)	
	5 shot	10-shot	5-shot	10-shot
N=3	80	90	53	63
N=5	52	84	44	56
N=7	34	45	28	40
N=10	32	35	22	28

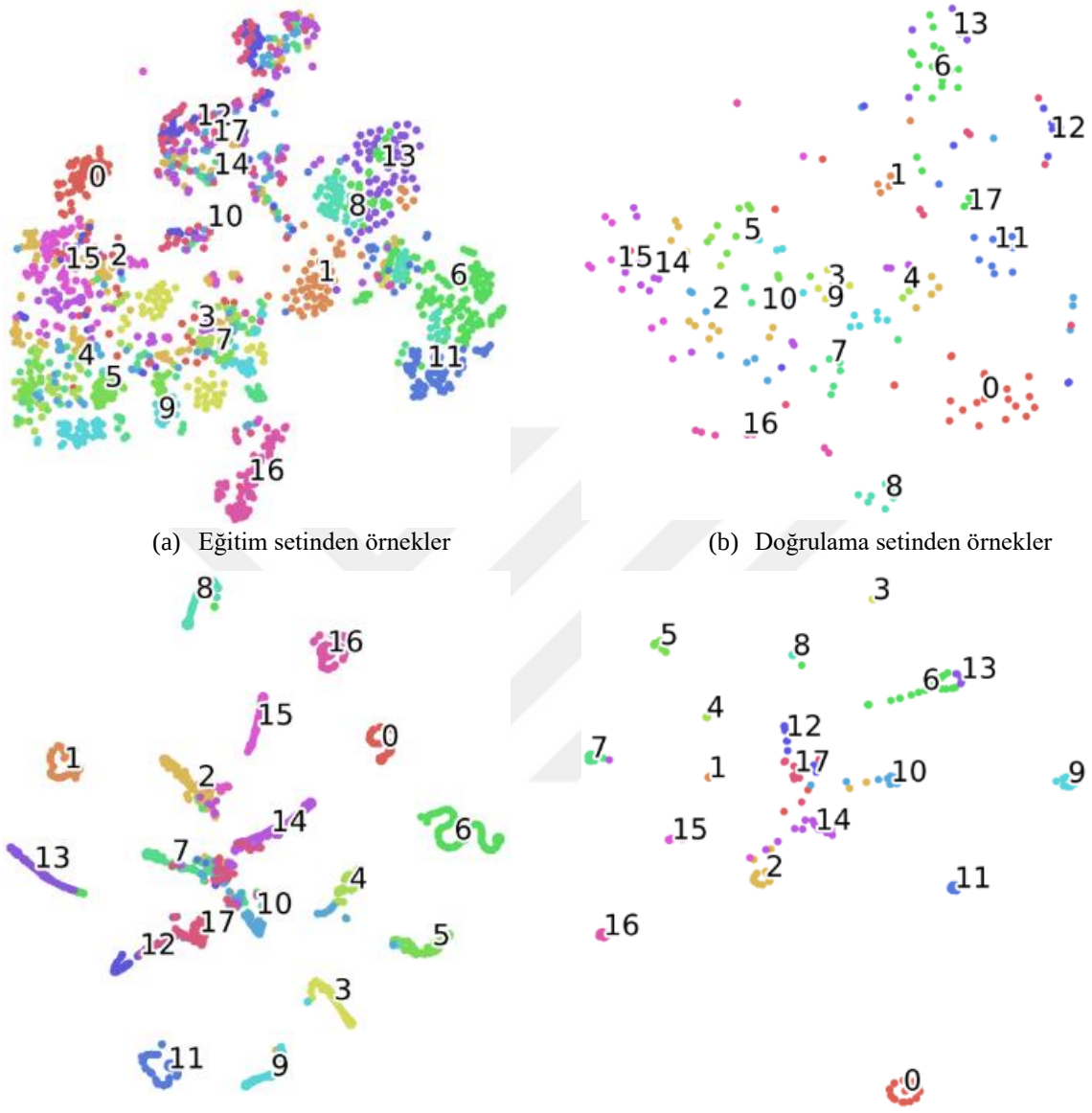


(a) *VS_Baglanti* (b) *VS_Nit2*

Şekil 3.76. GMRNet modelinin veri setlerindeki N-way K-shot sınıflandırma yüzde oranları

Tablo 3.26 incelendiğinde, N-sınıf sayısı arttıkça doğruluk oranının azaldığı, K-örnek sayısı arttıkça ise doğruluk oranının arttığı gözlemlenmiştir. Veri setimizdeki nesneler oldukça benzer olduğundan ve örnek sayısı sınırlı olduğundan, N-sınıf sayısı arttıkça doğruluk oranı azalmıştır. Benzer nesneleri yüksek performansla sınıflandırmak için veri setimiz görüntü iyileştirme teknikleri ile arttırılmıştır. Her iki veri setindeki görüntü sayısı arttırıldıktan sonra literatürdeki ilişki ağı modeli ve tasarladığımız ilişki ağı modeli üzerinde veri setleri uygulanarak yeni veri setinin performansa etkisi gözlemlenmiştir. Veri arttırma tekniklerinin uygulanması ile elde edilen yeni veri setlerinin adları sırasıyla *VS_Baglanti_Arttir* ve *VS_Nit2_Arttir*' dir. *VS_Baglanti_Arttir* veri setinin literatürdeki standart RN modeline uygulanmasıyla elde edilen t-SNE görüntüleri Şekil 3.77' de gösterlmiştir. *VS_Baglanti_Arttir* veri setinin GMRNet modeline uygulanmasıyla elde edilen t-SNE görüntüleri Şekil 3.78' de gösterilmiştir. VS_Baglanti_Arttir veri setinin standart ve önerilen İlişki ağlarına uygulanmasıyla elde edilen doğruluk oranları Şekil 3.79' da gösterilmiştir. *VS_Nit2_Arttir* veri setinin literatürdeki standart RN modeline uygulanmasıyla elde edilen t-SNE görüntüleri Şekil 3.80' de sunulmuştur. *VS_Nit2_Arttir* veri setinin GMRNet modeline uygulanmasıyla elde edilen t-SNE görüntüleri Şekil 3.81' de gösterilmiştir. *VS_Nit2_Arttir* veri setinin standart ve önerilen İlişki ağlarına uygulanmasıyla elde edilen doğruluk oranları Şekil 3.82' de sunulmuştur. Literatürdeki standart İlişki ağı modeli ile önerilen ilişki ağ modelinin performans karşılaştırılması Tablo 3.27' de sunulmuştur.

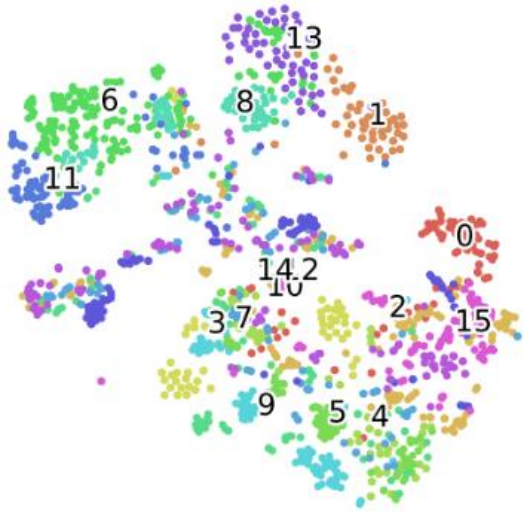
Literatürdeki derin metrik öğrenme tabanlı çalışmalarla bizim çalışmanın karşılaştırılması Tablo 3.28' de gösterilmiştir.



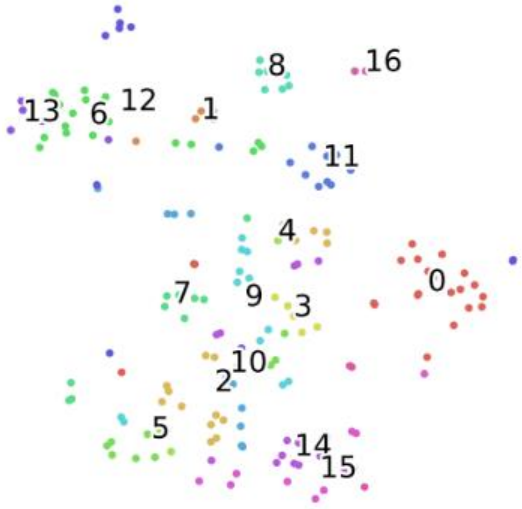
(a) Eğitim setinden örnekler (b) Doğrulama setinden örnekler

(c) İlişki ağı uygulanınca eğitim verisi (d) İlişki ağı uygulanınca doğrulama verisi

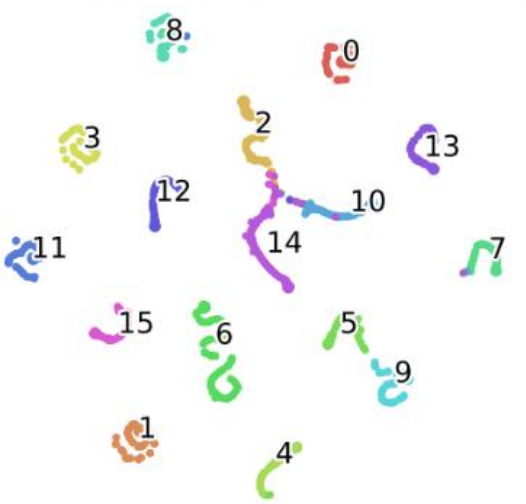
Şekil 3.77. *VS_Baglanti_Arttir* veri setinin literatürdeki standart RN modeline uygulanmasıyla elde edilen t-SNE görüntüleri.



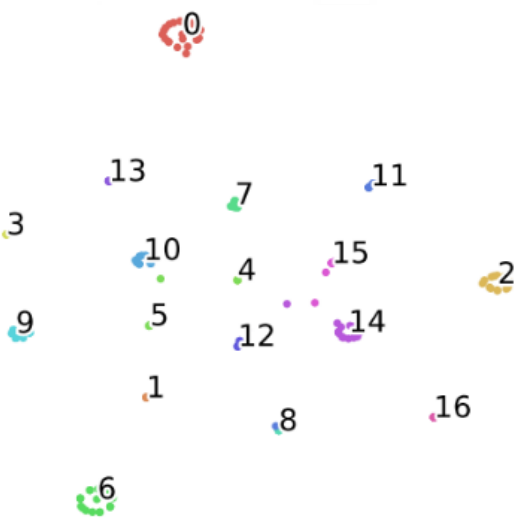
(a) Eğitim setinden örnekler



(b) Doğrulama setinden örnekler

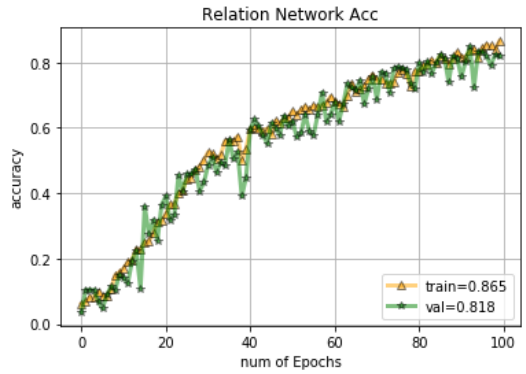


(c) İlişki ağı uygulanınca eğitim verisi

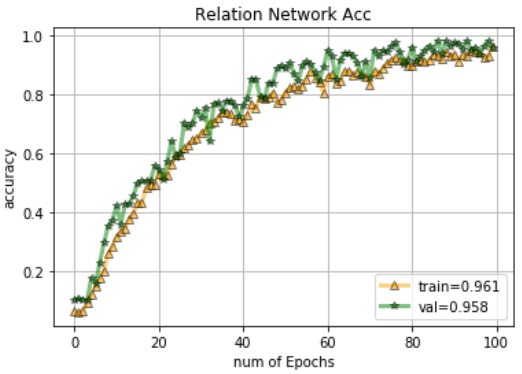


(d) İlişki ağı uygulanınca doğrulama verisi

Şekil 3.78. *VS_Baglanti_Arttir* veri setinin GMRNet modeline uygulanmasıyla elde edilen t-SNE görüntüleri.

