



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**TERMAL GÖRÜNTÜLERDE DERİN
ÖĞRENME YAKLAŞIMLARI İLE ELEKTRİK
ARIZALARININ SINIFLANDIRILMASI**

Gönül SAKALLI

YÜKSEK LİSANS TEZİ

Elektrik Elektronik Mühendisliği Anabilim Dalı

Mart-2023
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Gönül SAKALLI tarafından hazırlanan “Termal Görüntülerde Derin Öğrenme Yaklaşımları ile Elektrik Arızalarının Sınıflandırılması” adlı tez çalışması 16.03.2023 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof. Dr. Salih GÜNEŞ

.....

Danışman

Dr. Öğr. Üyesi Hasan KOYUNCU

.....

Üye

Dr. Öğr. Üyesi Mehmet DEMİRTAŞ

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Gönül SAKALLI

Tarih: 16.03.2023

ÖZET**YÜKSEK LİSANS TEZİ****TERMAL GÖRÜNTÜLERDE DERİN ÖĞRENME YAKLAŞIMLARI İLE
ELEKTRİK ARIZALARININ SINIFLANDIRILMASI****Gönül SAKALLI****Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı****Danışman: Dr. Öğr. Üyesi Hasan KOYUNCU****2023, 80 Sayfa****Jüri****Prof. Dr. Salih GÜNEŞ
Dr. Öğr. Üyesi Hasan KOYUNCU
Dr. Öğr. Üyesi Mehmet DEMİRTAŞ**

Asenkron motorlar, endüstriyel uygulamalarda diğer motor tiplerine olan avantajları sebebiyle sıklıkla tercih edilirler. Transformatörler ise elektrik sistemine beslenecek gerilimin ayarlanmasında kullanılan, vazgeçilemez bir diğer kategoriye teşkil ederler. Bu ekipmanlara dair arıza teşhisi ise geleneksel elektrik bazlı ölçümleri gerektiren analizlerle (stator akım sinyallerinin, manyetik akı dağılımlarının vb. derinlemesine incelenmesiyle) yerine getirilir. Termal görüntü analizleri; yapıya doğrudan müdahale gerektirmeyen, elektrikli ekipmanların durumlarını belirlemenin kolay bir yolu olarak karşımıza çıkmaktadır. Gerçeklenen tez çalışmasında, asenkron motorların ve transformatörlerin durumlarını ayırt etmek için, termal görüntü temelli analizler ele alınmaktadır. Bu amaçla her iki ekipman koşulları birleştirilerek, yirmi farklı durumun eldesi sağlanmıştır. Bu durumlar; soğutma fanı arızası, rotor arızası, motor içinde farklı fazlarda ve çeşitli oranlarda kısa devre arızaları, trafo içinde farklı oranlarda kısa devre arızaları, yüksüz motor ve yüksüz trafo olarak tanımlanmıştır. Sınıflama problemi bir ön işleme olmaksızın, etkin derin öğrenme mimarileri (DenseNet201, MobileNetV2, ResNet50, ShuffleNet, Xception) üzerinden gerçekleştirilmiştir. Deneylede mimarilere dair en yüksek performansların gözlenmesi için, sistem hiperparametreleri kapsamlı bir şekilde incelenmiştir. Burada transfer öğrenme görevini gerçekleştirmek, modellerin ana kısmını bozmamak ve termal görüntü sınıflaması için uygun modeli ortaya çıkarmak için dört olgu (mini parça boyutu, öğrenme oranı, LRDF değeri, optimize edici tipi) değerlendirilmiştir. Deneylede mimarileri veri artırma olmadan karşılaştırmak için, %80-%20 eğitim-test yöntemi üzerinden sonuçlar analiz edilmiştir. Sonuç olarak, üç fazlı motora ve bir fazlı trafoya ait termal görüntülerin sınıflandırılmasında %100 doğruluk elde edilerek, tüm derin öğrenme mimarilerinde en yüksek performanslar gözlemlenmiştir. Doğruluk tabanlı analizlere ek olarak, termal görüntü sınıflamasında en uygun mimariyi ortaya çıkarmak için derinlemesine bir analiz sunulmuştur. Çalışma neticesinde, derinlemesine analizlere göre ShuffleNet mimarisinin diğer yapılardan daha üstün olduğu tespit edilmiştir.

Anahtar Kelimeler: Asenkron motor, Derin öğrenme, Görüntü sınıflandırma, Kızılötesi, Termal, Transfer öğrenme, Transformatör

ABSTRACT**MS THESIS****CLASSIFICATION OF ELECTRICAL FAULTS IN THERMAL IMAGES
VIA DEEP LEARNING APPROACHES****Gönül SAKALLI****Konya Technical University
Institute of Graduate Studies
Department of Electrical and Electronics Engineering****Advisor: Asst. Prof. Dr. Hasan KOYUNCU****2023, 80 Pages****Jury
Prof. Dr. Salih GÜNEŞ
Asst. Prof. Dr. Hasan KOYUNCU
Asst. Prof. Dr. Mehmet DEMİRTAŞ**

Asynchronous motors are thoroughly preferred in industrial applications regarding their advantages in comparison with other motor types. Transformers constitute another oft-used and inevitable category for the adjustment of voltage to be fed to an electrical system. The fault diagnostics of these equipment are generally fulfilled by in-depth determinations of stator current signals, magnetic flux distributions, etc. which require comprehensive analyses of conventional electrical measurements. Herein, thermal image analyses arise as an easy way to identify the situations of electrical equipment in which there is no need for direct intervention to the structure. In thesis study, we handle the thermal image-based analyses to distinguish the situations of asynchronous motors and transformers. For this purpose, twenty different situations were obtained combining both asynchronous motor and transformer conditions. These conditions were defined as the cooling fan failure, rotor fault, short-circuit faults in different phases and in various rates inside the motor, short-circuit faults in different rates inside the transformer, no-load motor, and no-load transformer. Without a pre-processing step, classification process was performed utilizing efficient deep learning architectures (DenseNet201, MobileNetV2, ResNet50, ShuffleNet, Xception). In experiments, the hyperparameters of models are examined in a comprehensive manner to observe the highest scores of architectures that can be achieved. Herein, four phenomena (mini-batch size, learning rate, LRDF, optimizer type) are evaluated to perform the transfer learning task, not to spoil the main part of the models, and to reveal the appropriate model for thermal image classification. In trials, an 80%-20% training-test split is allowed to compare the models without data augmentation. As a result, the highest performance is observed with all deep learning architectures by attaining 100% accuracy for condition discrepancy of thermal images belonging to the three phase-motor and one phase-transformer. In addition to the accuracy-based analyses, an in-depth evaluation is presented to reveal the most appropriate architecture in thermal image classification. Regarding this, it's advised that ShuffleNet architecture seems as the most remarkable system among other transfer learning methods.

Keywords: Asynchronous Motor, Deep Learning, Image Classification, Infrared, Thermal, Transfer Learning, Transformer

ÖNSÖZ

Yüksek lisans dönemim boyunca kıymetli bilgi ve tecrübeleri ile bana yol gösterici ve destek olan, bu süreçte yardımlarını esirgemeyen danışmanım Konya Teknik Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği öğretim üyesi Dr. Öğr. Üyesi Hasan KOYUNCU hocama, çalışmalarım boyunca beni maddi ve manevi destekleyen aileme teşekkürü bir borç bilirim.

Gönül SAKALLI
KONYA-2023



İÇİNDEKİLER

ÖZET	i
ABSTRACT.....	ii
ÖNSÖZ	iii
İÇİNDEKİLER	iv
SİMGELER VE KISALTMALAR	vi
1.GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	2
2.1. Transformatör Durum Tespiti Üzerine Literatür Çalışmaları.....	2
2.2. Asenkron Motor Durum Tespiti Üzerine Literatür Çalışmaları	3
2.3. Kaynak Araştırması Özeti.....	6
3. MATERYAL VE YÖNTEM.....	8
3.1. Derin Öğrenme	8
3.2. Transfer Öğrenme	9
3.3. Evrimsel Sinir Ağları	10
3.3.1. Giriş katmanı.....	12
3.3.2. Evrişim katmanı	12
3.3.3. Doğrultulmuş doğrusal birim katmanı	13
3.3.4. Havuzlama katmanı	14
3.3.5. Tam bağlı katman	15
3.3.6. Seyreltme katmanı	15
3.3.7. Sınıflandırma katmanı.....	16
3.3.8. Optimizasyon teknikleri.....	17
3.4. Transfer Öğrenme Yöntemleri.....	18
3.4.1. DenseNet201 mimarisi	18
3.4.2. MobileNetV2 mimarisi	20
3.4.3. ResNet50 mimarisi	22
3.4.4. ShuffleNet mimarisi.....	22
3.4.5. Xception mimarisi.....	25
3.5. Kullanılan Veri Seti Özellikleri	27
3.6. Tez Ön Çalışmasında Kullanılan Teknikler ve Metrikler.....	28
3.6.1. Birincil derece istatistikler	28
3.6.2. Sınıflandırma algoritmaları ve karşılaştırma metrikleri.....	30
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	32
4.1. Tez Ön Çalışması.....	32
4.2. ResNet50 Mimarisi İle Asenkron Motor Durum Tespiti.....	35
4.3. Transfer Öğrenme Mimarileri İle Kapsamlı Durum Sınıflama	38
4.3.1. DenseNet201 temelli analizler.....	39

4.3.2. MobileNetV2 temelli analizler	40
4.3.3. ResNet50 temelli analizler	41
4.3.4. ShuffleNet temelli analizler	42
4.3.5. Xception temelli analizler	43
4.4. Detaylı Analiz ve Tartışma	44
5. SONUÇLAR VE ÖNERİLER	50
5.1 Sonuçlar	50
5.2 Öneriler	51
KAYNAKLAR	52
EKLER	59



SİMGELER VE KISALTMALAR

Simgeler

$f(x,y)$: Görüntüyü temsil eden fonksiyon
$\delta(j,i)$: Kronecker Delta fonksiyonu
$h(i)$: i .nci gri seviyeye dair tekrarlanma (histogram) değeri
$p(i)$: i .nci gri seviyeye dair OYF değeri
μ	: Ortalama değer
σ	: Standart sapma değeri
μ_3	: Eğrilik değeri
μ_4	: Basıklık değeri

Kısaltmalar

AmDD	: Ampirik Dalgacık Dönüşümü
ADD	: Ayrık Dalgacık Dönüşümü
AUC	: İşlem Karakteristik Eğrisi Altında Kalan Alan
BBA	: Bağımsız Bileşen Analizi
BDİ	: Birincil Derece İstatistikler
BM	: Bulanık Mantık
ÇÜA	: Çekişmeli Üretici Ağ
DVM	: Destek Vektör Makinesi
GA	: Genetik Algoritma
GDEM	: Gri Düzey Eş Oluşum Matrisi
GİB	: Grafik İşlem Birimi
HFÖÜ	: Harmonik Fazör Ölçüm Ünitesi
İEKBT	: İzole Edilmiş Kapılı – Bipolar Transistör
KA	: Karar Ağaçları
KAO	: Karınca Aslanı Optimizasyonu
KBA	: Komşu Bileşen Analizi
KeBA	: Kesirsel Boyut Algoritmaları
KDD	: Kısa Devre Dönüşleri
k-EYK	: k-En Yakın Komşuluk
KOA	: K-Ortalama Algoritması
LDA	: Lineer Diskriminant Analizi
MM	: Mahalanobis Mesafesi
OBO	: Ortalama Başarı Oranı
OYF	: Olasılık Yoğunluk Fonksiyonu
RO	: Rasgele Orman
SEM	: Sonlu Elemanlar Metodu
SR	: Stokastik Rezonans
TBA	: Temel Bileşen Analizi
TİB	: Toplam İstatistiksel Başarı
ÜÇDVVT	: Üç Çekirdek Destek Vektör Veri Tanımlaması
YİÖ	: Yerel İkili Örüntüler
YOA	: Yabani Ot Algoritması
YSA	: Yapay Sinir Ağları
2B	: İki Boyutlu
3B	: Üç Boyutlu

1.GİRİŞ

Elektrikli ekipmanların bakımı, endüstride güç iletiminin ve enerji akışının sürekliliği için önem arz etmektedir. Arıza tespiti, bakıma dayalı müdahalelerden önceki ilk adımı oluşturur. Sıklıkla kullanılan ve kalıcı olan malzemelerin kusurları, analiz edilmesi gereken önemli ve kapsamlı bir konu teşkil etmektedir. Bu noktada asenkron motorlar ve transformatörler, sırasıyla mekanik güce dayalı mühendislik uygulamaları ve elektriksel güç iletimi için kullanımı vazgeçilmez ekipmanlardır (Benbouzid, 1999; Muraleedharan ve ark., 2019).

Literatürde transformatörlerin arıza tespiti için farklı modeller kullanılmaktadır. Birçok çalışmada manyetik alan ölçümü (Muraleedharan ve ark., 2019; Li ve Zhou, 2020), manyetik akı kaçak dağılımları (Ouyang ve ark., 2020), titreşim sinyalleri (Huerta-Rosales ve ark., 2021), harmonik analizi (Granados-Lieberman ve ark., 2021), gerilim sinyali temelli analizler (Cao ve ark., 2021), kızılötesi görüntü analizi (Fang ve ark., 2021) vb. dikkate alınarak, transformatörlerin kısa devre ve aşırı ısınma durumları sınıflandırılmaktadır. Asenkron motorlara dair çeşitli durumlarının belirlenmesi ise manyetik akı dağılımları (Li ve ark., 2020), üç fazlı gerilim ve akım sinyalleri (Jorkesh ve Poshtan, 2021), ses sinyalleri (Yaman, 2021), stator akım ve titreşim sinyalleri (Saucedo-Dorantes ve ark., 2021), akım sinyalleri (Yi ve ark., 2022), başlangıç geçici akım sinyalleri (Martinez-Herrera ve ark., 2022), termografik görüntüler (Lozanov ve ark., 2020; Najafi ve ark., 2020; Choudhary ve ark., 2020; Khanjani ve Ezoji, 2021; Ibrahim ve ark., 2022(a); Ibrahim ve ark., 2022(b)) üzerinden gerçekleştirilmektedir. Burada termografik görüntüleme ile sağlanan ölçüm, yapıya doğrudan müdahale gerektirmeyen ve uzaktan gerçekleştirilen bir işlemdir. Bu nedenle termografik görüntüleme ile durum tespiti, gelişime açık ve ergonomik bir konu olarak karşımıza çıkmaktadır.