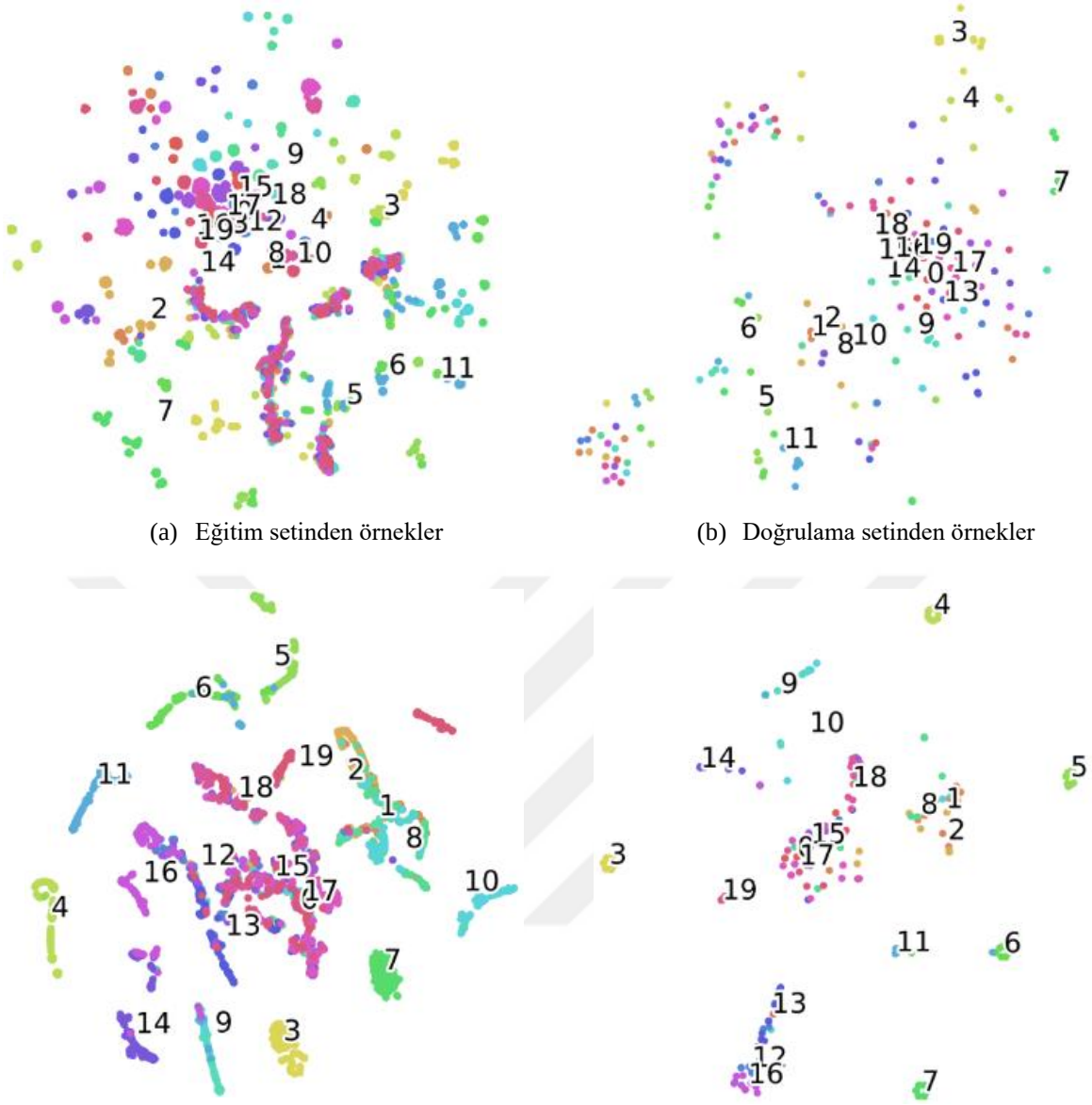


(a) Literatürdeki standart İlişki modeli



(b) GMRNet modeli

Şekil 3.79. *VS_Baglanti_Arttir* veri setinin standart ve önerilen İlişki ağlarına uygulanmasıyla elde edilen doğruluk oranları



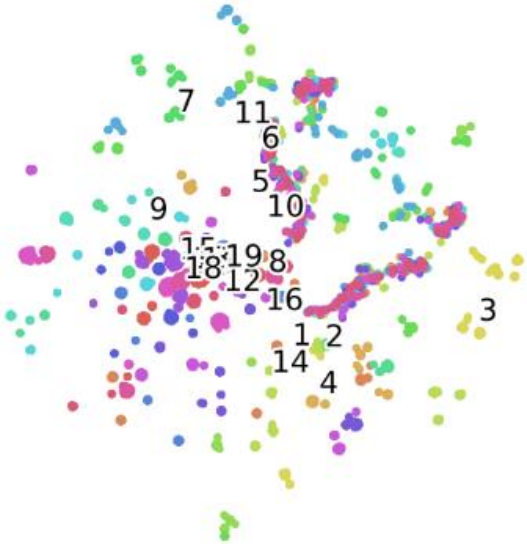
(a) Eğitim setinden örnekler

(b) Doğrulama setinden örnekler

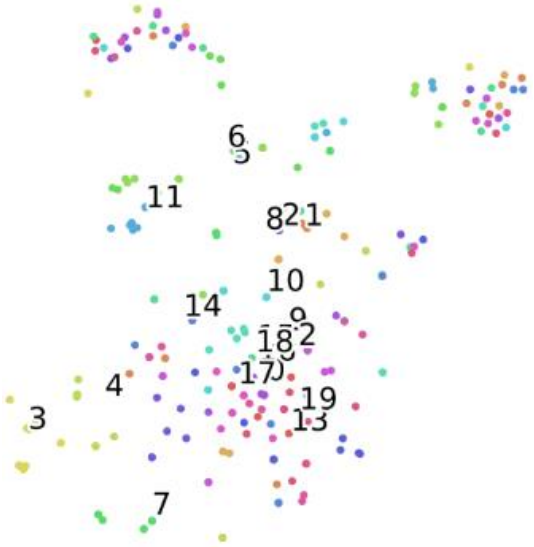
(c) İlişki ağı uygulanınca eğitim verisi

(d) İlişki ağı uygulanınca doğrulama verisi

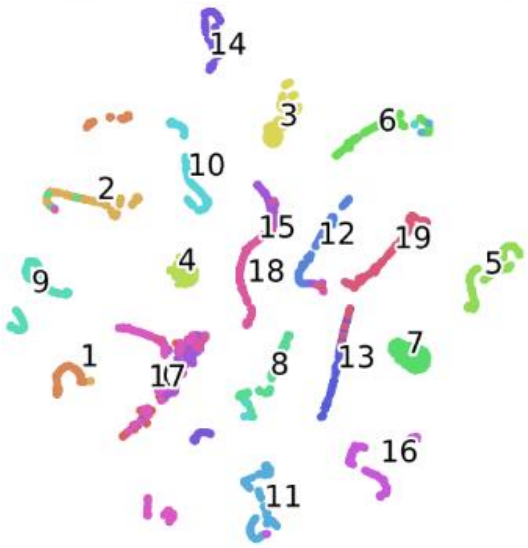
Şekil 3.80. *VS_Nit2_Arttir* veri setinin literatürdeki standart RN modeline uygulanmasıyla elde edilen t-SNE görüntüleri.



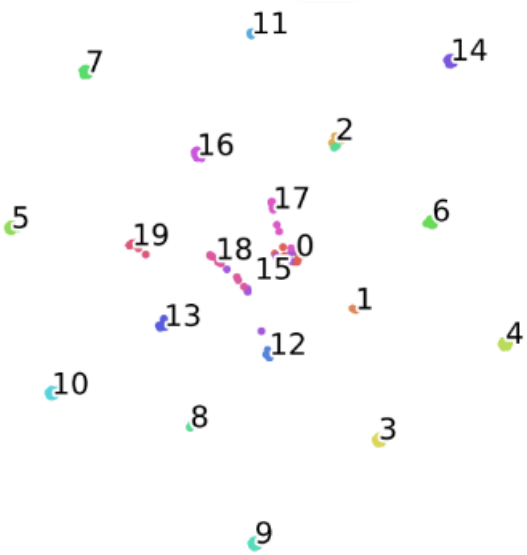
(a) Eğitim setinden örnekler



(b) Doğrulama setinden örnekler

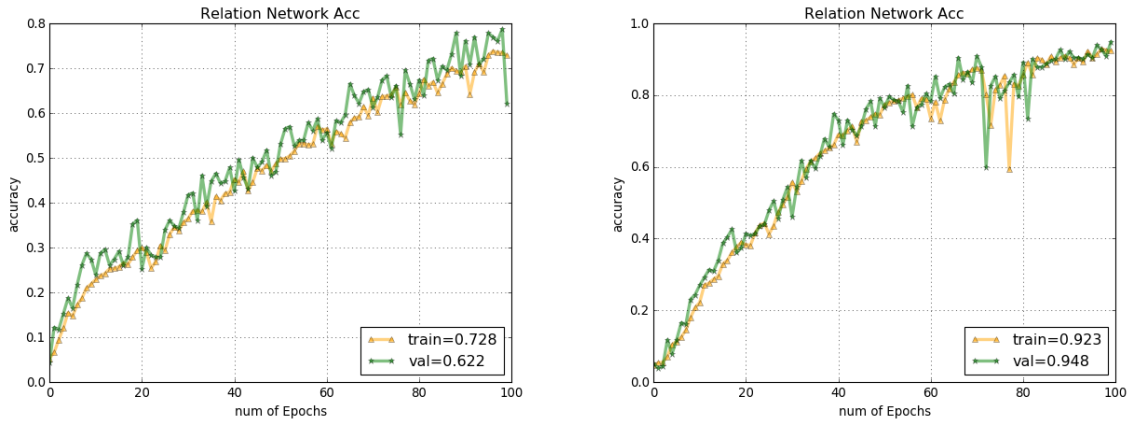


(c) İlişki ağı uygulanınca eğitim verisi



(d) İlişki ağı uygulanınca doğrulama verisi

Şekil 3.81. *VS_Nit2_Arttir* veri setinin GMRNet modeline uygulanmasıyla elde edilen t-SNE görüntüleri.