Literatür çalışmalarında genel olarak, tekil bir elektriksel ekipman ve ekipmana dair sınıflar ele alınmaktadır. Ancak birden fazla ekipman için çoklu sınıf durumlarını, termal görüntüleme üzerinden değerlendiren kapsamlı bir çalışma mevcut değildir. Tez çalışması kapsamında sık kullanımları nedeniyle üç fazlı bir asenkron motor ve tek fazlı bir trafoya ait çoklu durum tespiti işlenmektedir. Bu amaçla, literatürde etkinliğini ispat eden derin öğrenme yaklaşımları kullanılarak, kapsamlı bir analiz sunulmaktadır.

2. KAYNAK ARAŞTIRMASI

Bu bölüm kapsamında literatür çalışmaları incelenmekte, sırasıyla transformatör ve asenkron motor durumlarını ele alan belirli çalışmalar değerlendirilmektedir. Gerçeklenen tez çalışmasına dair motivasyon ise elde edilen bulgular analiz edilerek bölüm sonu sunulmaktadır.

2.1. Transformatör Durum Tespiti Üzerine Literatür Çalışmaları

Muraleedharan ve ark. (2019) trafo sargılarındaki kısa devre arızalarını belirlemek için, Sonlu Elemanlar Metodu (SEM) temelli bir yaklaşım önermişlerdir. ANSYS yazılımında 11kV/440V dönüşümlü bir trafo modellenerek, simülasyon analizleri gerçekleştirilmiştir. Manyetik alan şiddeti ve manyetik akı dağılımına göre normal ve üç farklı arıza (diskler arası %5, %10 ve %15'lik arıza) ayırt edilmiştir. Arızaları analiz etmede sargı merkezinin yanı sıra her iki simetrik disk üzerindeki manyetik özellikler incelenmiştir. Çalışma kapsamında (trafo arıza tanıma ile ilgili birçok literatür çalışmasında olduğu gibi) doğruluk veya herhangi bir başarı indeksi üzerinden değerlendirme sunulmamıştır. Burada bir arıza parametresi önerilerek, parametre üzerinden arızaların yüzdelik oranları tanımlanmıştır.

Li ve Zhou (2020), *Maxwell* yazılımını kullanarak simülasyon tabanlı analizler gerçekleştirmişlerdir. Burada 110kV'luk bir trafo modeli kurularak, sargılar arası kısa devre durumunda manyetik alan dağılımları incelenmiştir. Çalışma neticesinde manyetik alan özelliklerinin, normal ve sargılar arası kısa devre etiketlerini tanımlamaya yardımcı olabileceği önerilmiştir.

Jiang ve ark. (2020) 35kV/10.5kV dönüşüme sahip olan bir trafoyu modelleyerek, birkaç farklı durumun (normal, radyal deformasyon, eksenel deformasyon, dönüşler arası kısa devre, harici kısa devre) kategorizasyonunu ele almışlardır. Koşulların ayırt edilmesinde ise manyetik akı kaçak dağılımı dikkate alınmıştır. Sonuç olarak; durumları ayırt etmek için, manyetik farklardan yararlanılabileceği ve manyetik akının bir durumdan diğerine değişkenlik gösterdiği görsel olarak ispat edilmiştir.

Huerta-Rosales ve ark. (2021) arızasız durum ile 5, 10 ve 15 seviyelerinde üç farklı kısa devre durumunu ayırt etmek için, transformatörün titreşim sinyallerini analiz etmişlerdir. Ampirik Dalgacık Dönüşümü (AmDD) ve MUSIC algoritmasının

hibritleştirilmesiyle elde edilen MUSIC-AmDD tekniği, sinyallerdeki gürültüyü ve alakasız bölümleri ortadan kaldırmak için kullanılmıştır. Karar işleminin sağlanması, Kesirsel Boyut Algoritmaları (KeBA) üzerinden gerçekleştirilmiştir. Çalışma neticesinde önerilen modelin, üç farklı kısa devre durumunun ayırımında kullanılabileceği önerilmiştir.

Granados-Lieberman ve ark. (2021) gerilim ve akım sinyallerinden harmonikleri elde etmek için, 120/240V'luk bir transformatör üzerinden ölçüm almışlardır. Harmonik Fazör Ölçüm Ünitesi (HFÖÜ) ile elde edilen harmonikler, Bulanık Mantık (BM) yöntemi ile sınıflandırılmıştır. Bu amaçla normal ve beş farklı yüzdelikte (%5, %10, %15, %20, %25) Kısa Devre Dönüşleri (KDD) ele alınmıştır. Çalışma neticesinde önerilen modelin, altı farklı durum ayırımında %100 doğruluk sağladığı tespit edilmiştir.

Cao ve ark. (2021), bir katı hal trafosunu anahtarlamak için İzole Edilmiş Kapılı – Bipolar Transistör (İEKBT) kullanmışlardır. Çalışma neticesinde açık devre ve kısa devre arızalarının tespitinde, kollektör–emitör gerilimi ve trafo birincil sargı geriliminin ayırt edici nitelikte olduğu önerilmiştir.

Fang ve ark. (2021) termal görüntüdeki bölgesel sıcaklık özelliklerini işleyerek sıcaklık bilgisini, Gri Düzey Eş Oluşum Matrisi (GDEM) tekniğini kullanarak dokusal bilgiyi ve Hu moment metodunu ile şekilsel bilgiyi üretmişlerdir. Elde edilen bu bilgiler birleştirilerek, Çekişmeli Üretici Ağ (ÇÜA) mimarisine sunulmuştur. Önerilen model ile ekipman hatası ve aşırı ısınma kategorileri sırasıyla %82,2 ve %86,2 doğruluk oranları ile sınıflandırılmıştır. Burada modele dair ortalama başarının %84,2 olduğu tespit edilmiştir.

2.2. Asenkron Motor Durum Tespiti Üzerine Literatür Çalışmaları

Li ve ark. (2020) bir asenkron motora dair üç kategoriye (rotor eksenel kaçıklığı, kırık rotor çubuğu, shaft çatlağı) ayırt etmek için, SEM temelli değerlendirmeyi ve manyetik akı yoğunluk analizlerini ele almışlardır. Manyetik akı yoğunluğunun incelenmesi neticesinde şekilsel farklılıkların ve akı yoğunluğu değişimine dair lokasyon bilgisinin, sınıflama görevinde ayırt edici özellikler olduğu belirlenmiştir.

Lozanov ve ark. (2020), indüksiyon motoruna dair üç koşulu (soğutma fanı arızası, motor yatağı arızası ve arızasız) sınıflamak için termal görüntü temelli bir model önermişlerdir. Model kapsamında Otsu eşikleme kullanılarak motor yapısı segmente edilmiştir. Gerekli bilgi eldesi amacıyla histogram, GDEM ve bileşen temelli özellikler

incelenmiştir. Destek Vektör Makinesi (DVM) yöntemi ile %62,5-%37,5 eğitim-test ayrımı üzerinden deneyler gerçekleştirilmiş, önerilen model ile üç farklı durum kategorizasyonu %83,3 doğruluk ile gerçekleştirilmiştir.

Najafi ve ark. (2020) hem üç fazlı bir asenkron motor hem de tek fazlı bir transformatörün durumlarını içeren, ayrıntılı bir termal görüntü veri setini literatüre sunmuşlardır. Ancak gerçekleştirilen çalışmada, yalnızca asenkron motor denemelerine dair başarı skorları sunulmuştur. Asenkron motor durumları yüksüz, farklı fazlar için sekiz KDD, rotor arızası ve soğutma fanı arızası olarak gösterilirken, trafo durumları yüksüz ve çeşitli yüzdelerde sekiz farklı KDD olarak tanımlanmıştır. Önerilen modelde, motor bölümü segmentasyonu için Rasgele Orman (RO) ve AdaBoost metotları kullanılmıştır. Histogram temelli birincil ve ikincil derece istatistikler işletilerek gerekli bilgilerin üretimi sağlanmış, çıkarılan özellikler Karar Ağaçları (KA) sınıflayıcısına aktarılmıştır. %50-%50 eğitim-test ayrımı (10-kat çapraz doğrulama ile eğitim) üzerinden gerçekleştirilen deneylerde, 11 asenkron motor durumu %93,8 doğrulukla kategorize edilmiştir.

Choudhary ve ark. (2020) asenkron motora dair dört farklı durumu (normal, iç halka arızası, dış halka arızası, yağlama arızası) termal görüntüler üzerinden değerlendirmişlerdir. Önerilen model kapsamında; Ayrık Dalgacık Dönüşümü (ADD) kullanarak gürültü giderme, histogram temelli analizler ve Temel Bileşen Analizi (TBA) işleterek özellik çıkarma, Mahalanobis Mesafesi (MM) ile özellik seçimi ve DVM ile sınıflama görevi gerçekleştirilmiştir. Performans değerlendirmesi için 5-kat çapraz doğrulama yöntemi kullanılmış, önerilen modelin dört koşulu %100 sınıflama doğruluğu ile ayırt ettiği görülmüştür.

Khanjani ve Ezoji (2021) asenkron motor durumlarının kategorizasyonu için istatistiksel bir model önermişlerdir. Model kapsamında özellik çıkarımı için AlexNet, termal görüntülerin sıcak / soğuk ayrımı için K-Ortalama Algoritması (KOA) ve sınıflama için DVM yöntemi değerlendirilmiştir. Çalışma kapsamında iki veri seti kullanılarak, modelin %70-%30 eğitim-test ayrımı üzerinden analizi gerçekleştirilmiştir. İlk veri seti yüklü motor için 6 durum (arızasız, fan arızası, bir fazda %10 KDD, bir fazda %30 KDD, iki fazda %10 KDD, üç fazda %10 KDD) içermiştir. İkinci veri seti ise yüksüz motor temelli 11 durumu (arızasız, rotor blokesi, fan arızası, bir fazda %10 KDD, bir fazda %30 KDD, bir fazda %50 KDD, iki fazda %10 KDD, iki fazda %30 KDD, iki fazda %50 KDD, üç fazda %10 KDD, üç fazda %30 KDD) kapsamıştır.

Çalışma neticesinde, önerilen iki aşamalı KOA-DVM modeli ile her iki veri setinde %100 sınıflama başarısı kaydedilmiştir.

Jorkesh ve Poshtan (2021), üç fazlı indüksiyon motorunda yedi farklı durumun (arızasız, %8 ve %12,5 KDD, yatak arızası, kırık rotor çubuğu, pompanın kuru çalışması, dengesiz güç kaynağı) ayrımı için etkin bir model önermişlerdir. Model kapsamında akım ve gerilim sinyallerinin gürültüsüz eldesi için Bağımsız Bileşen Analizi (BBA), özellik çıkarma işlemi için Birincil Derece İstatistikler (BDİ) ve sınıflama için Yapay Sinir Ağları (YSA) metodu kullanılmıştır. Çalışma neticesinde önerilen modelin, %98 doğruluk ile yedi durumu ayırt edebildiği görülmüştür.

Yaman (2021) bir indüksiyon motorunda ses sinyallerini işleyerek, beş farklı durumun (bir kırık rotor çubuğu, iki kırık rotor çubuğu, üç kırık rotor çubuğu, yatak arızaları, arızasız) sınıflandırmasını ele almıştır. Önerilen modelde Yerel İkili Örüntüler (YİÖ) ve ADD metotları kullanarak özellik çıkarma, Komşu Bileşen Analizi (KBA) işleterek özellik seçme ve k-En Yakın Komşuluk (k-EYK) ile sınıflama sağlanmıştır. 10-kat çapraz doğrulama temelli deneylerde, indüksiyon motoruna dair beş durumun %99,93 doğrulukla kategorize edildiği tespit edilmiştir.

Saucedo-Dorantes ve ark. (2021), bir asenkron motordaki sekiz durumun (arızasız, kırık rotor çubuğu, dengesizlik, hiza kaçıklığı, dengesizlik + hiza kaçıklığı, kırık rotor çubuğu + dengesizlik, kırık rotor çubuğu + hiza kaçıklığı, kırık rotor çubuğu + dengesizlik + hiza kaçıklığı) sınıflanmasında stator akım ve titreşim sinyallerini incelemişlerdir. Önerilen modelde zaman ve frekans boyutu özellikleri üzerinden özellik çıkarma, Genetik Algoritma – TBA (GA-TBA) ve Lineer Diskriminant Analizi (LDA) yöntemleri kullanarak özellik seçme ve YSA ile sınıflama sağlanmıştır. Model performansı 5-kat çapraz doğrulama temelli deneylerle test edilmiştir. Çalışma neticesinde önerilen model ile indüksiyon motorundaki sekiz durumun %98,25 doğrulukla ayırt edilebildiği görülmüştür.

Yi ve ark. (2022) bir asenkron motorda arızalı / arızasız durum ayrımını, akım sinyali üzerinden işlem yapan bir model ile gerçekleştirmişlerdir. Model kapsamında Stokastik Rezonans (SR) yaklaşımı kullanarak gürültü eliminasyonu, Karınca Aslanı Optimizasyonu (KAO) yöntemi işleterek özellik seçimi ve Üç Çekirdek Destek Vektör Veri Tanımlaması (ÜÇDVVT) metodu ile sınıflama gerçekleştirilmiştir. Deneylerde %50-%50 eğitim-test ayrımı üzerinden sistem performansı analiz edilmiştir. Çalışma neticesinde önerilen modelin, asenkron motora dair ikili sınıflamayı %98 doğrulukla sağladığı kaydedilmiştir.

Martinez-Herrera ve ark. (2022), asenkron motora dair beş durumun (arızasız, bir kırık rotor çubuğu, iki kırık rotor çubuğu, rulman dış bilezik hasarı, dengesiz mekanik yük) ayrımında işlevsel bir model önermişlerdir. Model kapsamında YSA girişine sunulacak veri, homojenlik ve basıklık özellikleri üzerinden elde edilmiştir. Deneylerde %70-%30 eğitim-test ayrımı ile değerlendirmeler yapılmış, önerilen model ile beş durumun %100 doğrulukla kategorize edildiği gözlemlenmiştir.