(a) Literatürdeki standart İlişki modeli (b) GMRNet modeli

Şekil 3.82. *VS_Nit2_Arttir* veri setinin standart ve önerilen İlişki ağlarına uygulanmasıyla elde edilen doğruluk oranları

Tablo 3.27. Literatürdeki standart İlişki ağı modeli ile önerilen ilişki ağ modelinin performans karşılaştırılması

Veri seti	Literatürdeki standart ilişki modeli	Önerilen ilişki ağı GMRNet
VS_Baglanti_Arttir	86.50	96.10
VS_Nit2_Arttir	72.80	92.30

Tablo 3.28. Literatürdeki derin metrik öğrenme tabanlı çalışmalarla bizim çalışmanın karşılaştırılması

Referans	Metot	Uygulanan problem	Doğruluk oranı (%)
[62]	VGG16	Görüntü segmentasyonu	45.80
[63]	GoogLeNet	Kişi tanımlama	63.66
[65]	CNN	Yüz ve akrabalık doğrulaması	98.00
[66]	Üçlü Ağ	Denizcilik gemisi tanımlama	68.00
[67]	ResNet50 ile Üçlü Kayıp	Etiketli açıklama verileri	81.20
[68]	Derin Metrik Ağ	Çok-etiketli görüntü sınıflandırması	92.85
[69]	İlişki ağı	Görüntü sınıflandırma	65.32
[70]	İlişki ağı	Görüntü sınıflandırma	51.80
[71]	İlişki ağı	Meme kanseri sınıflandırması	86.29
[72]	İlişki ağı	Kişi tanımlama	88.90
[73]	İlişki ağı	Hiperspektral görüntü sınıflandırması	98.03
Bizim çalışma	İlişki ağı	Benzer nesnelerin sınıflandırılması	96.10 ve 92.30

Bu çalışmanın performans çıktıları incelendiğinde, N-way K-shot öğrenme ile veri setimizdeki sınıf sayısı arttıkça başarı oranının düştüğü fark edilmiştir. Geliştirilen model, verileri daha az sınıfla daha kolay sınıflandırmaktadır. Ayrıca, veri setimizdeki örnek sayısının fazla olması ile modelin daha

etkili bir şekilde öğrenebildiği keşfedilmiştir. Bu çalışmada, N=3, 5, 7 ve 10 ve K=5 ve 10 değerleri üzerinde denemeler yapılmıştır. Böylece detaylı deneysel sonuçlar elde edilmiştir.

Tezin bu bölümündeki çalışmada, yeni bir geometrik ortalama tabanlı ilişki ağ modeli geliştirilmiştir. Önerilen çalışmada, standart ilişki ağından farklı olarak geliştirilen modele eğilim modülü olarak adlandırdığımız yeni bir modül eklenmiştir. Elde edilen ağa GMRNet adı verilmiştir. Geliştirilen GMRNet, veri artırma işleminden sonra veri setlerimiz üzerinde de test edilmiştir. t-SEN görselleri incelendiğinde, GMRNet veri noktalarını normal bir ilişki ağından daha başarılı bir şekilde sınıfları birbirinden ayırdığı tespit edilmiştir. Bu çalışma için kullandığımız veri setleri, endüstriyel süreçlerdeki gerçek görüntülerini de içermektedir.

Tezin bu bölümdeki çalışmanın ana katkıları şunlardır:

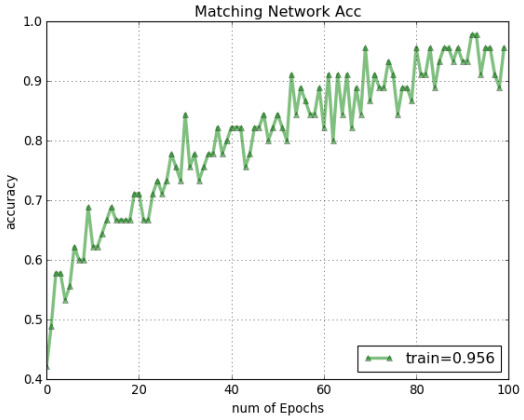
- GMRNet, yüksek derecede benzerliğe sahip nesneleri sınıflandırmaya yöneliktir. Bu çalışmada tasarlanan ağ, yeni ve etkilidir. Performansı açısından geliştirilen ağ, [9-12]' den daha başarılı olduğu keşfedilmiştir.
- Tasarlanan yeni ağın performansını test etmek için iki farklı gerçek veri seti oluşturulmuştur. Her iki veri seti de gerçek dünya endüstri verilerinden oluşturulmuştur. Literatürdeki diğer çalışmaların çoğu, önceden var olan hazır veri setlerini kullanmıştır.
- Çalışmamızda üretilen veri setlerinde, birbirine çok benzeyen nesneleri ayırt etmek endüstriyel süreçler için pratik bir çözüm sağlamaktadır. Kırılmış, paslanmış veya eskimiş yedek parça, vida veya cıvata gibi nesnelerin benzerleriyle karıştırılmadan sınıflandırılması bu süreçler için fayda sağlamaktadır. Ayrıca geliştirilen bu yöntem, birçok çeşitteki nesne türlerine de kolayca uyarlanabilmektedir.
- Bu çalışmada, standart bir ilişki ağına eğilim modülü eklenmiştir. Bu modülün eklenmesiyle oluşan mimari yenidir. Yeni ilişki modeli GMRNet'in performansı, *VS_Baglanti* ve *VS_Nit2* veri setlerinde sırasıyla %86,5' ten %96,1' e ve %72.8' den %92,3' e yükselmiştir. Sonuç olarak, standart ilişki ağ modeline kıyasla iki veri setinde doğruluk oranı %10 ve %20 oranında artmıştır.

3.7. Bağlantı Elemanlarının Karşılaştırma Ağı ve Prototipik Ağı ile BSÖ Tabanlı Sınıflandırılma Sonuçları

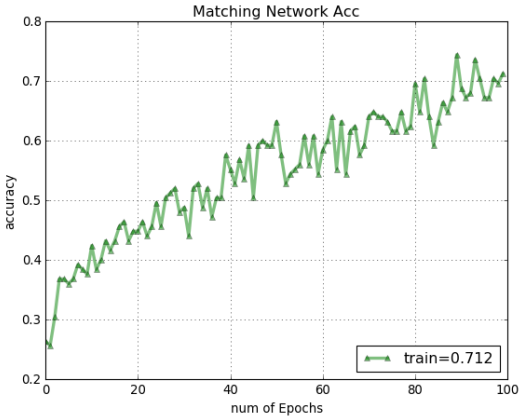
Tezin bu bölümünde, meta-öğrenme algoritmalarından karşılaştırma ağı ve prototipik ağı kullanılarak bağlantı elemanı sınıflandırması yapılmıştır. Her iki model de temelde aynı çalışma sistematiğine sahiptir. Ancak eğitim seti ile sorgu setinin karşılaştırma aşamasında, iki ağ birbirinden farklılık göstermektedir. Bu çalışmanın amacı, aynı veri seti üzerinde iki ağın performansını incelemektir. N-way K-shot öğrenme sınıflandırmasının uygulanması aşamasında N=3, 5, 7 ve 10 sınıf sayısı ve K=5 ve 10 sınıftaki örnek sayısı üzerinde değerlendirmeler yapılmıştır. Bu çalışmada kullanılan veri seti, *VS_Baglanti*' dir. Kullanılan hiper-parametre değerleri Tablo 3.29' da gösterilmiştir.

Tablo 3.29. Karşılaştırma ve Prototipik ağında ortak kullanılan hiper-parametre değerleri

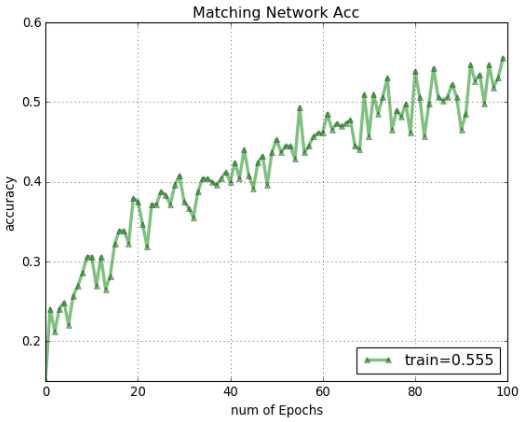
Hiper-parametre	Değeri
Optimizsayon algoritması	Adam
İterasyon	100
Parti-boyutu	32
Öğrenme oranı	0.001
Giriş görüntü boyutu	150x150