İbrahim ve ark. (2022(a)) termal görüntülerde asenkron motora dair yedi durumun (normal, iç yatak arızası, dış yatak arızası, rulman arızası, bir kırık rotor çubuğu, beş kırık rotor çubuğu, sekiz kırık rotor çubuğu) analizini ele almışlardır. Önerilen model kapsamında özellik çıkarımı için GDEM, özellik seçimi için Yabancı Ot Algoritması (YOA) ve sınıflama için XGBoost yöntemi kullanılmıştır. Önerilen modelin, 10-kat çapraz doğrulama temelli deneylerde %93 sınıflama doğruluğu sağladığı görülmüştür.

İbrahim ve ark. (2022(b)) tarafından gerçekleştirilen bir diğer çalışmada, asenkron motora dair termal görüntüler işlenerek farklı bir model önerilmiştir. Çalışma kapsamında 10 durum (normal, iç yatak arızası, dış yatak arızası, rulman arızası, bir kırık rotor çubuğu, beş kırık rotor çubuğu, sekiz kırık rotor çubuğu, iç yatak arızası + bir kırık rotor çubuğu, dış yatak arızası + beş kırık rotor çubuğu, bilyeli yatak arızası + sekiz kırık rotor çubuğu) analiz edilmiştir. Önerilen model kapsamında özellik çıkarımı için EfficientNetB0 mimarisi ve sınıflama için RO tekniği işletilmiştir. Çalışma neticesinde önerilen model ile 10-kat çapraz doğrulama temelli testlerde %97 doğruluk kaydedilmiştir.

2.3. Kaynak Araştırması Özeti

Literatürde asenkron motor ve transformatör durumlarını ele alan çalışmalar incelendiğinde, genel olarak çoğu çalışmada;

- Tekil bir ekipman üzerinden durum tespit(ler)inin gerçekleştirildiği,
- Yapıya doğrudan müdahale gerektirecek işlem süreçleri ile sınıflama sağlandığı,
- Ekipman durum tespiti için geleneksel ölçüm yöntemleri ile modellerin tasarlandığı,

- Doğrudan yapıya müdahale gerektirmeyen (uzaktan) bir ölçüm yöntemi olması sebebiyle, termal görüntülemenin etkin bir araç olarak kullanılabilceği,
- Etkin derin öğrenme / transfer öğrenme yaklaşımlarının termal görüntülerde problem üzerine kapsamlı analiz edilmediği,
- Yine derin öğrenme / transfer öğrenme yaklaşımlarının yer aldığı ve birden fazla ekipmana dair çoklu-sınıf kategorizasyonun detaylı bir şekilde ele alınmadığı gözlemlenmiştir.

Elde edilen bulgular ışığında; tez çalışmasının motivasyonunu oluşturan ve aşağıda nitelenen maddeler önemlidir. Burada bilhassa ilk madde, halen gelişime açık bir konu teşkil etmektedir:

1. Çok-sınıflı duruma sahip olan birden fazla elektrikli ekipman için termal görüntüleme üzerinden kapsamlı bir sınıflama işlemi,
2. Elektrikli ekipman(lar)a ait durumların sınıflandırılması için verimli transfer öğrenme mimarilerinin kapsamlı incelenmesi,
3. Termal görüntüleme tabanlı bir mühendislik uygulamasında transfer öğrenme mimarilerinin ayrıntılı karşılaştırılması,
4. Transfer öğrenme mimarilerine dair hiperparametre düzenlemelerinin ve bunların sistem üzerindeki etkilerinin derinlemesine analizi.

Tez konusu kapsamında yukarıda belirtilen motivasyonların, mümkün mertebe yerine getirilmesi amaçlanmıştır.

3. MATERYAL VE YÖNTEM

Bu bölüm kapsamında; ilk aşamada derin öğrenme, transfer öğrenme, Evrişimli Sinir Ağları (ESA) ve içyapısı izah edilmektedir. Sonrasında tez kapsamında ele alınan beş transfer öğrenme mimarisi (DenseNet201, MobileNetV2, ResNet50, ShuffleNet, Xception) incelenerek, içyapı izahatları görseller üzerinden gerçekleştirilmektedir. Burada mimarilerin görsel temsilleri; seyreltme (*dropout*) katmanı, tam bağlı katman veya her iki bloğu içerebilmekte, bu blokların bir veya birkaçı ile sınıflandırma katmanı elde edilmektedir. Sonrasında kullanılan veri seti özellikleri sunulmakta, en son bölümde ise tez ön çalışması için işletilen teknikler ve metrikler özetlenmektedir.

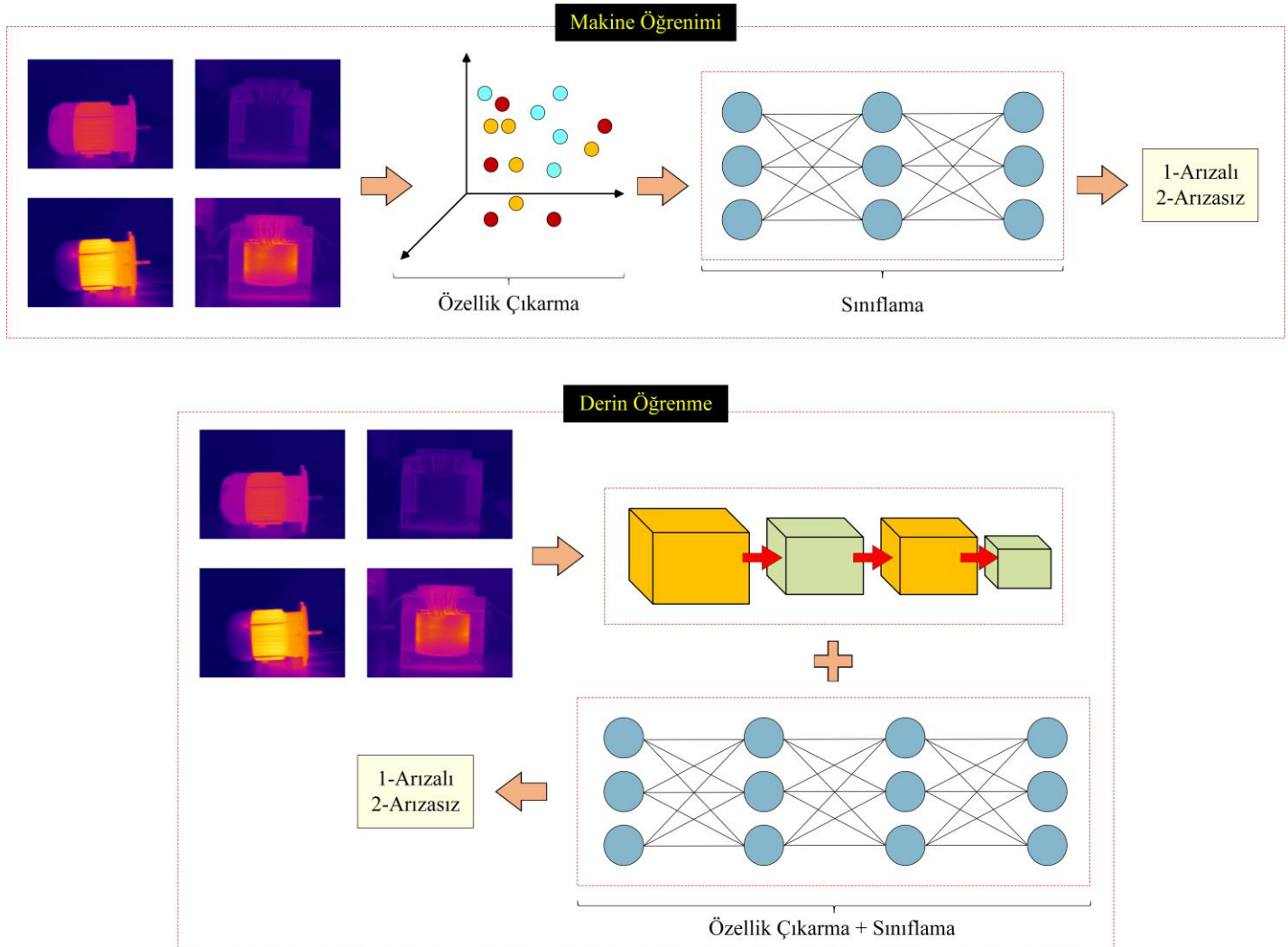
3.1. Derin Öğrenme

Derin öğrenme, insan beyninin bilgi işlem mantığından ilham alan *makine öğrenimi* bilimnin alt dalıdır. Derin öğrenme çalışma mantığı; belirli sınıflara sahip verilere dair etiket bilgilerini ve veriler arasındaki ilintiyi, büyük miktarda veri ve bu verilerin analizi ile tespit etmektir. Derin öğrenme mimarileri; farklı katman birimlerinin işletildiği yapılar olup, her katmanda anlamlı bilgilerin eldesini ve elde edilen bu bilgilerin sistem içi (katmanlar arası) yorumlanmasını temel olarak tasarlanırlar (Lu ve ark., 2015; Gupta ve ark., 2022).

Derin öğrenme kapsamında ağ yapısı, gerçekleştirilecek bir sınıflama işlemi için kendi kendine girdi özelliklerini öğrenmeye çalışır. Makine öğrenmesinde ise bu işlem özel ve ayrı bir birim (özellik çıkarma) ile gerçekleştirilir. Derin öğrenme kapsamında her katman için farklı bilgiler üretilir ve bir katmanın çıktısı diğerinin girdisini sağlar. Katmanlar boyunca çıkarılan özellikler, düşük seviyeden başlanarak katman özelliklerine göre yüksek seviyeye evrilir. Diğer bir deyişle, makine öğrenme modelleri tahminler veya kararlar sağlamak için özellik çıkarma algoritmalarını kullanırlar. Derin öğrenme ise katmanlar arası bilgi transferi ile bu görevi gerçekleştirir. Şekil 3.1’de makine öğrenme ve derin öğrenme için, trafo durum sınıflaması üzerine bir gösterim sunulmaktadır (Lu ve ark., 2015; Gupta ve ark., 2022).

Derin öğrenme mimarileri genel olarak özellik çıkarma ve transfer öğrenme görevleri için işletilir. Özellik çıkarma işlemi için eldeki yapı tekrar eğitilmelidir. Özellik çıkarımı kapsamında kullanılan derin öğrenme yöntemlerinde, önceden eğitilmiş yapı yeniden eğitilir ve probleme göre sınıflama sağlanır. Transfer öğrenme

kapsamında ise önceden eğitilen mimariye dair son üç katmanı kaldırılmakta / düzenlemekte ve istenirse performansa dayalı yeni katmanlar sisteme dahil edilebilmektedir (Lu ve ark., 2015; Gupta ve ark., 2022).



Şekil 3.1. Makine öğrenmesi ve derin öğrenme yöntemlerinde görüntü sınıflama örneği

3.2. Transfer Öğrenme

Transfer öğrenme; önceden farklı bir veri seti üzerinde eğitilen ve parametreleri ayarlanan bir derin öğrenme mimarisinin, yeni bir kategorizasyon görevine uyarlanmasıdır. Transfer öğrenme mimarileri, bellek kullanımına dair sınırlama sağlanması için etkili bir araçtır. Diğer bir deyişle transfer öğrenme mimarileri; yeni görevin, eski görevi gerçekleştiren ve deneyimli bir ESA ile sağlanması ve bu sayede performansın artırılması amacını taşır. Transfer öğrenme kapsamında ele alınan ve özel

bir işlem olan ince ayar (*fine tuning*) ise yeni eğitilecek veri kümesinin, önceden eğitilen veri kümesinden daha küçük olması ile yeni verinin daha çok eğitilmesi anlamına gelir. Böylece, kendini önceden ispat eden ESA mimarisinin yeni veri adaptasyonu sağlanmış olur (Cireşan ve ark., 2012; Yosinski ve ark., 2014; Orenstein ve Beijbom, 2017; Krizhevsky ve ark., 2017).

Transfer öğrenme mimarileri, günümüz problem çözümlerinde çok sık tercih edilmektedir. Bir problemin yeni bir ESA üzerinden çözümü; sisteme dair uzunca eğitimlerin gerçekleştirilmesi ve sistem hiperparametrelerinin detaylı incelenmesi ile gerçekleştirilebilir. Bu noktada transfer öğrenme mimarileri büyük bir kolaylık sunmaktadır. Transfer öğrenme mimarileri için ince ayar ele alınır ise hem yeni bir ESA eğitim sürecinden daha az sürede (daha hızlı) eğitim sağlanabilmekte, hem de sistem maliyetleri yeni ESA tasarımına nazaran azalmaktadır. Böylece büyük bellek kaynakları, fazlaca hesaplama gücü ve hesaplama süresi hususlarında minimizasyon sağlanabilmektedir (Cireşan ve ark., 2012; Yosinski ve ark., 2014; Orenstein ve Beijbom, 2017; Krizhevsky ve ark., 2017).

ESA mimarileri içinde ilk katmanlar üzerinden eğrilik, kenar, renk ve köşe gibi özellikler öğrenilirken, ilerleyen katmanlarda daha üst seviye özelliklerin öğrenilmesi sağlanır. Transfer öğrenme işleminde; önceden eğitilmiş ESA mimarisine dair son üç katman (tam bağlı, aktivasyon, sınıflandırma) kaldırılarak, geriye kalan bölümler genel olarak muhafaza edilir. Kaldırılan katmanlar için yeni veri üzerine düzenlemeler sağlanarak, öğrenme işleminin yeni görev adaptasyonu gerçekleştirilir (Cireşan ve ark., 2012; Yosinski ve ark., 2014; Orenstein ve Beijbom, 2017; Krizhevsky ve ark., 2017).

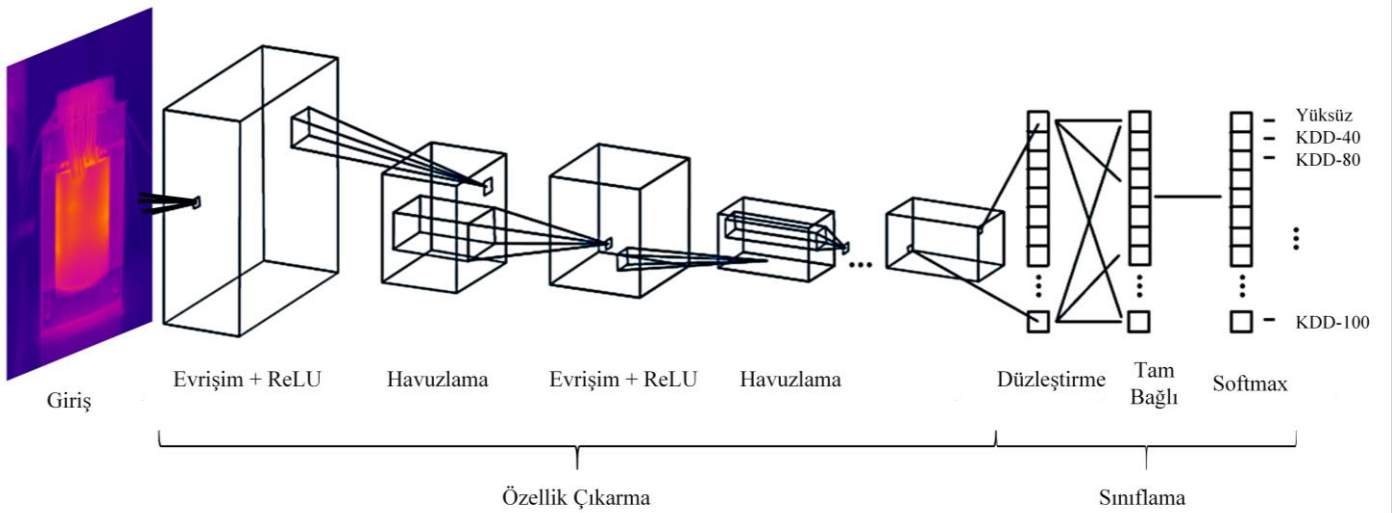
3.3. Evrişimsel Sinir Ağları

Evrişimsel Sinir Ağları (ESA), içyapısında optimizasyonu sağlayarak öğrenmeyi gerçekleştiren bir sinir ağına benzetilebilir. ESA mimarileri; yüksek kapasite ve işlem gücü sağlayan Grafik İşlem Birimi (GİB)'nin işletilmesi ile birçok hesaplama karmaşıklığı içeren işlemleri gerçekleştirebilmektedir. Burada tipik bir sinir ağı için, katman artışı ile meydana gelen hesaplama karmaşıklığı da elimine edilebilmektedir. ESA temel çalışma mantığı ise girişe sunulan veride mevcut bilgileri, içyapısında yer alan katmanlar (işlevsel filtreler) üzerinden analiz edebilmesidir. Burada ESA mimarileri; YSA yapısını içinde barındırdığı gibi, verilen bir görüntü üzerinden işlenecek örüntü bilgisini kendisi oluşturabilmektedir. Bu avantaj sayesinde ESA

mimarileri, doğrudan iki boyutlu (2B) görüntüler üzerine uygulanabilmektedir (O'Shea ve Nash, 2015).

ESA mimarisi, kendisine adını veren ve üç boyutlu (3B) düzende tanımlı evrişim katmanlarını içerir. Bu katmanlar genişlik, yükseklik ve derinlik kıstaslarına göre tanımlanır. Evrişim katmanlarının kullanımı ile özellik eldesi sağlanmakta, katman sayısının artışı ile elde edilecek özellik sayısı da artmaktadır. Artırılan bu özellik sayısı hesaplama karmaşıklığına sebebiyet verebileceğinden, faydalı azaltma olarak bilinen işlem sayesinde gereksiz özelliklerin eliminasyonu sağlanır. Böylece hem mimarinin cevap verme hızı iyileşir, hem de çıktı iyileştirilmiş olur (O'Shea ve Nash, 2015).

ESA kapsamında kullanılan yegane katmanlar; giriş, evrişim (*convolution*), doğrultulmuş doğrusal birim (*ReLU*), havuzlama (*pooling*), tam bağlı (*fully-connected*), seyreltme (*dropout*) ve sınıflama olarak ifade edilebilir. Burada herbir evrişim katmanı bir havuzlama katmanı ile ardışıl bağlanır ve böylece üst seviye özellik çıkarma işlemi için gerekler sağlanır. Doğrusal birim katmanı havuzlama öncesi veya sonrası işletilebilmektedir. ESA yapısının son aşamasında sınıflama sağlanmaktadır. Burada tam bağlı katman, seyreltme ve sınıflandırma katmanları ele alınır. Eğitim işlemi kapsamında filtrelere dair değerler güncellenmektedir ve mimari bu değerleri eğitim verisine göre analiz eder. Filtrelere sunulan girdiler; ilk olarak kenar vb. basit yapıların eldesine odaklanırken, mimarinin derinliğine göre analiz edilen özellikler daha spesifik ve üst seviye hale gelir. Şekil 3.2 kapsamında temel bir ESA yapısı ve katmanları belirtilerek, bir trafoya dair farklı durumların incelendiği örnek bir uyarılma sunulmaktadır (Gu ve ark., 2018; Tabian ve ark., 2019).



Şekil 3.2. Basit ESA yapısı ve trafo durum örneği (Tabian ve ark., 2019)

3.3.1. Giriş katmanı

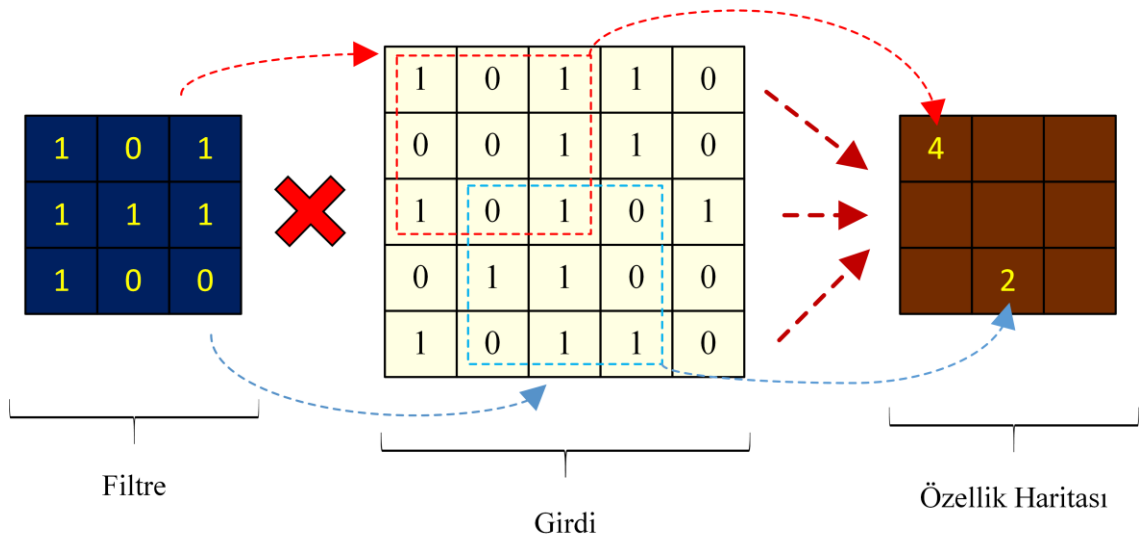
Giriş katmanı olarak isimlendirilen birim, ESA mimarisine verilerin sunulduğu bölümü simgelemektedir. Transfer öğrenme kapsamında bir ESA inceleniyor ise girdi boyutlarının, gerekli mimari giriş boyutuna göre düzenlenmesi gereklidir. Bir ESA mimarisine sunulan girdi boyutu ne kadar büyük ise mimari kapsamında incelenecek özelliklerin sayısı da o denli fazla olacaktır. Genel olarak bu durum, mimari başarısını artırabilse de sistemin cevap verme süresini de paralel olarak artıracaktır. Girdi boyutunun küçük seçilmesi ise hesaplama süresini azaltsa da sistemin gerekli öğrenmeyi az özellik üzerinden sağlayamaması durumu da sistem başarısını düşürebilmektedir. ESA giriş katmanına sunulan girdi sistemde işlenmekte, katmanlar arası bilgi aktarımı ve mimarinin veri üzerindeki etkinliğine göre sistem için tanımlı sınıflardan birisine atanmaktadır (Tabian ve ark., 2019).

3.3.2. Evrişim katmanı

Evrişim katmanı; ESA yapısına asıl ismini veren katman olup, giriş bilgisi üzerinde bir dizi filtreleme işlemine dayanır. Evrişim katmanı içinde her bir filtre, girdi bilgisindeki belirli özelliklerin eldesini sağlamaktadır. İşletilen ilk evrişim katmanında köşe, kenar gibi alt seviye özellikler tespit edilirken, katman derinliği arttıkça daha üst seviye bilginin yer aldığı özellikler elde edilir. Bu katmanda bir filtre, girdi verisi üzerinde yatay ve dikey olarak kaydırılır. Her kaydırmada filtreye karşılık gelen girdi matrisi ile filtre içi değerler çarpılır. Elde edilen değerlerin toplamı alınarak özellik haritası elde edilir. Bu işleme *evrişim* adı verilmektedir (Tabian ve ark., 2019; Ofir ve Ben-Artzi, 2022).

İşletilen filtre boyutları ESA içi farklılaşabilmektedir. Özelliklerin eldesi, özellik haritası adı verilen bilgi ile sağlanır. Bu özellik haritası ise evrişim işleminin çıktısı olarak bilinir. Girdi ve filtre boyutları sırasıyla ' $K \times K \times D$ ' ve ' $m \times m \times D$ ' olarak ifade edilirse, evrişim neticesinde üretilecek harita boyutları (yatay ve dikey temelli) ' $(K-m+1) \times (K-m+1)$ ' olarak elde edilir (Tabian ve ark., 2019; Ofir ve Ben-Artzi, 2022).

Şekil 3.3'de basit bir evrişim örneği sunulmaktadır. Burada evrişim ile üretilecek özellik haritasının, girdi boyutu 5×5 ve filtre boyutu 3×3 durumu için (yukarıda belirtilen kural gereği) 3×3 boyuta sahip olacağı görülmektedir (Tabian ve ark., 2019; Ofir ve Ben-Artzi, 2022).



Şekil 3.3. Basit evrişim örneği

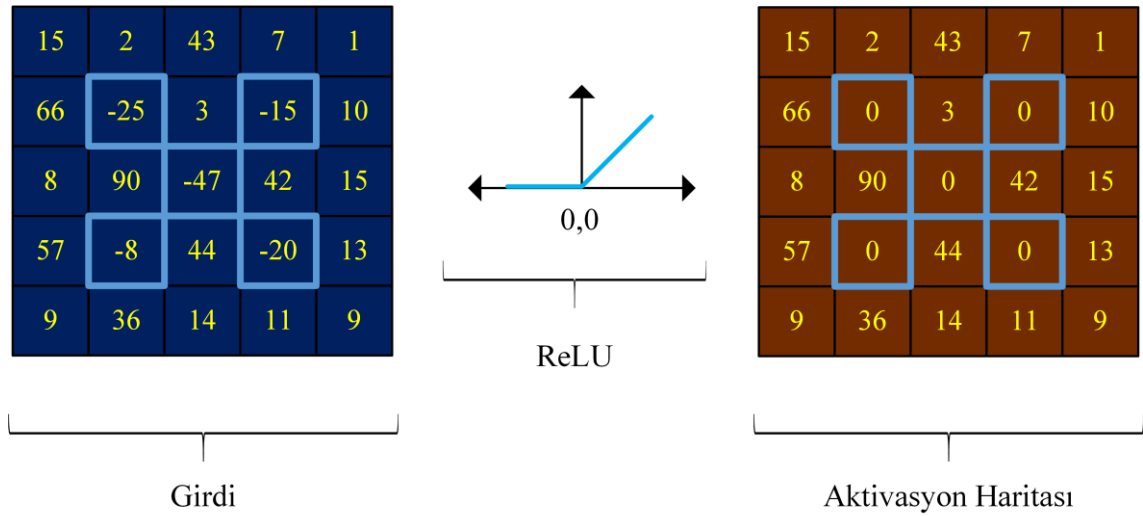
3.3.3. Doğrultulmuş doğrusal birim katmanı

Doğrultulmuş doğrusal birim katmanı, aktivasyon katmanı olarak da bilinen ve genel olarak evrişim katmanı sonrasında işletilen katmandır. Doğrultulmuş doğrusal birim, terimin İngilizce kısaltma karşılığı olan *ReLU* ile ifade edilir. Bu katmanda gerçekleştirilen işlem; girdi verisine Eşitlik (3.1)'de tanımlı fonksiyon üzerinden bir dönüşüm uygulanmasını temel alır (Hijazi ve ark., 2015).

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3.1)$$

Katmanın temel amacı, Eşitlik (3.1)'de de görüldüğü üzere girdi verisindeki negatif değerler için sıfır değerinin atanmasıdır. Böylece girdi verisindeki negatif değerlerin eliminasyonu sağlanır (Hijazi ve ark., 2015).

ESA mimarilerinde aktivasyon haritasının eldesi için, *ReLU* harici farklı aktivasyon katmanları da işletilebilmektedir. Bu noktada *ReLU* fonksiyonu, hem geriye yayılım hem de ileri besleme görevlerinde hızlı işlem yaptığı için sıklıkla tercih edilir. Şekil 3.4'de bir girdi verisi üzerine *ReLU* uygulaması gösterilmektedir. Girdi matrisinde yer alan beş adet negatif değer, *ReLU* fonksiyonunun uygulanması sonrası sıfır değerine çekilmektedir (Hijazi ve ark., 2015).