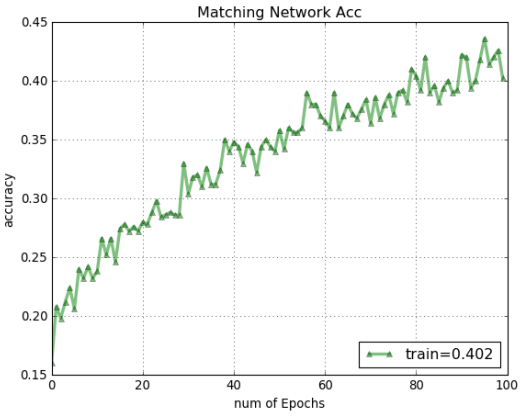
(a) 3-way 5-shot



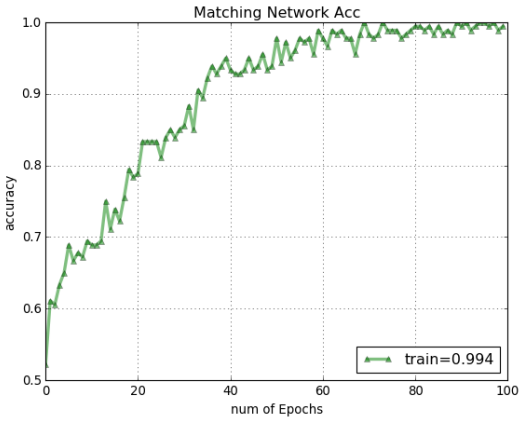
(b) 5-way 5-shot



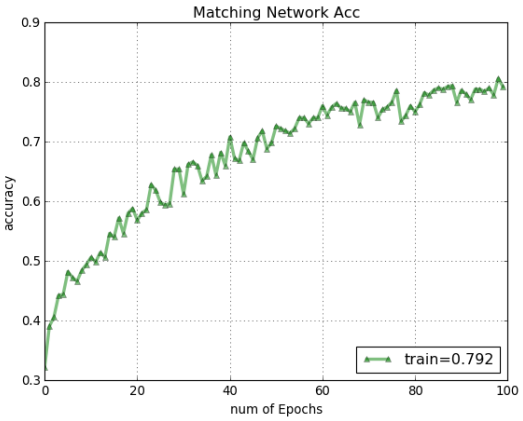
(c) 7-way 5-shot



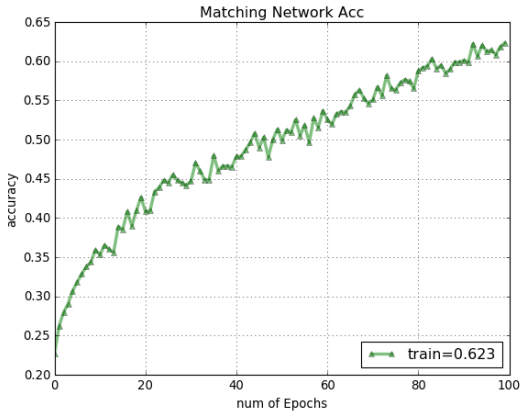
(d) 10-way 5-shot



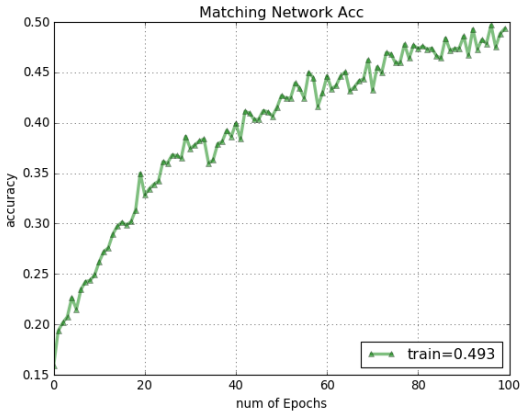
(e) 3-way 10-shot



(f) 5-way 10-shot

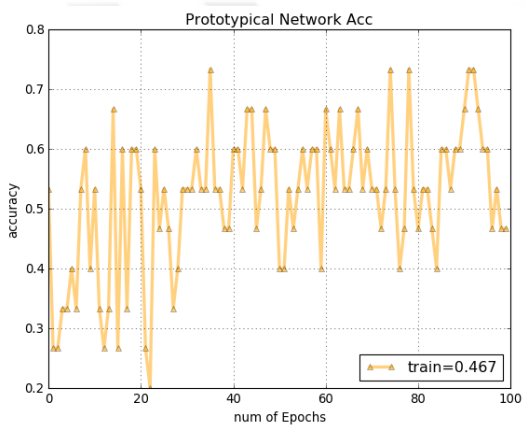


(g) 7-way 10-shot

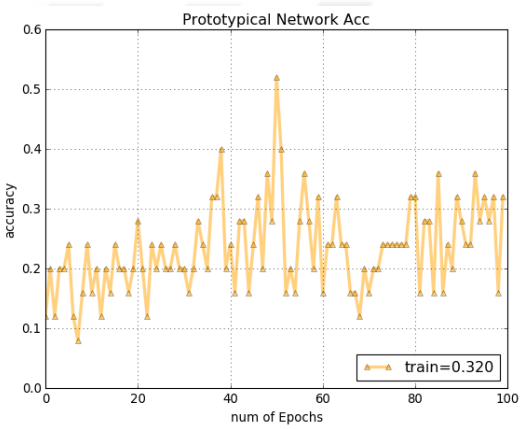


(h) 10-way 10-shot

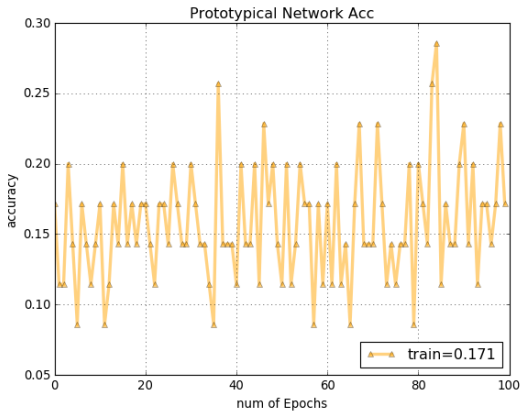
Şekil 3.83. Önerilen Karşılaştırma modelinin N-way K-shot sınıflandırma sonuçları



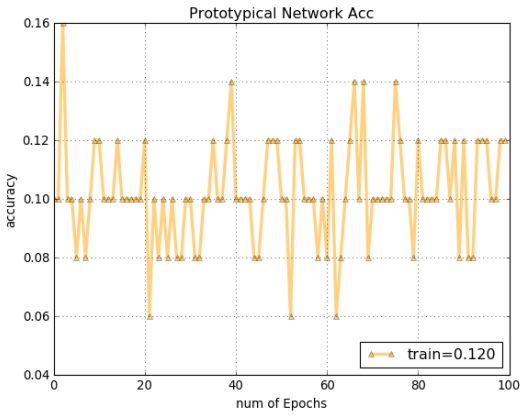
(a) 3-way 5-shot



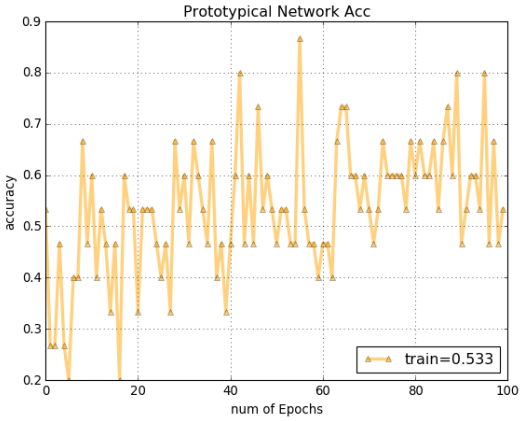
(b) 5-way 5-shot



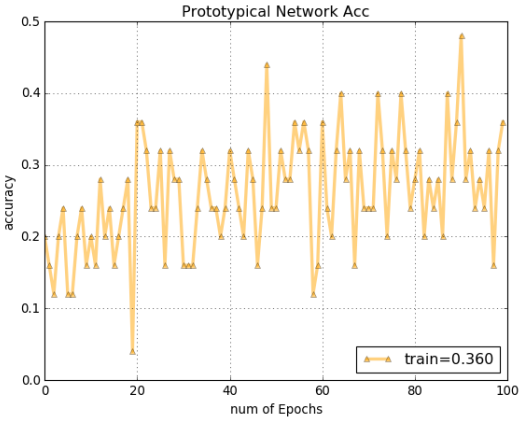
(c) 7-way 5-shot



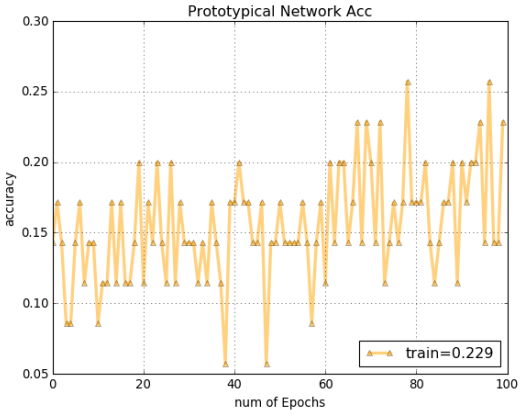
(d) 10-way 5-shot



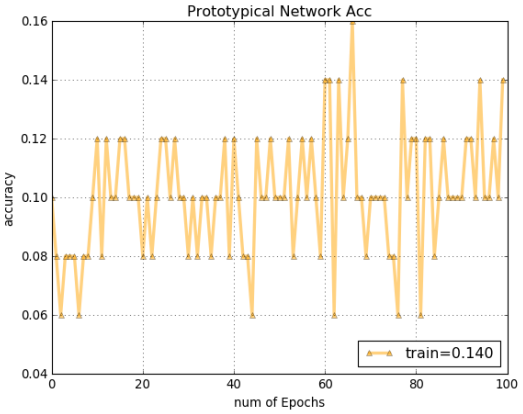
(e) 3-way 10-shot



(f) 5-way 10-shot



(g) 7-way 10-shot

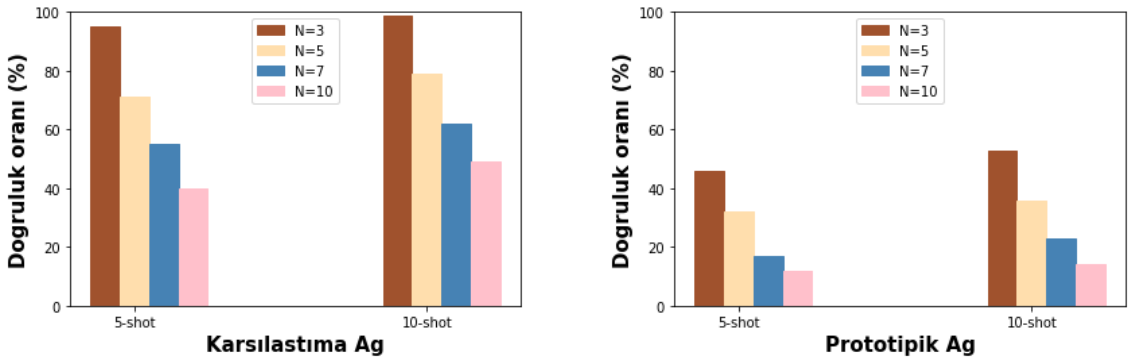


(h) 10-way 10-shot

Şekil 3.84. Önerilen Prototipik modelinin N-way K-shot sınıflandırma sonuçları

Tablo 3.30. Önerilen Karşılaştırma ağı ve Prototipik ağın N-way K-shot sınıflandırma sonuçları

N	Karşılaştırma Ağın Doğruluk oranı (%)		Prototipik Ağın Doğruluk oranı (%)	
	5 shot	**10-shot**	**5-shot**	**10-shot**
N=3	95	99	46	53
N=5	71	79	32	36
N=7	55	62	17	23
N=10	40	49	12	14



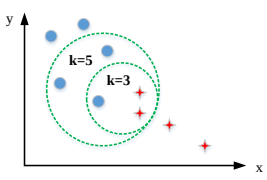
Şekil 3.85. Karşılaştırma ağın ve Prototipik ağın N-way K-shot sınıflandırma yüzde oranları

N-way K-shot sınıflandırma için önerilen Karşılaştırma ağ sonuçları Şekil 3.83' te ve önerilen Prototipik ağ sonuçları Şekil 3.84' te sunulmuştur. Karşılaştırma ağın ve Prototipik ağın N-way K-shot sınıflandırma yüzde oranları Şekil 3.85' te gösterilmiştir. Önerilen Karşılaştırma ağı ve Prototipik ağın N-way K-shot sınıflandırma sonuçları Tablo 3.30' da gösterlmiştir. Tablo 3.30 incelendiğinde, N-sınıf sayısı arttıkça doğruluk oranının azaldığı, K-örnek sayısı arttıkça ise doğruluk oranının arttığı gözlemlenmiştir. Veri setimizdeki nesneler oldukça benzer olduğundan ve örnek sayısı sınırlı olduğundan, N-sınıf sayısı arttıkça doğruluk oranı azalmıştır. Ayrıca aynı KSA mimarisini ve parametrelerini kullanmalarına rağmen karşılaştırma ağının, prototipik ağına göre daha yüksek başarı gösterdiği fark edilmiştir. Bunun nedeni ise prototipik ağ, eğitim setinde her sınıfa ait olan ve o sınıfı temsil eden prototiplerle işlem yaptığından, prototipik işlemi için aritmetik ortalama sonucu elde edilen referans görüntüsünün yeterli seviyede olmamasından ve veri setimizdeki nesnelerin çok benzer olmasından olabilmektedir.

3.8. Makine Öğrenmesi Sınıflandırma Yaklaşımları ile Tezdeki Yaklaşımların Karşılaştırılması

3.8.1. K-En Yakın Komşu Algoritması

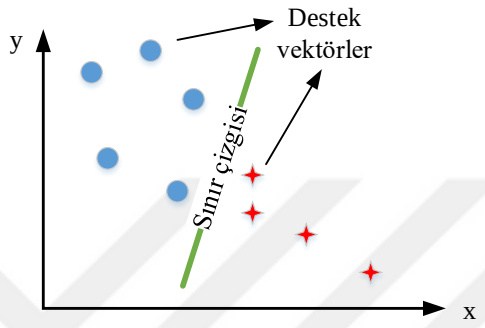
K-en yakın komşu (k-NN) algoritması, makine öğrenmesi tabanlı sınıflandırma yaklaşımlarından birisidir. Sınıf bilgisinin belli olduğu veri setine, yeni bir verinin eklenmesiyle bu yeni verinin hangi sınıfa ait olduğunun bulunması için kullanılmaktadır [138]. Yeni verinin veri setindeki diğer verilere olan uzaklığını hesaplayarak sınıf bilgisine ulaşmayı sağlamaktadır. Şekil 3.86' da gösterildiği gibi benzer verilerin birbirine yakın olduğu fikri esas alarak çalışmaktadır.



Şekil 3.86. K-NN algoritmasının çalışma prensibi

3.8.2. Destek Vektör Makinesi

Destek vektör makinesi (DVM), istatistiksel tabanlı bir öğrenme yaklaşımıdır. Denetimli bir sınıflandırma yöntemidir. Eğitim aşamasında, eğitim veri setindeki veri ve etiketler arasındaki ilişkiyi öğrenmektedir. Bu ilişkiler için destek vektörlerden oluşmaktadır [139]. DVM, sınıflandırılacak iki grup arasına bir sınır vektörü çizdikten sonra grupları ayırt etmeye başlamaktadır [140]. Şekil 3.87' de gösterildiği gibi bahsedilen sınıf çizgisi, iki sınıfın da elemanlarına en uzak olan nokta olarak seçilmektedir.



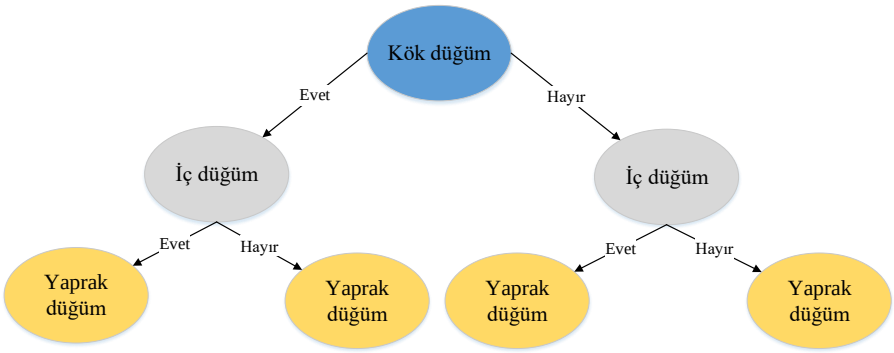
Şekil 3.87. DVM algoritmasının çalışma prensibi

3.8.3. Naive Bayes Algoritması

Naive Bayes (NB) karar mekanizması, sınıflandırma yapılırken oluşan hataları minimuma indirmeyi amaçlamaktadır. Bu işlemi, sınıflandırılacak veri ile ilgili ön-bilgisi olması sayesinde yapabilmektedir. Bu algoritmanın temelinde, olasılık hesaplama bulunmaktadır. Her olasılık hesabında, verileri defalarca taramaktadır [141]. Bu yüzden bu yaklaşım, yüksek maliyet gerektirmektedir. Algoritmanın çalışma prensibi: bir ihtimalin olması durumunda diğer ihtimalin olma olasılığını hesaplamaktadır.

3.8.4. Karar Ağaçları

Karar ağaçları (KA), örüntü sınıflandırmada kullanılan makine öğrenmesi yaklaşımlarından birisidir. Veri setinde birden çok bağımsız değişken ve bir tane bağımlı değişken bulunurken bağımlı değişken kategorik türde olunca sınıflandırma ağacı; sürekli yapıda ise regresyon ağacı oluşmaktadır. Sınıflandırma ağaçlarında, sınıf değeri daha önceden belirlenmektedir. Dallardan ve düğümlerden oluşan ağaçta, yaprak düğüm alt düğümü hiç olmayan noktadır. Yaprak düğümler, sınıf değerleri içermektedir [142]. KA' da amaç: veri özelliklerinden çıkarılan basit kurallarla hedef değişken değerini tahmin etmektedir. KA, anlaşılabilir basit kurallarla sınıflandırma yaptığı için sıklıkla tercih edilmektedir [141]. KA, kendi başına kullanılmanın yanında rastgele orman, gradyan güçlendirme gibi algoritmaları da kullanabilmektedir. KA algoritmasının düğüm yapısı örneği Şekil 3.88' deki gibidir.

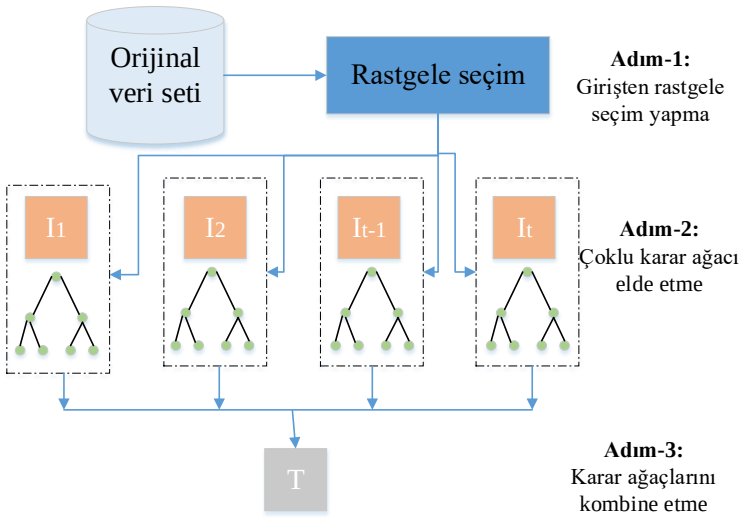


Şekil 3.88. KA algoritmasının düğüm yapısı örneği

3.8.5. Extra Ağaçlar

Extra ağaçlar (EA), aşırı derecede rastgele ağaçlar olarak ta adlandırılmaktadır. Sınıflandırma uygulamalarında, aşırı öğrenmeyi ve fazla uydurma sorunlarını çözmek için birden fazla ağaç oluşturulmaktadır. Özelliklerin rastgele alt kümelerini kullanarak düğümleri bölerek çalışmaktadır. Her ağaç için eğitim örnekleri kümesi kullanılır [143]. Rastgele orman sınıflandırma yönteminden türetilmiştir. Veri setindeki kopya değerler kullanılarak ağ eğitilmiştir [144]. Düğümler dallara ayrılırken karar verme kriteri kullanılmadan rastgele olarak dallanma uygulanmaktadır. Rastgele özellik örneğinin seçimi, birbiriyle ilişkili birden çok karar ağacının oluşmasına yol açmaktadır. EA algoritmasının çalışma prensibi Şekil 3.89' da gösterilmiştir.

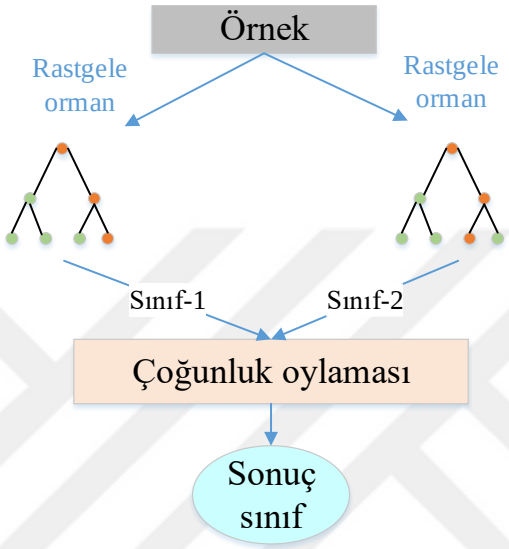
Bazı veri analiz problemlerinde, karmaşıklık ve işlem yükü EA ile azalırken büyük veri kümelerinde yüksek gürültü barındırması durumunda performans düşüklüğü bulunmaktadır. Bu yöntem, istatistiksel açıdan değerlendirildiğinde bias artışı ve varyans düşüklüğüne neden olmaktadır [143].



Şekil 3.89. EA algoritmasının çalışma prensibi

3.8.6. Rastgele Orman Algoritması

Rastgele orman algoritması (RO), yapısında birden fazla KA bulundurmaktadır. Algoritmada kullanılacak olan ağaç sayısı, veri setinden rastgele olarak seçilen veri kümeleri kadardır [139]. Seçilen bu rastgele eğitim setlerinde eğitim gerçekleşmektedir. Her bir KA üzerinde tahmin yapılmaktadır. Tahminler arasından en yüksek üretilen sonuç tercih edilmektedir. RO algoritmasının çalışma prensibi, Şekil 3.90' da gösterilmiştir.



Şekil 3.90. RO algoritmasının çalışma prensibi

3.8.7. Gradyan Arttırma Algoritması

Gradyan arttırma algoritması (GA), sınıflandırma ve regresyon problemlerinde kullanılan bir makine öğrenmesi tekniğidir. Çalışma prensibi: zayıf tahminlerle bir modeli güçlendirmek ve daha iyi bir hale getirmektir. Zayıf tahmin modellerini bir araya getirmektedir. Karar ağaçlarından oluşan bir modelin oluşturulup bu model üzerinden tahmin alınmaktadır [139].

3.8.8. Ridge Regresyon Algoritması

Ridge regresyon algoritması (RR), çok değişkenli regresyon verileri üzerinde analiz yapmaktadır. Algoritmanın amacı: kayıp fonksiyonların kareleri toplamını minimum yapan katsayıları bulmaktır. Tüm değişkenleri kullanarak modeli oluşturmaktadır. Model üzerinde ilgisiz alanları çıkartmak yerine bunların katsayı değerlerini sıfır olarak değiştirmektedir [145].

3.8.9. Makine Öğrenmesi Algoritmaları ile Bizim Çalışmalarımızın Kıyaslanması

Tez kapsamında hazırladığımız *VS_Pant*, *VS_Baglanti*, *VS_Nit1* ve *VS_Nit2* veri setleri, literatürde en sık kullanılan makine öğrenmesi tabanlı sınıflandırma yaklaşımlarında denenmiştir.

Makine öğrenmesi algoritmalarının uygulanması ile elde edilen sonuçlar Tablo 3.31' de detaylı şekilde ifade edilmiştir.

Tablo 3.31. Literatürdeki sınıflandırma metotları ile önerdiğimiz metotların performans karşılaştırılması

Algoritma	VS_Pant	VS_Baglanti	VS_Nit1	VS_Nit2
DVM	0.800	0.833	0.384	0.517
EA	0.514	0.666	0.153	0.258
GA	0.771	0.479	0.384	0.396
KA	0.800	0.520	0.307	0.534
K-NN	0.400	0.479	0.100	0.206
NB	0.714	0.395	0.461	0.327
RO	0.771	0.770	0.461	0.637
RR	0.771	0.729	0.461	0.500
Bizim çalışmamız	Bölüm 3.3' te 0.940	Bölüm 3.3' te 0.995 Bölüm 3.5' te 0.830 Bölüm 3.6' da 0.961 Bölüm 3.7' de 0.990	Bölüm 3.4' te 0.994	Bölüm 3.6' da 0.923

Tablo 3.31 incelendiğinde, tez kapsamında gerçekleştirdiğimiz KSA tabanlı yaklaşımlar, literatürde sıklıkla kullanılan yöntemlerden daha üstün sonuçlar elde etmiştir. Literatürde sıklıkla sınıflandırma alanında kullanılan makine öğrenmesi tabanlı yaklaşımlardan destek vektör makinesi, ekstra ağaçlar, gradyan azaltma, karar ağaçları, k-en yakın komşu algoritması, naive bayes, rastgele orman algoritması ve ridge regresyon algoritmaları kullanılmıştır.

VS_Pant veri seti yukarıda bahsedilen makine öğrenmesi algoritmaları ile sınıflandırma işlemine tabii tutulduğunda en iyi sonuç derin öğrenme yaklaşımdan elde edilmiştir. Derin öğrenme yaklaşımının dışında en iyi sonucu veren ilk üç algoritma ise destek vektör makinesi (0.80), karar ağaçları (0.80), ridge regresyon(0.77), rastgele orman(0.77) ve naive bayes(0.71) teknikleridir.

VS_Baglanti veri seti yukarıda bahsedilen makine öğrenmesi algoritmaları ile sınıflandırma işlemine tabii tutulduğunda en iyi sonuç derin öğrenme yaklaşımdan elde edilmiştir. Derin öğrenme yaklaşımının dışında en iyi sonucu veren ilk üç algoritma ise destek vektör makinesi (0.83), rastgele orman(0.77) ve ridge regresyon(0.72) teknikleridir.

VS_Nit1 veri seti yukarıda bahsedilen makine öğrenmesi algoritmaları ile sınıflandırma işlemine tabii tutulduğunda en iyi sonuç derin öğrenme yaklaşımdan elde edilmiştir. Derin öğrenme yaklaşımının dışında en iyi sonucu veren ilk üç algoritma ise naive bayes (0.46), rastgele orman(0.46), ridge regresyon(0.46), destek vektör makinesi(0.38), gradyan arttırma(0.38) ve karar ağaçları(0.30) teknikleridir.