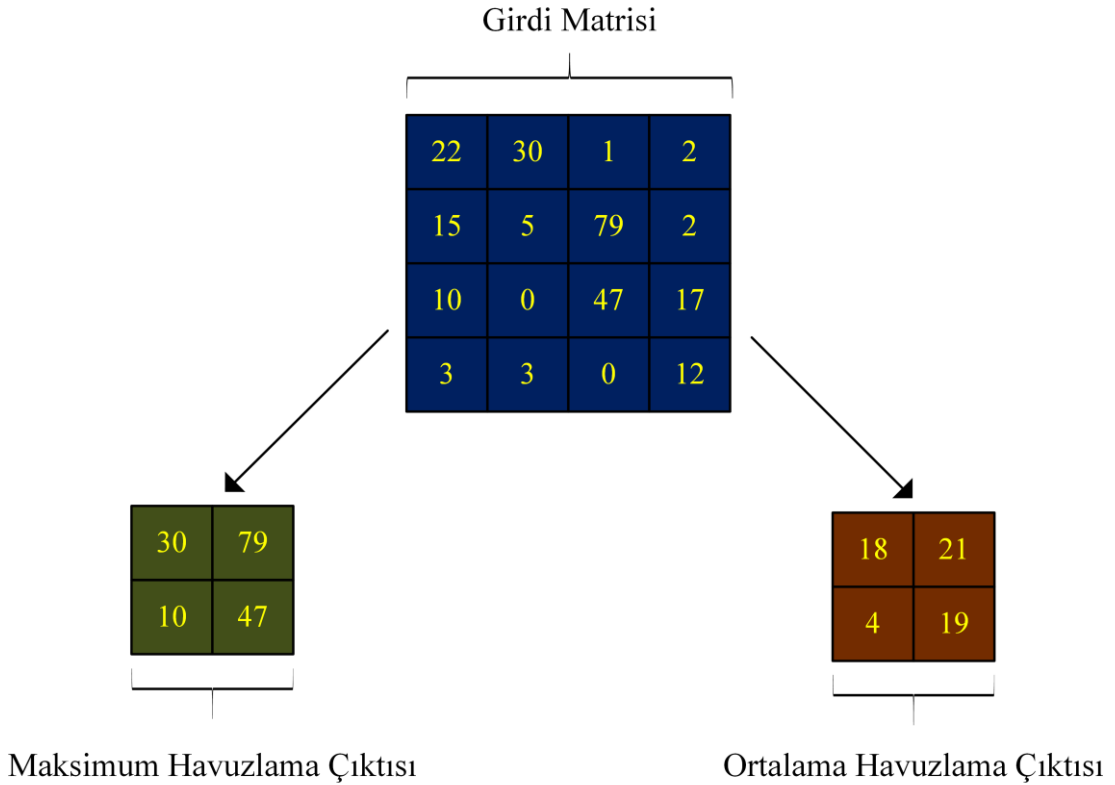
Şekil 3.4. ReLU fonksiyonunun bir girdi matrisine uygulanması

3.3.4. Havuzlama katmanı

Havuzlama katmanı, aktivasyon katmanı veya doğrultulmuş doğrusal birim katmanı sonrasında uygulanan katmandır. Bu kapsamda girdi verisinin, farklı (azaltılmış) boyutlarda ve özelliklerde formlarının eldesi sağlanır. Diğer bir deyişle, ESA içi boyut azaltma (özellik azaltma) işleminin uygulanmasını sağlayan birimdir (Hijazi ve ark., 2015).

Havuzlama katmanındaki havuzlama işlemi, aşağı (alt) örnekleme olarak da isimlendirilmektedir. Bu amaçla; büyük boyutlu girdi matrisine belirli boyutta tanımlı bir filtre uygulanarak, girdi verisinden daha küçük boyuta sahip yeni bir matris tanımlanır. Bu boyut azaltma işlemi faydalı azaltma olarak isimlendirilir ve ESA içi bilginin daha hızlı değerlendirilmesine olanak sağlar. Havuzlama kapsamında maksimum, ortalama, toplam gibi filtre tipleri uygulanabilmektedir. En sık kullanılan havuzlama tipi maksimum havuzlama olarak bilinir. İkinci en sık kullanıma sahip olan havuzlama tipi ise ortalamadır. Maksimum ve ortalama havuzlama, girdi verisi ile filtrelerin eşleştiği lokasyonda yer alan bilgiler için sırasıyla maksimum değerini ve ortalama değerini eldesi üzerinden gerçekleştirilir (Hijazi ve ark., 2015).

Şekil 3.5 kapsamında; 2×2 boyuta sahip maksimum ve ortalama filtrelerin, 4×4 boyutlu bir girdi verisine uygulanması sonucu temel havuzlama işleminin nasıl gerçekleştirildiği sunulmaktadır. Burada faydalı azaltma işlemi, girdi matrisinden belirli fonksiyonlar / işlemler ile anlamlı bilgi eldesini nitelemektedir (Hijazi ve ark., 2015).



Şekil 3.5. Maksimum ve ortalama havuzlama için bir örnek gösterim

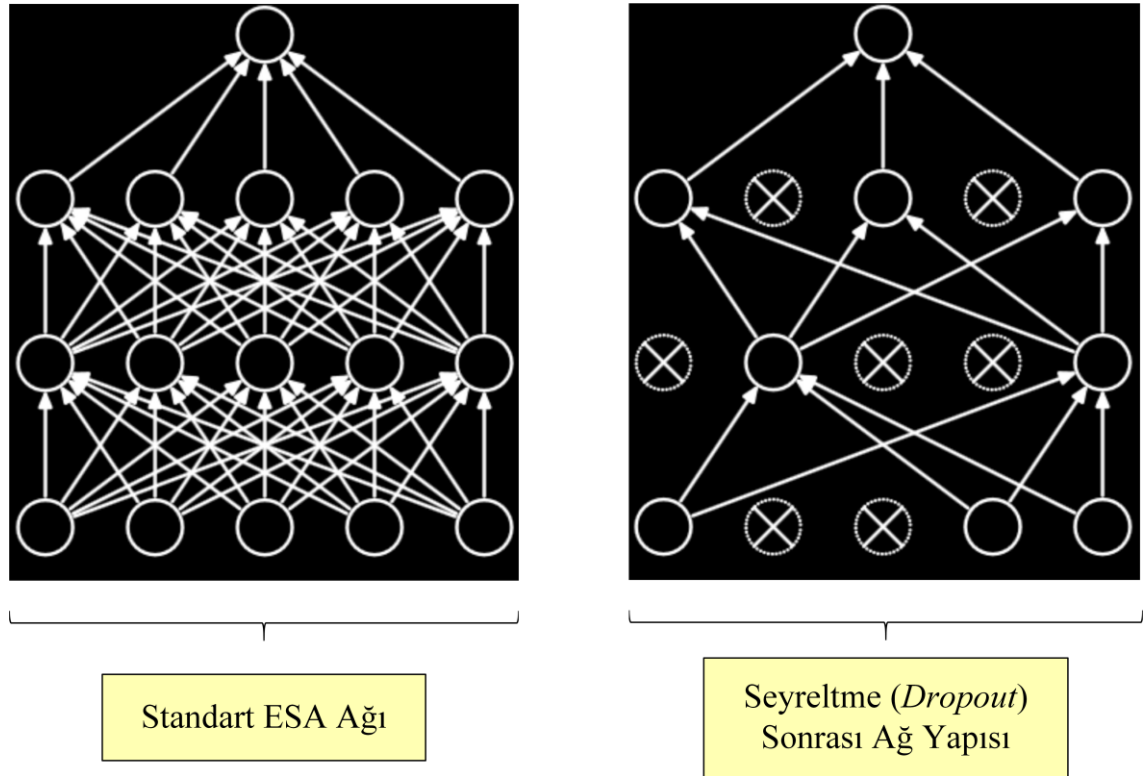
3.3.5. Tam bağlı katman

ESA kapsamında sınıflandırma işlemi tam bağlı katman üzerinden değerlendirilir. ESA mimarisinde bu katman öncesi özellik çıkarma işlemi sağlanırken, son havuzlama katmanı sonrası sınıflamada işletilecek olan bilgi bu katman girişine beslenir. Tam bağlı katman içerisinde kapsamlı YSA yapısı bulunmaktadır. Burada işletilen YSA yapısı, makine öğrenimi kapsamında işletilen YSA yapılarından farklılık gösterir ve bu farklılık gizli katman sayısındaki artıştır. ESA özellik çıkarma birimi üzerinden sağlanan özellikler, bu katmanda yer alan derin sinir ağı yapısı ile kapsamlı analiz edilmektedir. Bu katmanda eğitim için sunulan veriler analiz edilerek, çıkış eldesi için gerekli bilgilerin üretilmesi sağlanır (Hijazi ve ark., 2015; Özkan ve Ülker, 2017).

3.3.6. Seyreltme katmanı

ESA mimarileri, tipik YSA yapısından daha çok katman içermektedir. Bu nedenle incelenecek veri sayısı (dolaylı olarak veri analizi), katman sayısındaki artış ile paralel artış gösterir. Buradaki artış ise mimariye ek bir hesaplama maliyeti oluşturur.

Ayrıca fazla miktarda işlenecek veri girdisi, mimarının bu veriyi ezberlemesine sebebiyet verebilmektedir. Bu nedenle ESA içi işletilen YSA yapısına dair bazı nöronlar (düğümler) kaldırılır ve bu işlem seyreltme olarak isimlendirilir. Seyreltme işlemi, hesaplama maliyetinin azaltılmasını ve ağın veriyi ezberlemesini önlemektedir. Şekil 3.6'da tipik bir ESA ağı ile rasgele seyreltme uygulanmış ağ yapısına dair basit bir gösterim sunulmaktadır (Özkan ve Ülker, 2017; Apaydin ve ark., 2020).



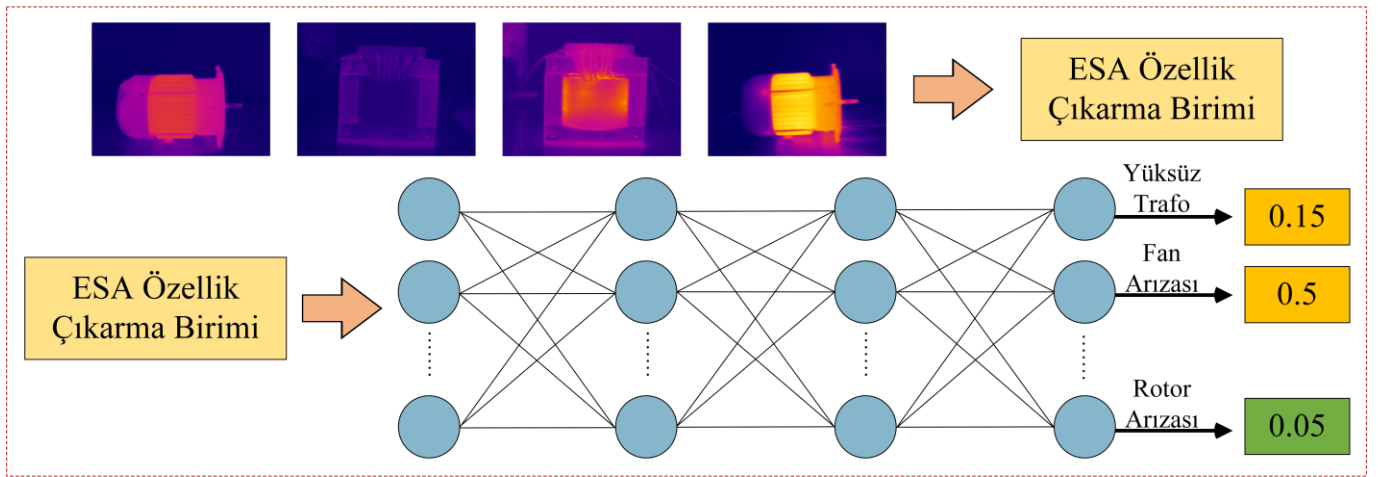
Şekil 3.6. Evrişimsel sinir ağına dair seyreltme örneği (Apaydin ve ark., 2020)

3.3.7. Sınıflandırma katmanı

ESA mimarisinde kategorizasyon işlemi sınıflandırma katmanında sağlanır. Bu katman; tam bağlı ve seyreltme katmanları sonrası işletilmekte, bütün mimarının cevabını teşkil edecek olan sistem çıktısını üretmektedir. Katman kapsamında çeşitli fonksiyonlar işletilerek sistem cevabı belirlenmektedir. Ancak en sık kullanılan fonksiyon, Eşitlik 3.2'de belirtilen *Softmax* fonksiyonudur (Es-Sabery ve ark., 2021).

$$f(x) = \frac{e_i^x}{\sum_{k=1}^K e_i^x} \quad (3.2)$$

Eşitlik (3.2) kapsamında x , e_i^x ve K değişkenleri sırasıyla girdiyi, girdi için eksponansiyel çıktıyı ve veride kullanılan etiket sayısını temsil eder. *Softmax* fonksiyonu için etiket sayısı kadar çıkış tanımlıdır. *Softmax* katmana sunulan nümerik değerler, bu katman sonrası her bir sınıf için olasılıklara dönüştürülmektedir. Her bir etiket için belirli bir olasılık değeri $[0,1]$ aralığında tanımlanır ve bu olasılıkların toplamı '1' değerine eşittir. Girdi verisine dair sınıf ise elde edilen en yüksek olasılık değerine göre belirlenir. Şekil 3.7'de sınıflamaya dair *softmax* çıktıları, asenkron motor ve trafo durumlarının birleştiği veri için bir örnek gösterimle izah edilmektedir (Hijazi ve ark., 2015; Bhandare ve Kaur, 2021; Es-Sabery ve ark., 2021).



Şekil 3.7. Asenkron motor ve trafo görüntüleri için softmax fonksiyonu çıktı örneği

3.3.8. Optimizasyon teknikleri

Derin öğrenme kapsamında, tahmini çıkış ile elde edilmesi gereken (gerçek) çıkış arası fark (hata değeri) minimize edilmeye çalışılır. Bu amaçla ESA içi optimizasyon teknikleri işletilmekte, YSA yapısında yer alan ağırlık değerleri için optimum atama amaçlanmaktadır. Böylece hata (kayıp fonksiyonu) mümkün mertebe düşük tutulabilmektedir. Gradyan iniş yöntemi, hata azaltmada en sık kullanılan metottur. Literatürde sık kullanılan gradyan iniş temelli algoritmalara *sgdm*, *rmsprop* ve *adam* yaklaşımları örnek gösterilebilir (Jiang ve ark., 2020).

Sgdm yaklaşımı, tüm parametrelere tekil bir öğrenme oranı üzerinden işlem yapmaktadır. *Rmsprop* yaklaşımı; *adam* ve *momentum* yöntemlerinin temelini teşkil etmekte, *adagrad* algoritmasındaki öğrenme katsayısının azalması dezavantajına dair

önerilmektedir. *Adam* yaklaşımı ise bir *momentum* terimi işleterek güncelleme görevini yerine getirir (Jiang ve ark., 2020).