VS_Nit2 veri seti yukarıda bahsedilen makine öğrenmesi algoritmaları ile sınıflandırma işlemine tabii tutulduğunda en iyi sonuç derin öğrenme yaklaşımdan elde edilmiştir. Derin öğrenme yaklaşımının

dışında en iyi sonucu veren ilk üç algoritma ise rastgele orman(0.63), karar ağaçları(0.53) ve destek vektör makinesi(0.51) teknikleridir.

4. SONUÇLAR

Makine öğrenmesi tabanlı nesne sınıflandırma çalışmaları, son zamanlarda artış göstermiştir. Literatürde konu ile ilgili birçok yöntem bulunmasına karşın nesne sınıflandırma alanında en çok kullanılarak popüler olan yaklaşım derin öğrenmedir. Bu tez kapsamında, derin öğrenme modelleri oluşturularak farklı türde birçok nesne grubunun sınıflandırılması amaçlanmıştır. Bu doğrultuda birbirine oldukça benzeyen nesne gruplarını sınıflandırmak tercih edilmiştir. Sınıflandırma kullanılan veri setleri, tez çalışmasında oluşturulmuştur. Bu veri setleri kullanılarak gerçekleştirilen çalışmalar, endüstriyel süreçlerde kullanılarak kullanıcının problemlerine çözüm olmuştur. Tezde gerçekleştirilen nesne sınıflandırma yöntemleri incelendiğinde bu yöntemlerin; yeni, etkin ve yüksek performanslı modeller olduğu fark edilmiştir. Makine öğrenmesi tabanlı nesne sınıflandırma amacıyla literatürde en sık kullanılan sekiz tane metot ile bizim geliştirdiğimiz yedi farklı yöntem ile kıyaslanmıştır. Elde edilen bulgular incelendiğinde, tezde gerçekleştirdiğimiz yöntemlerin literatürdekilere kıyasla oldukça başarılı olduğu gözlemlenmiştir.

Tez kapsamında, yüksek benzerlikli nesnelerin sınıflandırılması amacıyla derin metrik öğrenme yöntemlerine de başvurulmuştur. Bu yöntemler meta-öğrenme algoritmaları olarak anılmaktadır. Meta-öğrenme algoritmaları ile tezde oluşturduğumuz dört farklı veri seti sınıflandırılmıştır. Ayrıca meta öğrenme algoritması tabanlı olan ilişki ağı üzerinde çalışılarak yeni bir ilişki ağı tasarımı gerçekleştirilmiştir. Tez kapsamında tasarlanan ilişki ağı ile literatürdeki ilişki ağı kıyaslandığında tasarladığımız modelin yüksek performans sağladığı gözlemlenmiştir.

Tez çalışmamız süresince; ağ model mimarisinin, ağ parametrelerinin, eğitim ve test aşamasında kullanılan veri setlerinin ve sonuç sorgu verisinin gizliliğinin sağlanması için derin öğrenme yöntemleri birlikte kullanılarak diferansiyel gizlilik tekniği uygulanmıştır. Tez kapsamında hazırladığımız veri setlerinden yedek parça görüntülerinin gizliliğinin önem arz etmesi ve üçüncü şahıslara karşı korunmasının önemli olmasından dolayı diferansiyel gizlilik tekniği ile bu problem çözülmüştür.

Tezde uygulanılan tekniklerden biri ise aktarım öğrenme yöntemidir. Bu yöntemin çalışmamızda kullanım amacı; veri setimizde yeterli miktarda görüntü bulunmayışı ve hesaplama maliyetimizin düşürülmesinin amaçlanmasıdır. Geliştirilen uygulama, birçok farklı türdeki veri setleri için de hazır bir uygulama niteliğindedir. Kullanım kolaylığı ve yüksek performans sağlaması adına birçok sektörde tercih edilmesi beklenmektedir. Özellik çıkarım ve ince-ayar tabanlı hibrit yaklaşımı sayesinde nesne sınıflandırma alanında yüksek performans sağlamaktadır.

Tez süresince geliştirilen uygulamalardan Siyam ağları, Prototipik ağları, Karşılaştırma ağları ve İlişki ağları benzerlik metriği tabanlı olarak çalışmaktadır. Sınıflandırılması planlanan nesne grupları arasındaki benzerliğin esas alınmaktadır. Benzerlik ölçümlerinin yapıldığı ve bu ölçüme dayalı olarak nesne grubunun etiket değerinin belirlenmesi süreci ile diğer derin öğrenme modellerinden farklılaşmaktadır.

Tezde gerçekleştirilen tüm yöntemlerin detaylı anlatımı yapılmış olup her bir yöntemden elde edilen deneysel bulgular değerlendirilmiştir. Tez kapsamında, 3 adet SCI-E indeksli dergi yayın, 4 adet uluslararası konferanslarda sunulmuş bildiri ve 1 adet TR dizin indeksli dergi makalesi yapılarak literatüre bu alanda katkı sağlanmıştır.

KAYNAKLAR

[1] Şeker, A., Diri, B., & Balık, H. H. (2017). Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme, *Gazi Mühendislik Bilimleri Dergisi*, 3(3), ss. 47-64

[2] Yılmaz, Ö. Ü. A, & Yayın, K. (2021). *Derin Öğrenme*, Kodlab Yayın Dağıtım Yazılım Ltd. Şti.

[3] Togootogtokh, E., & Amartuvshin, A. (2018). Deep Learning Approach for Very Similar Objects Recognition Application on Chihuahua and Muffin Problem, *arXiv preprint* arXiv:1801.09573.

[4] Zhou, H., Xie, F., Jiang, Z., Liu, J., Wang, S., Zhu, S. (2017). Multi-classification of skin diseases for dermoscopy images using deep learning, *2017 IEEE International Conference on Imaging Systems and Techniques (IST)*.

[5] Korolev, S., Safiullin, A., Belyaev, M., Dodonova, Y. (2017). Residual and plain convolutional neural networks for 3D brain MRI classification, *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*.

[6] Farooq, A., Anwar, S., Awais, M., & Rehman, S. (2017). A deep CNN based multi-class classification of Alzheimer's disease using MRI, In *2017 IEEE International Conference on Imaging systems and techniques (IST)*, ss. 1-6.

[7] Algorry, A. M., García, A. G., & Wofmann, A. G. (2017). Real-Time Object Detection and Classification of Small and Similar Figures in Image Processing, In *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, ss. 516-519.

[8] Xie, J., & Zhu, M. (2019). Handcrafted features and late fusion with deep learning for bird sound classification, *Ecological Informatics*, cilt 52, ss. 74-81.

[9] Lundervold, A. S., & Lundervold, A. (2019). An overview of deep learning in medical imaging focusing on MRI, *Zeitschrift für Medizinische Physik*, cilt 29(2), ss. 102-127.

[10] Khan, S., Islam, N., Jan, Z., Din, I. U., & Rodrigues, J. J. C. (2019). A Novel Deep Learning based Framework for the Detection and Classification of Breast Cancer Using Transfer Learning, *Pattern Recognition Letters*.

[11] Zhang, J., Xie, Y., Wu, Q., & Xia, Y. (2019). Medical image classification using synergic deep learning, Medical image analysis, 54, 10-19.

[12] Han, J. H., Choi, D.J., Hong, S.K., and Kim, H.S. (2019). Motor Fault Diagnosis Using CNN Based Deep Learning Algorithm Considering Motor Rotating Speed, In *2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA)*, ss. 440-445.

[13] Bhadra, R., Dutta, S., Kedia, A., Gupta, S., Panigrahy, P.S., and Chattopadhyay, P. (2018). Applied Machine Learning for Bearing Fault Prognostics, In *2018 IEEE Applied Signal Processing Conference (ASPCON)*, ss. 158-162.

[14] Li, X., Liu, Z., Qu, Y., and He, D. (2018). Unsupervised Gear Fault Diagnosis Using Raw Vibration Signal Based on Deep Learning, In *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*, ss. 1025-1030.

[15] Yousif, H., Yuan, J., Kays, R. & He, Z., (2017). Fast human-animal detection from highly cluttered camera-trap images using joint background modeling and deep learning classification, In *IEEE International Symposium Circuits and Systems (ISCAS)*, Baltimore, MD, USA, ss. 1-4.

[16] Dolph, C. V., Alam, M., Shboul, Z., Samad, M. D. & Iftekharuddin K. M. (2017). Deep learning of texture and structural features for multiclass Alzheimer's disease classification, In *International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, ss. 2259-2266.

[17] Anavi, Y., Kogan, I., Gelbart, E., Geva, O. & Greenspan, H., A. (2015). Comparative study for chest radiograph image retrieval using binary texture and deep learning classification, In *Engineering in Medicine and Biology Society (EMBC)*, Milan, Italy.

[18] Sevakula, R. K., Singh, V., Verma, N. K. & Kumar C., Cui Y. (2019). Transfer Learning for Molecular Cancer classification using Deep Neural Networks, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, cilt 16(6), ss. 2089-2100.

[19] Seth, S. & Biswas, S. (2017). Multimodal Spam Classification Using Deep Learning Techniques, In *13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, Jaipur, India, ss. 346-349.

[20] Karahan, S. & Akgül, Y. S. (2016). Eye Detection by Using Deep Learning, In *2016 24th Signal Processing and Communication Application Conference (SIU)*, Zonguldak, Turkey, ss. 2145-2148.

[21] Karabulut, E. M. (2016). *Investigation of Deep Learning Approaches for Biomedical Data Classification*, Doktora Tezi, Çukurova Üniversitesi, Elektrik Mühendisliği Bölümü.

[22] Hatipoglu, P. U. (2016). *Time Series Classification Using Deep Learning*, Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi, Endüstri Mühendisliği Bölümü.

[23] Lehr, J., Schlüter, M., and Krüger, J. (2019). Classification of similar objects of different sizes using a reference object by means of convolutional neural networks, In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, ss. 1519-1522.

[24] Taheritanjani, S., Haladjian, J., and Bruegge, B. (2019). Fine-grained visual categorization of fasteners in overhaul processes, In *2019 5th International Conference on Control, Automation and Robotics (ICCAR)*, ss. 241-248.

[25] Zhao, X., Zhang, Y., and Wang, N. (2019). Bolt loosening angle detection technology using deep learning, *Structural Control and Health Monitoring*, cilt 26(1).

[26] Ruiz, L., Torres, M., Gómez, A., Díaz, S., González, J.M., and Cavas, F. (2020). Detection and Classification of Aircraft Fixation Elements during Manufacturing Processes Using a Convolutional Neural Network, *Applied Sciences*, cilt 10(19).

[27] Kamal, N. D. M. (2016). Fastener recognition using combination of colour features with shape-based features and Zernike moments, In *2016 IEEE Student Conference on Research and Development (SCOReD)*, ss. 1-5.

[28] Huang, Q. (2019). High-precision quality inspection for screws using artificial intelligence technology, *CIB W78-International Council for Research and Innovaiton in Building and Construction*.

[29] Song, L., Li, X., Yang, Y., Zhu, X., Guo, Q., and Yang, H. (2018). Detection of micro-defects on metal screw surfaces based on deep convolutional neural networks, *Sensors*, cilt 18(11).

[30] Wei, X., Jiang, S., Li, Y., Li, C., Jia, L., Li, Y. (2019). Defect Detection of Pantograph Slide Based on Deep Learning and Image Processing Technology, *IEEE Transactions on Intelligent Transportation Systems*, cilt. 21, sayı. 3, ss. 947-958.