3.4. Transfer Öğrenme Yöntemleri

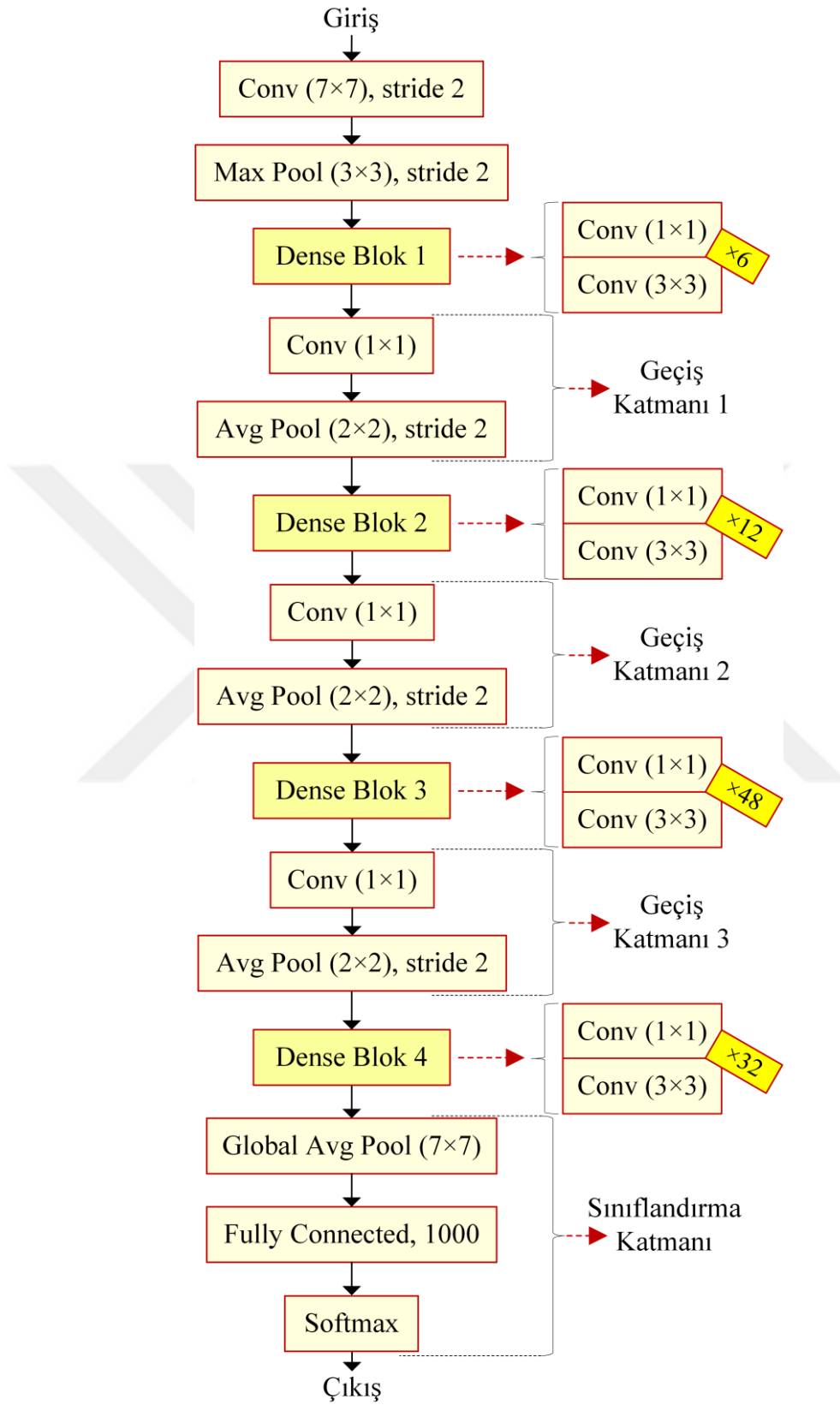
Tez çalışması kapsamında kullanılan 5 adet transfer öğrenme mimarisi bulunmaktadır. Bu bölüm kapsamında mimarilere dair görsel temsiller ayrıntılı bir biçimde ele alınmaktadır.

Mimarilerin gösterimlerinde ESA içi sınıflama aşaması için tam bağlı (*fully-connected*), seyreltme (*dropout*) veya bu iki işlemin de yer aldığı bloklar kullanılmaktadır. Bu üç durum da YSA tekniğine dair kullanımı temsil etmekte, bazı mimarilerde ise bu gösterimler yukarıda belirtilen üç durumdan biri ile ifade edilmektedir.

3.4.1. DenseNet201 mimarisi

DenseNet201, ileri beslemeli işletim kuramına dayanan Yoğun Bağlantılı Evrişimli Ağlar (*DenseNet*)'in etkin bir türevidir (Huang ve ark., 2017). *DenseNet* kökeninde mimari, kaybolan gradyan problemini en aza indirmeyi amaçlar ve özellik yayılımını geliştirmek için, mevcut girdi verilerini sonraki katmanlara girdi olarak sunar. Burada *DenseNet*; bütün bir sistemin girdiyi daha iyi anlamasını sağlamak için, önceki katmanların girdisi veya dönüştürülmüş girdi (özellik haritası) ile sonraki katmanları yetkilendiren yoğun (*dense*) blokları kullanır. Özetle; *DenseNet* tabanlı mimariler, özellik haritalarının yeniden kullanımını dikkate alır ve daha yüksek performans eldesi için (daha derin bir mimari tasarımı sağlayan) kısa bağlantıları işletir. DenseNet201 mimarisi, Şekil 3.8 kapsamında görüldüğü gibidir (Huang ve ark., 2017; Jaiswalve ark., 2021).

DenseNet201 mimarisi giriş görüntüsünü değerlendirmek için evrişim katmanları, çok bağlantılı evrişim katmanlarına sahip yoğun (*dense*) blokları, ortalama (*average*) ve maksimum (*max*) havuzlama yaklaşımlarını, tam bağlı katmanı ve bir *softmax* fonksiyonunu içermektedir. DenseNet201'de, geçiş katmanında azaltılan ve yoğun bir bloğun girişine beslenecek gerekli öznelik haritasını tanımlamak için, evrişim bloğu ve ortalama havuzlama kullanılarak geçiş (*transition*) katmanları elde edilir.



Şekil 3.8. DenseNet201 mimarisi

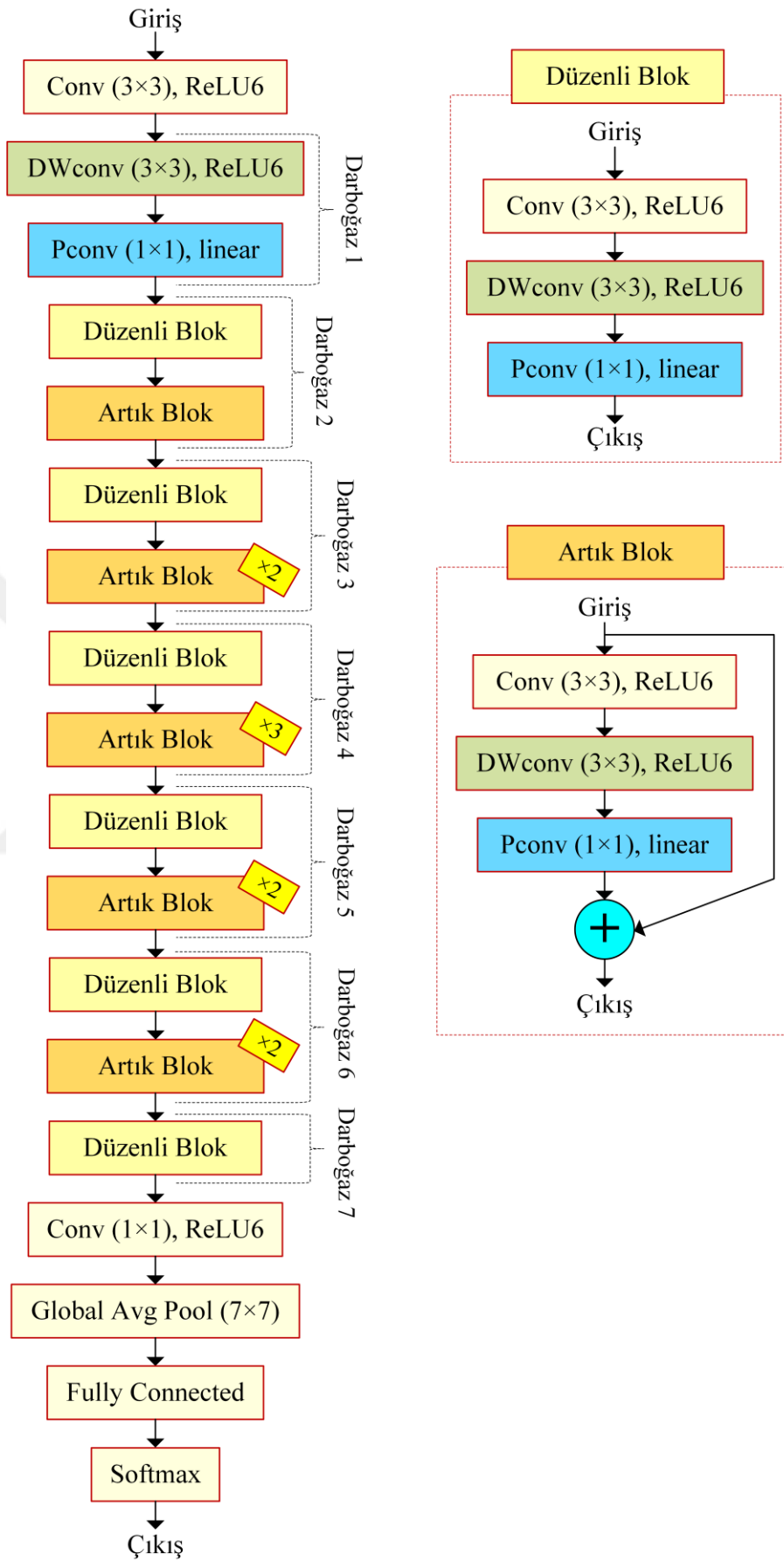
3.4.2. MobileNetV2 mimarisi

Hafif (*Lightweight*) *MobileNet* mimarilerinin verimli bir türevi olan MobileNetV2, temel tasarıdan daha yüksek performans sağlamak için belirli olguların işletilmesini temel alır. Burada mimari; evrişim katmanları, derinlemesine evrişim (*Depth-Wise convolution: DWconv*) katmanları, noktasal evrişim (*Pointwise convolution: Pconv*) katmanları, üç tip evrişim içeren düzenli (*regular*) ve artık (*residual*) blokları, ortalama (*average*) havuzlamayı, tam bağlı katmanı ve *softmax* fonksiyonunu içermektedir. MobileNetV2 mimarisine dair akış şeması Şekil 3.9'da görüldüğü gibidir (Sandler ve ark., 2018; Wang ve ark., 2019; Marin ve ark., 2021).

Mimaride derinlemesine ayrılabilir evrişim, ilk aşamada derinlemesine evrişim (*DWconv*) ve ardından noktasal evrişim (*Pconv*) kullanılarak sağlanır. Derinlemesine evrişimde, tek bir evrişim çekirdeği her giriş kanalı için ayrı ayrı işlenir ve filtreleme gerçekleştirilir. Noktasal evrişimde ise derinlemesine evrişim çıktılarının lineer eldesi için çekirdek boyutu 1×1 atanır. MobileNetV2 kapsamında derinlemesine evrişim çekirdek boyutu, MobileNetV1'e benzer şekilde 3×3 atanarak hesaplama maliyetleri en aza indirilir (Sandler ve ark., 2018; Wang ve ark., 2019; Marin ve ark., 2021).

Mimarinin tüm blokları (*Pconv* hariç) aktivasyon fonksiyonu olarak *ReLU6* işletir. Burada *ReLU6*, *ReLU* fonksiyonunun geliştirilmiş bir versiyonu olarak karşımıza çıkar ve maksimum çıkış değerini '6' ile sınırlıdır (Sandler ve ark., 2018; Wang ve ark., 2019; Marin ve ark., 2021).

MobileNetV2'nin diğer iki düzenlemesi, lineer darboğazlar (*bottleneck*) ve atlamalı bağlantılar olarak bilinir. Şekil 3.9'da görüldüğü gibi düzenli ve artık blokların kombinasyonu, darboğaz (*bottleneck*) modüllerinin çalışması olarak ifade edilir. Artık bloklar, evrişim tabanlı işlemlere ek olarak (normal bloklardan farklı olarak) atlama bağlantısını içerir. Burada verimli özellik haritaları önerilerek, bilgi kaybını önlemek için atlama bağlantıları işletilir. Evrişim tabanlı işlemlerde veri genişletme, veri filtreleme ve veri sıkıştırma için sırasıyla üç tip evrişim (*Conv*, *DWconv*, *Pconv*) kullanılır. Mimari kökeninde artık bloklar, dar katmanlar arasında gerçekleştirilen atlama bağlantılarının bağlantı şekli nedeniyle *ters artık bloklar* olarak da isimlendirilebilir (Sandler ve ark., 2018; Wang ve ark., 2019; Marin ve ark., 2021).



Şekil 3.9. MobileNetV2 mimarisi

3.4.3. ResNet50 mimarisi

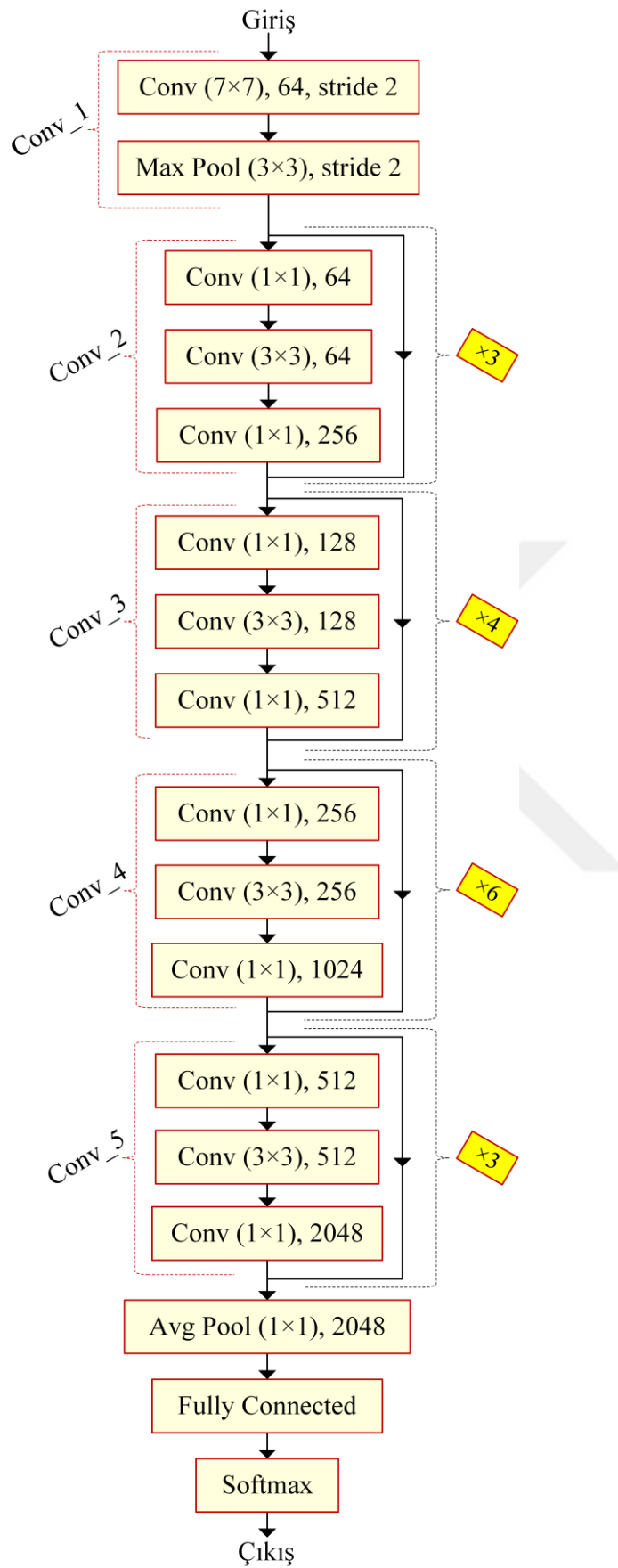
Daha yüksek katman sayısı, bir ESA mimarisinde sistem performansını artırabilmektedir. Burada ESA mimarisi, mevcut sürümden daha derin tasarımlar ise eğitim sürecinde sistem doğruluğu azalabilir. Bununla ilgili olarak; ResNet50 dahil olmak üzere *ResNet* tabanlı mimariler, daha derin ağlarda kaybolan gradyan problemini ortadan kaldırmak amacıyla önerilmektedir (Rao ve ark., 2021; He ve ark., 2016; Kim ve ark., 2021).

Şekil 3.10 kapsamında ResNet50 mimari tasarımı sunulmaktadır. ResNet50 mimarisi; eşit boyutta tanımlı çekirdeklere (*kernel*) sahip evrişimlerin yer aldığı beş evrişim bloğunu (*Conv_1, ..., Conv_5*), ortalama (*average*) ve maksimum (*max*) havuzlama yaklaşımlarını, tam bağlı katmanı ve *softmax* fonksiyonunu içermektedir (Rao ve ark., 2021; He ve ark., 2016; Kim ve ark., 2021).

Artık Ağlar (*ResNets*) mantığından ilham alan veya kullanan mimariler (MobileNetV2, ResNet101, vb.), performansı yüksek seviyelere çıkarmak ve daha derin katmanlardaki bilgi kaybını önlemek için artık (atlamalı) bağlantıları dikkate alır. *ResNet* tabanlı mimarilerde, birkaç katmanı atlamak ve bir katmanın çıktısını sonraki katmanlara bağlamak için artık bağlantılar işletilir. Ayrıca atlama bağlantılarının kullanımı için, mimaride herhangi bir ek parametreye gerek duyulmaz. Diğer bir deyişle artık bloklar, katmanlar arasındaki bağlantıları atlayarak bypass işlemi gerçekleştirir. Böylece daha etkili öznitelik haritaları üretilerek, mimarinin daha fazla eğitim hatası alması önlenmektedir. Bu şekilde *ResNet* tabanlı mimariler; katman geçişleri arasındaki bilgi kaybını engelleyerek, kaybolan gradyan problemini en aza indirmeyi amaçlar (Rao ve ark., 2021; He ve ark., 2016; Kim ve ark., 2021).

3.4.4. ShuffleNet mimarisi

ShuffleNet mimarisi, düşük hesaplama karmaşıklığı ile ilgili olarak mobil uygulamalarda verimli bir mimari olarak ortaya çıkmıştır. Eşit koşullarda ShuffleNet; ResNet ve ResNetX mimarilerinden daha az karmaşıklık sağlar, çok daha yüksek doğruluk skorları elde eder ve AlexNet mimarisinden daha üstün bir performans sergiler (Zhang ve ark., 2018).

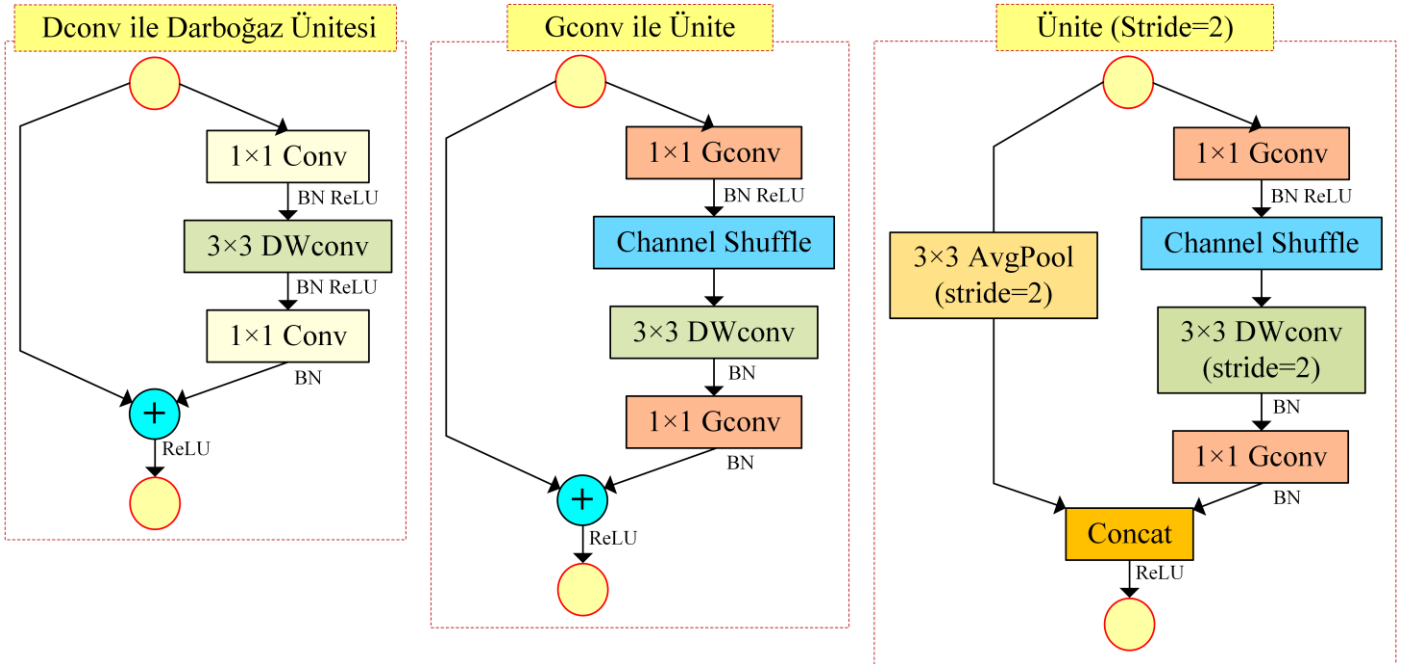


Şekil 3.10. ResNet50 mimarisi

ShuffleNet'in tasarım mantığı; darboğaz (*bottleneck*) fenomeni, artık (*residual*) bloklar, derinlemesine evrişim (*DWconv*) ve noktasal grup evrişimi (*Gconv*) olmak üzere dört kavramdan oluşur. Burada daha derin modellere ulaşmak ve mimarinin hesaplama maliyetlerini azaltmak için yalnızca dört kavram ele alınmaktadır. Çizelge 3.1 ve Şekil 3.11 sırasıyla ShuffleNet mimarisini ve mimari modüllerini göstermektedir (Zhang ve ark., 2018; Jamali ve ark., 2021).

Çizelge 3.1. ShuffleNet mimarisi

Katman	Çıkış Boyutu	Çekirdek Boyutu	Stride Değeri	Tekrarlanma	Çıkış Kanalları				
					g=1	g=2	g=3	g=4	g=5
İmge	224×224				3	3	3	3	3
Conv1	112×112	3×3	2	1	24	24	24	24	24
MaxPool	56×56	3×3	2						
Stage2	28×28		2	1	144	200	240	272	384
	28×28		1	3	144	200	240	272	384
Stage3	14×14		2	1	288	400	480	544	768
	14×14		1	7	288	400	480	544	768
Stage4	7×7		2	1	576	800	960	1088	1536
	7×7		1	3	576	800	960	1088	1536
GlobalPool	1×1	7×7							
Fully Connected					1000	1000	1000	1000	1000
Karmaşıklık					143M	140M	137M	133M	137M



Şekil 3.11. ShuffleNet modülleri

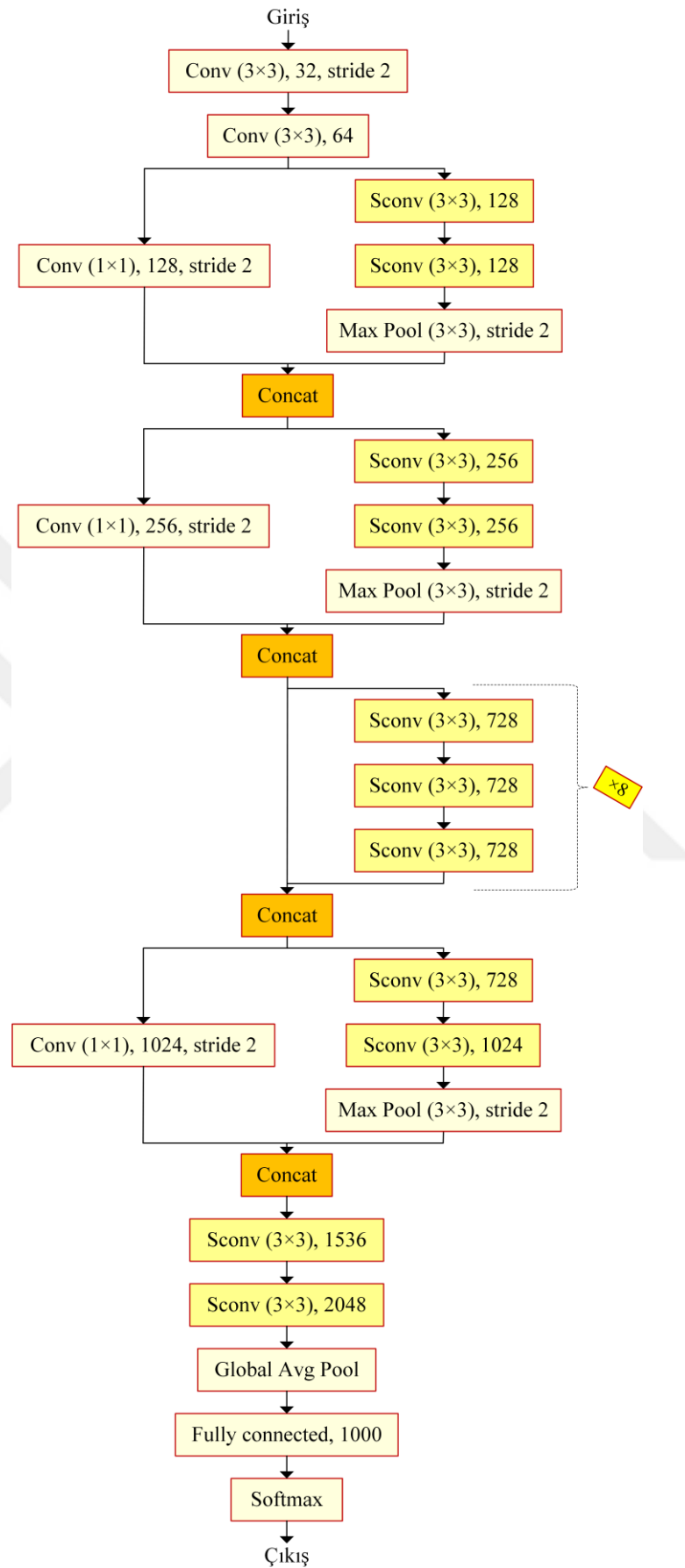
ShuffleNet mimarisi kapsamında evrişim katmanları, derinlemesine evrişim katmanları (*DWconv*), noktasal grup evrişim katmanları (*Gconv*), filtre birleştirme (*filter-concat* veya *concat*), kanal karıştırma (*channel shuffle*) bloğu, ortalama (*average*) ve maksimum (*max*) havuzlama yaklaşımları ile tam bağlı bir katman işletilmektedir.

Atlama bağlantıları içeren darboğaz birimleri, sistem performansını yükselterek daha yetkin özellik haritaları sağlamaktadır. Grup evrişimi; mimari başarısını etkili bir şekilde artırabilen farklı özellik düzeylerini elde etmek için, bir katmana birden çok çekirdek uygulanmasını temel alır. Ayrıca grup evrişimi, hesaplama süresini azaltarak paralel Grafik İşlem Birimleri (GİB)'nin aynı anda kullanımını sağlar. Kanal karıştırma, daha sağlam kanal grupları elde etmek ve mimarinin girdi-çıkışı davranışını güçlendirmek için üç aşamalı bir süreç (yeniden şekillendirme, transpoze etme, düzleştirme) içermektedir. Derinlemesine evrişim, ek hesaplama yükünü ortadan kaldırmak için yalnızca darboğaz katmanlarında uygulanır (Zhang ve ark., 2018; Jamali ve ark., 2021).