[31] Pathak, Y., Shukla, P. K., Tiwari, A., Stalin, S., and Singh, S. (2020). Deep transfer learning based classification model for COVID-19 disease, *Pattern Recognit. Lett.*, cilt. 152, ss. 122-128.

[32] Akgun, D., Kabakuş, A. T., Senturk, Z. K. , Senturk, A., and Kucukkulahli, E. (2021). A transfer learning-based deep learning approach for automated COVID-19 diagnosis with audio data, *Turkish Journal of Electrical Engineering & Computer Sciences*, cilt. 29(SI-1), ss. 2807-2823, 2021, DOI: 10.3906/elk-2105-64.

[33] Sravan, V., Swaraj, K., Meenakshi, K., and Kora, P. (2020). A deep learning based crop disease classification using transfer learning, *Mater. Today: Proc.*, DOI: 10.1016/j.matpr.2020.10.846.

[34] Kudva, V., Prasad, K., and Guruvare, S. (2020). Hybrid transfer learning for classification of uterine cervix images for cervical cancer screening, *J. Digit. Imaging*, cilt. 33(3), ss. 619-631, 2020, DOI: 10.1007/s10278-019-00269-1.

[35] Thenmozhi, K., and Reddy, U. S. (2019). Crop pest classification based on deep convolutional neural network and transfer learning, *Comput. Electron. Agric.*, cilt. 164, DOI: 10.1016/j.compag.2019.104906

[36] Talo, M., Baloglu, U. B., Yıldırım, O., and Acharya, U. R. (2018). Application of deep transfer learning for automated brain abnormality classification using MR images, *Cogn. Syst. Res.*, cilt. 54, ss. 176-188, DOI: 10.1016/j.cogsys.2018.12.007.

[37] Mehrotra, R., Ansari, M. A., Agrawal, R., and Anand, R. S. (2020). A transfer learning approach for AI-based classification of brain tumors, *Machine Learning with Applications*, cilt. 2, DOI: 10.1016/j.mlwa.2020.100003.

[38] Yang, K., Yang, T., Yao, Y., and Fan, S. D., (2021). A transfer learning-based convolutional neural network and its novel application in ship spare-parts classification, *Ocean Coast Manag.*, cilt. 215, DOI: 10.1016/j.ocecoaman.2021.105971

[39] Ali, M. S., Miah, M. S., Haque, J., Rahman, M. M., and Islam, M. K. (2021). An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models, *Machine Learning with Applications*, cilt. 5, DOI: 10.1016/j.mlwa.2021.100036.

[40] Rahman, Z., and Ami, A. M. (2020). A transfer learning based approach for skin lesion classification from imbalanced data, In *2020 11th International Conference on Electrical and Computer Engineering (ICECE)*, ss. 65-68.

[41] Kumar, S., and Janet, B. (2021). DTMIC: Deep transfer learning for malware image classification, *Journal of Information Security and Applications*, cilt. 64, DOI: 10.1016/j.jisa.2021.103063.

[42] Giraddi, S., Seeri, S., Hiremath, P. S., and Jayalaxmi, G. N. (2020). Flower Classification using Deep Learning models, In *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, ss. 130-133.

[43] Wang, I. H., Lee, K. C., and Chang, S. L. (2020). Images Classification of Dogs and Cats using Fine-Tuned VGG Models, In *2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, ss. 230-233.

[44] Lee, S. W. (2021). Novel classification method of plastic wastes with optimal hyper-parameter tuning of Inception_ResnetV2, In *2021 4th International Conference on Information and Communications Technology (ICOIACT)*, ss. 274-279.

[45] Qian, Y., Li, G., Lin, X., Zhang, J., Yan, J., Xie, B., and Qin, J. (2019). Fresh tea leaves classification using inception-V3, In *2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP)*, ss. 415-419.

[46] Junaidi, A., Lasama, J., Adhinata, F. D., and Iskandar, A. R. (2021). Image Classification for Egg Incubator using Transfer Learning of VGG16 and VGG19, In *2021 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, ss. 324-328.

[47] Rajayogi, J. R., Manjunath, G., and Shobha, G. (2019). Indian food image classification with transfer learning, In *2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, cilt. 4, ss. 1-4.

[48] Espejo-Garcia, B., Malounas, I., Mylonas, N, Kasimati, A., and Fountas, S. (2022). Using EfficientNet and transfer learning for image-based diagnosis of nutrient deficiencies, *Computers and Electronics in Agriculture*, cilt. 196.

[49] Hossain, M. E., Islam, A., and Islam, M. S. (2020). A Proficient Model to Classify Bangladeshi Bank Notes for Automatic Vending Machine Using a Tıny Dataset with One-Shot Learning & Siamese Networks, In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, ss. 1-4.

[50] Ullah, A., Muhammad, K., Haydarov, K., Haq, I. U., Lee, M., and Baik, S. W. (2020). One-Shot Learning for Surveillance Anomaly Recognition using Siamese 3D CNN, In *2020 International Joint Conference on Neural Networks (IJCNN)*, ss. 1-8.

[51] Lungu, I. A., Hu, Y. and Liu, S. C. (2020). Multi-resolution siamese networks for one-shot learning, In *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, ss. 183-187.

[52] Yuan, J., Guo, H., Jin, Z., Jin, H., Zhang, X., and Luo, J. (2017). One-shot learning for fine-grained relation extraction via convolutional siamese neural network, In *2017 IEEE International Conference on Big Data (Big Data)*, ss. 2194-2199.

[53] Agarwal, S., Gudi, R., and Saxena, P. (2020). One-Shot learning based classification for segregation of plastic waste, *arXiv preprint*, 2020.

[54] Pho, H. A., Lee, S., Tuyet-Doan, V. N., and Kim, Y. H. (2020). Radar-Based Face Recognition: One-Shot Learning Approach, *IEEE Sensors Journal*, cilt. 21, sayı. 5, ss. 6335-6341.

[55] Sabri, N. I. A., and Setumin, S. (2021). One-Shot Learning for Facial Sketch Recognition using the Siamese Convolutional Neural Network, In *2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, ss. 307-312.

[56] Eloff, R., Engelbrecht, H. A., and Kamper, H. (2019). Multimodal one-shot learning of speech and images, In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, ss. 8623-8627.

[57] Van Zyl, T. L., Woolway, M., and Engelbrecht, B. (2020). Unique Animal Identification using Deep Transfer Learning For Data Fusion in Siamese Networks, In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, ss. 1-6.

[58] Vorugunti, C. S., Mukherjee, P., and Pulabaigari, V. (2019). OSVNet: convolutional siamese network for writer independent online signature verification, In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, ss. 1470-1475.

[59] Putra, A. A. R., and Setumin, S. (2021). The Performance of Siamese Neural Network for Face Recognition using Different Activation Functions, In *2021 International Conference of Technology, Science and Administration (ICTSA)*, ss. 1-5.

[60] Liu, Z., McClung, A., Yeung, H. W., Chung, Y. Y., and Zandavi, S. M. (2018). Top-down person re-identification with Siamese convolutional neural networks, In *2018 International Joint Conference on Neural Networks (IJCNN)*, ss. 1-8.

[61] Sokar, G., Hemayed, E. E., and Rehan, M. (2018). A generic OCR using deep siamese convolution neural networks, In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, ss. 1238-1244.

[62] Li, Y. (2021). Supervised Few-Shot Image Segmentation with Deep Metric Learning, In *2021 International Conference on Electronic Information Technology and Smart Agriculture (ICEITSA)*.

[63] Yang, X., Zhou, P., and Wang, M. (2020). Person reidentification via structural deep metric learning, *IEEE transactions on neural networks and learning systems*, cilt. 30(10), ss. 2987-2998.

[64] Duan, Y., Lu, J., Zheng, W., and Zhou, J. (2020). Deep Adversarial Metric Learning, *IEEE Transactions on Image Processing*, cilt. 29.

[65] Lu, J., Hu, J., and Tan, Y. P. (2017). Discriminative deep metric learning for face and kinship verification, *IEEE Transactions on Image Processing*, cilt. 26(9), ss. 4269-4282.

[66] Gundogdu, E., Solmaz, B., Koç, A., Yücesoy, V., and Alatan, A. A. (2017). Deep distance metric learning for maritime vessel identification, In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, ss. 1-4.

[67] Zhao, X., Qi, H., Luo, R., and Davis, L. (2019). A weakly supervised adaptive triplet loss for deep metric learning, In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*.

[68] Li, C., Liu, C., Duan, L., Gao, P., and Zheng, K. (2019). Reconstruction regularized deep metric learning for multi-label image classification, *IEEE transactions on neural networks and learning systems*, cilt. 31(7), ss. 2294-2303.

[69] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. (2018). Learning to compare: Relation network for few-shot learning, In *Proceedings of the IEEE conference on computer vision and pattern recognition*, ss. 1199-1208.

[70] Hui, B., Zhu, P., Hu, Q., and Wang, Q. (2019). Self-attention relation network for few-shot learning, In *2019 IEEE international conference on Multimedia & Expo Workshops (ICMEW)*, ss. 198-203.

[71] Rhee, S., Seo, S., and Kim, S. (2017). Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification, *arXiv preprint arXiv:1711.05859*.

[72] Park, H., and Ham, B. (2020). Relation network for person re-identification, In *Proceedings of the AAAI conference on artificial intelligence*, cilt. 34, sayı 07, ss. 11839-11847.

[73] Gao, K., Liu, B., Yu, X., Qin, J., Zhang, P., and Tan, X. (2020). Deep relation network for hyperspectral image few-shot classification, *Remote Sensing*, cilt. 12(6).

[74] Yadav, G., Maheshwari, S., and Agarwal, A. (2014). Contrast limited adaptive histogram equalization based enhancement for real time video system" In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Delhi, India, ss. 2392-2397.

[75] Aydin, I, Gray Level Transformations and Spatial Filtering Lecture Notes, Firat University, [Online]. Available: http://web.firat.edu.tr/iaydin/bmu357/bmu_357_bolum2.pdf

[76] Kurt, B., and Nabiyev, V. (2010). Dijital Mamografi Görüntülerinin Kontrast Sınırlı Adaptif Histogram Eşitleme ile İyileştirilmesi, *Proceedings of the VII. Ulusal Tıp Bilişimi Kongresi*, Gazimağusa, KKTC, ss. 14-17.

[77] Yoon, H., Han, Y., and Hahn, H. (2009). Image Contrast Enhancement based Sub-histogram Equalization Technique without Over-equalization Noise, *International Journal of Computer Science and Engineering*, cilt. 3, sayı. 2.

[78] Çiçek, G., Özmen, A. and Akan, A. (2019). The Effect of Data Augmentation on ADHD Diagnostic Model using Deep Learning, In *2019 Medical Technologies Congress (TIPTEKNO)*, Izmir, Turkey, ss. 1-4.

[79] Karakose, E. (2014). *Modeling, simulation and development of fault diagnosis methods for pantograph-catenary system in railway systems*, Doktora Tezi, Fen Bilimleri Enstitüsü, Fırat Üniversitesi.

[80] LeCun, Y., Bengio Y., and Hinton, G. (2015). Deep learning, *Nature*, cilt 21, sayı 7553, ss. 436-444.

[81] Wei, X., Jiang, S., Li, Y., Li, C., Jia, L., Li, Y. (2019). Defect Detection of Pantograph Slide Based on Deep Learning and Image Processing Technology, *IEEE Transactions on Intelligent Transportation Systems*, cilt 21, sayı 3, ss. 947-958.

[82] Bilgiç, A. (2017). *Derin Öğrenmeye Dayali Yüz Ve Vücut Biyometrilerinin Tümleştirilmesi*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesitesi, Fen bilimleri Enstitüsü, Elektronik ve Haberleşme Mühendisliği Anabilim Dalı.

[83] Ribani, R., and Marengoni, M. (2019). A survey of transfer learning for convolutional neural networks, In *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, ss. 47-57.

[84] Simonyan K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition, *arXiv Prepr. arXiv1409.1556*.

[85] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision, In *Proceedings of the IEEE conference on computer vision and pattern recognition*, ss. 2818–2826.

[86] Ucar, M. (2021). Diagnosis of Glaucoma Disease using Convolutional Neural Network Architectures, *Dokuz Eylul University Faculty of Engineering Journal of Science and Engineering*, cilt. 23(68), ss. 521-529.

[87] Nguyen, L. D., Lin, D., Lin, Z., and Cao, J. (2018). Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation, In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, ss. 1-5.

[88] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. ve Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, *arXiv Prepr. arXiv /1704.04861*.

[89] Zeren, M. T. (2020). *Comparison of ssd and faster r-cnn algorithms to detect the airports with data set which obtained from unmanned aerial vehicles and satellite images*, Yüksek Lisans Tezi, Beykent Üniversitesi, Endüstri Mühendisliği Bölümü.

[90] Baydilli, Y. Y. (2021). Polen Taşıyan Bal Arılarının MobileNetV2 Mimarisi ile Sınıflandırılması, *European Journal of Science and Technology*, sayı.21, ss. 527-533..

[91] Huang, G., Liu, Z., Maaten, L. V. D., and Weinberger, K. O. (2017). Densely Connected Convolutional Networks, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[92] Aktas, A. (2020). *Image processing applications with deep learning methods*, Yüksek Lisans Tezi, Marmara Üniversitesi, Bilgisayar Mühendisliği.

[93] Tan M. and Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks, *36th Int. Conf. Mach. Learn. ICML 2019*, ss. 10691–10700.

[94] Bayram, B., Kılıc, B., Ozoglu, F., Erdem, F., Sivri, S., Delen, A., Bayrak, O. C. (2019). A study on object recognition with deep learning, In *10. Turkiye Ulusal Fotogrametri ve Uzaktan Algılama Birligi Teknik Sempozyumu (TUFUAB 2019)*.

[95] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition, In *Proceedings of the IEEE conference on computer vision and pattern recognition*, ss. 770-778.

[96] Dandil, E., and Serin, Z. (2020). Breast Cancer Detection on Histopathological Images Using Deep Neural Networks, *European Journal of Science and Technology*, Özel sayı, ss. 451-463.

[97] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions, In *IEEE conference on computer vision and pattern recognition*.

[98] Zaheer, R., and Shaziya, H. (2019). A study of the optimization algorithms in deep learning, In *2019 third international conference on inventive systems and control (ICISC)*, ss. 536-539.

[99] Soydaner, D. (2020). A comparison of optimization algorithms for deep learning, *International Journal of Pattern Recognition and Artificial Intelligence*, cilt 34(13).

[100] Kurt, F. (2018). *Analysis of the effects of hyper-parameters in convolutional neural networks*, Yüksek Lisans Tezi, Hacettepe Üniversitesi, Bilgisayar Mühendisliği Bölümü.

[101] Yazan, E. and Talu, M. F. (2017). Comparison of the stochastic gradient descent-based optimization techniques, In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, Malatya, Turkey, ss. 1-5.

[102] Alpaydın, E. (2018). *Yapay Öğrenme*, Boğaziçi Üniversitesi Yayınevi, İstanbul.

[103] Salouhou, A. (2019). *Deep learning approaches in handwritting character recognition and image classification*, Yüksek Lisans Tezi, Fatih Sultan Mehmet Vakıf Üniversitesi, Bilgisayar Mühendisliği Bölümü.

[104] Kingma D. P. and Ba, J. (2014). Adam: a method for stochastic optimization, *arXiv preprint arXiv:1412.6980*, ss. 1–15.

[105] Zhang, Z. (2018). Improved adam optimizer for deep neural networks, In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, Banff, Canada, ss. 1-2.

[106] Mehta, S., Paunwala, C. and Vaidya, B. (2019). CNN based Traffic Sign Classification using Adam Optimizer, In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India, ss. 1293-1298.

[107] Colab Research. (2019). Implement Differential Privacy with TensorFlow Privacy. [Online]. Available: https://colab.research.google.com/github/tensorflow/privacy/blob/master/tutorials/Classification_Privacy.ipynb?ref=hackernoon.com#scrollTo=KwDK47gfLsYf

[108] Papernot, N., McDaniel, P., Sinha, A., and Wellman, M. (2016). Towards the science of security and privacy in machine learning, *arXiv preprint arXiv:1611.03814*.

[109] Papernot, N., and McDaniel, P. (2016). The limitations of deep learning in adversarial settings, In *Proceedings of the 1st IEEE European Symposium on Security and Privacy*, Saarbrücken, Germany.

[110] Sutskever, I., Vinyals, O. and Le, Q. V. (2014). Sequence to sequence learning with neural networks, *Advances in Neural Information Processing Systems*, ss. 3104–3112.

[111] Friedman, A. and Schuster, A. (2010). Data mining with differential privacy, In *Proceedings of the 16th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, Washington, DC, USA, ss. 493-502.

[112] Hu, X., Yuan, M., Yao, J., Deng, Y., Chen, L., Yang, Q., and Zeng, J. (2015). Differential privacy in telco big data platform, *Proceedings of the VLDB Endowment*, cilt 8, sayı 12, ss. 1692-1703.

[113] Grigorescu, S. M. (2018). Generative One-Shot Learning (GOL): A semi-parametric approach to one-shot learning in autonomous vision. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, ss. 7127-7134.

[114] Koch, G., Zemel, R. & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*.

[115] Rao, D. J., Mittal, S. & Ritika, S. (2017). Siamese neural networks for one-shot detection of railway track switches, *arXiv preprint.*

[116] Torres, L., Ribeiro, B. & Arrais, J. (2020). Exploring a Siamese Neural Network Architecture for Drug Discovery, *Deep Drug Discovery and Deployment Project, University of Coimbra*, CISUC, DIE.

[117] Hsiao, S. C., Kao, D. Y., Liu, Z. Y. & Tso, R. (2019). Malware image classification using one-shot learning with siamese networks, *Procedia Computer Science*, cilt. 159, ss:1863-1871.

[118] Kang, D., Kwon, H., Min, J., & Cho, M. (2021). Relational embedding for few-shot classification, In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, ss. 8822-8833.

[119] Baik, S., Choi, J., Kim, H., Cho, D., Min, J., & Lee, K. M. (2021). Meta-Learning with Task-Adaptive Loss Function for Few-Shot Learning, *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, ss. 9465-9474.

[120] Sun, X., Wang, B., Wang, Z., Li, H., Li, H. & Fu, K. (2021). Research progress on few-shot learning for remote sensing image interpretation, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, cilt. 14, ss. 2387-2402.

[121] Wang, Y., Yao, Q., Kwok, J. T. and Ni, L. M. (2020). Generalizing from a Few Examples: A Survey on Few-Shot Learning, *ACM Comput. Surv.*, cilt. 1, sayı.1.

[122] Yue, Z., Zhang, H., Sun, Q., Sheng Hua, X. (2020). Interventional Few-Shot Learning, *arXiv:2009.13000*.

[123] Ravichandran, A., Bhotika, R. and Soatto, S., Few-Shot Learning with Embedded Class Models and Shot-Free Meta Training, In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, ss. 331-339.

[124] Cakir, F., He, K., Xia, X., Kulis, B. & Sclaroff, S. (2019). Deep metric learning to rank, In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, ss. 1861-1870.

[125] Huai, X., Xue, H., Miao, C., Yao, L., Su, L., Chen, C., & Zhang, A. (2019). Deep Metric Learning: The Generalization Analysis and an Adaptive Algorithm, In *IJCAI*, ss. 2535-2541.

[126] Zheng, W., Zhang, B., Lu, J., & Zhou, J. (2021). Deep relational metric learning, In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, ss. 12065-12074.

[127] Wang, J., Zhou, F., Wen, S., Liu, X., & Lin, Y. (2017). Deep metric learning with angular loss, In *Proceedings of the IEEE international conference on computer vision*, ss. 2593-2601.

[128] Kaya, M. & Bilge, H. S. (2019). Deep metric learning: A survey, *Symmetry*, cilt 11(9).

[129] Ge, W. (2018). Deep metric learning with hierarchical triplet loss, In *Proceedings of the European Conference on Computer Vision (ECCV)*, ss. 269-285.

[130] Khan, A., Fleming, E., Schofield, N., Bishop, M. & Andrews, N. (2021). A deep metric learning approach to account linking, *arXiv preprint arXiv:2105.07263*.

[131] He, J., Hong, R., Liu, X., Xu, M., Zha, Z., Wang, M. (2020). Memory-Augmented Relation Network for Few-Shot Learning, *ACM Multimedia Conference*.

[132] Zhou, X., Li, J., Wang, Z., He, R. & Tan, T. (2021). Image inpainting with contrastive relation network, In *2020 25th International Conference on Pattern Recognition (ICPR)*, ss. 4420-4427.

[133] Liu, J., Shi, C., Tu, D., Shi, Z. & Liu, Y. (2021). Zero-Shot Image Classification Based on a Learnable Deep Metric, *Sensors*, cilt 21(9).

[134] Karaduman, G. and Akin, E. (2020). A Deep Learning Based Method for Detecting of Wear on the Current Collector Strips' Surfaces of the Pantograph in Railways, *IEEE Access*, cilt. 8.

[135] Li, Y. & Wei, X. (2017). Pantograph slide plate abrasion detection based on deep learning network, *International Conference on Electrical and Information Technologies for Rail Transportation*, ss. 215-224.

[136] Adesuyi T. A. & Kim B. M. (2019). Preserving Privacy in Convolutional Neural Network: An∈-tuple Differential Privacy Approach, In *2019 IEEE 2nd Int Conf on Knowl Innov and Invent (ICKII)*, Seoul, South Korea, ss. 570-573.

[137] Arachchige P. C. M., Bertok P., Khalil I., Liu D., Camtepe S. & Atiquzzaman M. (2020). Local differential privacy for deep learning, *IEEE Internet Things J*, cilt 7(7), ss. 5827-5842.

[138] Şekerci, M., Kandemir, R., Sözlük kullanarak Türkçe el yazısı tanıma, Edirne, 2006.

[139] Türkmen, E. (2021). *Makine öğrenmesi yöntemleri ile banka pazarlama tahmini*, Yüksek Lisans Tezi, İstanbul Kültür Üniversitesi, Lisansüstü Eğitim Enstitüsü/Bilgisayar Mühendisliği Ana Bilim Dalı/Bilgisayar Mühendisliği Bilim Dalı.

[140] Mohan, L., Pant, J., Suyal P. and Kumar, A. (2020). Support Vector Machine Accuracy Improvement with Classification, In *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, Bhimtal, India.

[141] Karakaya, R. (2020). *Makine öğrenmesi yöntemleriyle el yazısı tanıma*, Yüksek Lisans Tezi, Sakarya Üniversitesi.

[142] Yılmaz, H. (2014). *Random forests yönteminde kayıp veri probleminin incelenmesi ve sağlık alanında bir uygulama*, Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi.

[143] Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees, *Machine learning*, cilt. 63(1), ss. 3-42.

[144] Emanet, S., Karatas Baydogmus, G., & Demir, O. (2021). Effects of Feature Selection Methods on Machine Learning Based Intrusion Detection System Performance, *Dicle University Journal of Engineering*, cilt. 12(5).

[145] He, J., Ding, L., Jiang, L., & Ma, L. (2014). Kernel ridge regression classification, In *2014 International Joint Conference on Neural Networks (IJCNN)*, ss. 2263-2267.