3.4.5. Xception mimarisi

Ekstrem başlangıç (*Extreme Inception: Xception*) mimarisi; başlangıç (*inception*) fenomeninin yanı sıra evrişim blokları, ayrılabilir evrişim blokları, artık (*residual*) bağlantılar ve mimari içi uyumluluk gözetilerek önerilmektedir. Şekil 3.12 kapsamında Xception mimarisi sunulmaktadır (Chollet, 2017; Leonardo ve ark., 2018).

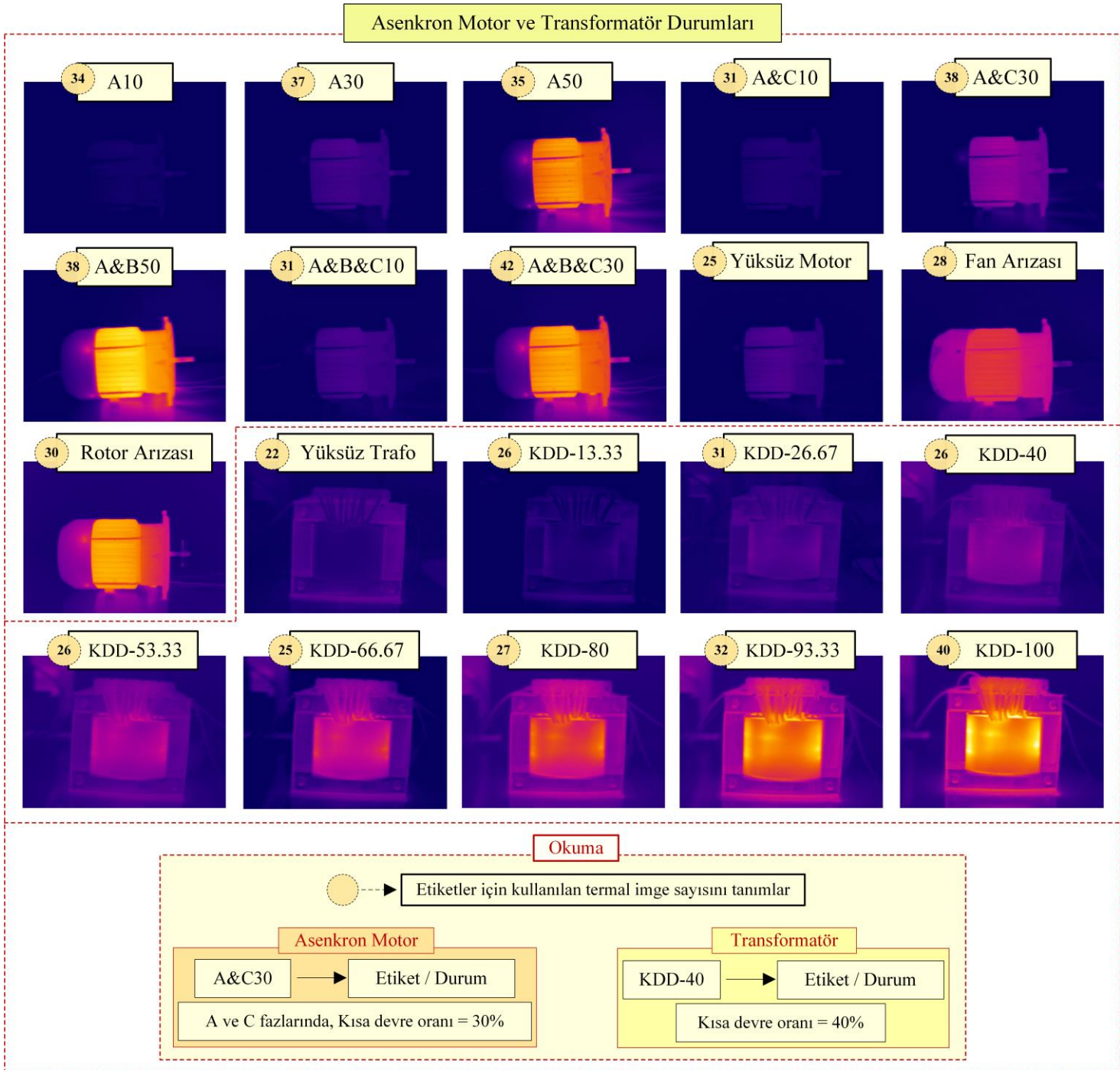
Xception mimarisi; evrişim blokları, iyileştirilmiş – derinlemesine ayrılabilir evrişim blokları (*Sconv*), filtre birleştirme (*filter-concat* veya *concat*), ortalama (*average*) ve maksimum (*max*) havuzlama yaklaşımları, tam bağlı katman ve *softmax* fonksiyonunu içermektedir. Bu noktada Xception, *inception* modülleri yerine iyileştirilmiş – derinlemesine ayrılabilir evrişimleri kullanır. Burada mimari; *inception* temelli ağların mantığında olduğu gibi, kanallar arası ve uzamsal korelasyonları ayırmayı amaçlar. Derinlemesine ayrılabilir evrişim; her kanal için uzamsal bağıntıyı ayrı ayrı haritalayan ve kanallar arası bağıntıyı 1×1 derinlemesine evrişim ile yakalayan özel bir dönüşümdür. Mimarideki iyileştirilmiş – derinlemesine ayrılabilir evrişim için ilk olarak noktasal evrişim, ikinci adımda ise derinlemesine evrişim işletilir. Ayrıca Xception mimarisinde, *ResNet* temelli mimarilerde olduğu gibi atlama bağlantıları kullanılarak, eğitim hatasının düşük seviyelerde tutulması sağlanır (Chollet, 2017; Leonardo ve ark., 2018).



Şekil 3.12. Xception mimarisi

3.5. Kullanılan Veri Seti Özellikleri

Kullanılan veri seti, hem asenkron motorların hem de transformatörlerin durumlarını inceleyen Najafi ve ark. (2020)'nin çalışmalarında sunulmaktadır. Ancak bu çalışmada sınıflama işlemi, iki ekipman için de ayrı ayrı gerçekleştirilmektedir. Tez çalışmasında ise bu veriler birleştirilerek, 20 durumun doğrudan sınıflandırılması ele alınmaktadır. Birleştirilmiş veri seti Şekil 3.13'de sunulmaktadır.



Şekil 3.13. Veri seti bilgileri

Şekil 3.13’de görüldüğü gibi veri seti, asenkron motora ait 11 durumu ve transformatöre bağlı 9 durumu içermektedir. Motor koşulları sırasıyla bir fazda %10 KDD, bir fazda %30 KDD, bir fazda %50 KDD, iki fazda %10 KDD, iki fazda %30 KDD, iki fazda %50 KDD, üç fazda %10 KDD, üç fazda %30 KDD, yüksüz motor, soğutma fan arızası ve rotor arızası olarak nitelenmektedir. Transformatör koşulları ise sırasıyla yüksüz transformatör, %13,33 KDD, %26,67 KDD, %40 KDD, %53,33 KDD, %66,67 KDD, %80 KDD, %93,33 KDD ve %100 KDD şeklinde belirtilmektedir.

Durumların görsel olarak incelenmesine göre sınıflandırma için gerekli olan bilgi, termal görüntülemenin temel amacını temsil eden ısı değişimidir. Farklı ekipmanlara ait KDD içeren tüm etiketlerin kombinasyonu, rotor arızası, soğutma arızası ve iki farklı yüksüz durumun eklenmesi ile 20 farklı durumu içeren kapsamlı bir veri seti elde edilmektedir. Önerilecek / kullanılacak bir sınıflandırma yapısının, termal görüntüler üzerinden bütün sınıfları ayırt edebilmesi için kararlı bir yapıda olması gereklidir. Burada sınıflayıcı yapı;

- İki ekipmana dair KDD ayırımını sağlamalı ve bu ayırmaları birbiri ile karıştırmamalıdır.
- Rotor ve fan arızalarını ayırt ederek, bu arızaları KDD arızaları ile karıştırmamalıdır.
- İki yüksüz durumu da birbirinden ayırt edebilmelidir.

3.6. Tez Ön Çalışmasında Kullanılan Teknikler ve Metrikler

Bu bölümde; tez ön çalışmasında ele alınan istatistiksel model için, yapı içi işletilen metotlara değinilmektedir. Model kapsamında; termal görüntülerden altı adet BDİ özelliği çıkarılarak, elde edilen özniteliklerin temel sınıflayıcılara sunulması sağlanmaktadır.

3.6.1. Birincil derece istatistikler

Birincil Derece İstatistikler (BDİ), bir görüntü histogramından veya yoğunluk değerlerinden elde edilen analizlerdir. En sık kullanılan BDİ parametreleri; ortalama, standart sapma, eğrilik, basıklık, enerji ve entropi şeklinde sıralanabilir. Bu özelliklerin eldesinde 2B bir görüntü tanımlaması bir $f(x,y)$ fonksiyonu ile sağlanabilir. Burada 2B

görüntünün yatay ve dikey koordinatları, $(x=0,1,\dots,X-1)$ ve $(y=0,1,\dots,Y-1)$ olmak üzere iki değişken temelli ifade edilebilir. ‘ G ’ görüntünün bir yoğunluk değerini gösterir ise fonksiyon çıktısı olan ‘ i ’ (ayrık yoğunluk) değeri $(i=0,1,\dots,G-1)$ kapsamında değer alabilmektedir. Histogram ise görüntü boyunca ‘ G ’ değerlerinin tekrar sayıları olarak tanımlanır (Sakalli ve Koyuncu, 2022).

2B uzayda görüntü boyutu genişlik ve yükseklikten oluşurken, 3B uzayda bu değişkenlere dilim sayısı eklenir. Burada ‘ M ’, ‘ N ’ ve ‘ L ’ değişkenleri sırasıyla dilimlerin genişliğini, uzunluğunu ve sayısını gösterirse, İlgi Hacim (*Volume of Interest: VOI*) için toplam voksel sayısı bu değişkenler çarpılarak bulunur. Histogram ve Kronecker Delta fonksiyonu sırasıyla $h(i)$ ve $\delta(j,i)$ temsilleri ile gösterilirse, bu fonksiyonların çıkışları sırasıyla Eşitlik (3.3) ve Eşitlik (3.4)’e göre elde edilir (Sakalli ve Koyuncu, 2022).

$$h(i) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \delta(f(x, y), i) \quad (3.3)$$

$$\delta(j, i) = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases} \quad (3.4)$$

Yoğunluk değeri ‘ i ’ için Olasılık Yoğunluk Fonksiyonu (OYF) çıkışı $p(i)$; Eşitlik (3.5)’te görüldüğü üzere, her voksel yoğunluk değerinin VOI’deki toplam voksel sayısına bölünmesi ile ifade edilir (Sakalli ve Koyuncu, 2022).

$$p(i) = \frac{h(i)}{1 * L * M * N}, \quad i = 0, 1, 2, 3, \dots, G-1 \quad (3.5)$$

Eşitlik (3.5)’teki OYF işletilerek, bir VOI’nin BDİ özellikleri ifade edilebilmektedir. Eşitlik (3.6) – Eşitlik (3.11) arası sunulan sırasıyla ortalama (μ), standart sapma (σ), eğrilik (μ_3), basıklık (μ_4), enerji ve entropi üzerinden genel BDİ özellikleri elde edilir (Sakalli ve Koyuncu, 2022).

$$\mu = \sum_{i=0}^{G-1} i p(i) \quad (3.6)$$

$$\sigma = \sqrt{\sum_{i=0}^{G-1} (i - \mu)^2 p(i)} \quad (3.7)$$

$$\mu_3 = \sigma^{-3} \sum_{i=0}^{G-1} (i - \mu)^3 p(i) \quad (3.8)$$

$$\mu_4 = \sigma^{-4} \sum_{i=0}^{G-1} (i - \mu)^4 p(i) - 3 \quad (3.9)$$

$$Enerji = \sum_{i=0}^{G-1} [p(i)]^2 \quad (3.10)$$

$$Entropi = - \sum_{i=0}^{G-1} p(i) \log_2 [p(i)] \quad (3.11)$$

3.6.2. Sınıflandırma algoritmaları ve karşılaştırma metrikleri

Tez ön çalışmasında; termal görüntülerden elde edilen BDİ özellikleri beş farklı sınıflayıcı yapıya sunularak, en etkin model tespiti araştırılmaktadır. Burada kullanılan sınıflayıcılar; Yapay Sinir ağları (YSA), k-En Yakın Komşuluk (k-NN), Rastgele Orman (RO), Lojistik Regresyon (LR) ve Destek Vektör Makineleri (DVM) olarak belirlenmiştir. Ön çalışma deneyleri ise *Orange 3.32.0* yazılımı kullanılarak sağlanmaktadır. Ön çalışma doğrudan tez konusu kapsamında olmadığı için, burada yalnızca kullanılan metotlara dair işletilen parametreler sunulmaktadır. Yöntemlerin parametre düzenlemeleri Çizelge 3.2’de görüldüğü gibidir (Sakalli ve Koyuncu, 2022).

Çizelge 3.2. Tez ön çalışmasında işletilen sınıflayıcılar ve parametre düzenlemeleri

Sınıflayıcılar	Parametre Düzenlemeleri		
	<i>Aktivasyon</i>	<i>Gizli Düğüm Sayısı</i>	
YSA	Identity	2,4,...,20	
	Logistic		
	Tanh		
	Relu		
k-EYK	Uniform	Euclidean	2,3,...,10
		Manhattan	
	Distance	Chebyshev	
		Mahalanobis	
RO	<i>Ağaç Sayısı</i>		
	10,20,...,100		
LR	<i>Regülerizasyon Tipi</i>		
	Lasso (R1)		
	Ridge (L2)		
DVM	<i>Çekirdek Tipi</i>		
	Polynomial		
	Linear		
	Radial basis function (RBF)		

Tez çalışması kapsamında; analizler kapsamlı sağlandığı ve bilhassa yüksek oranlar kaydedildiği için yalnızca *doğruluk* metriği temelli karşılaştırmalar sunulmaktadır. Ancak tez ön çalışmasında istatistiksel bir model önerildiği ve modele dair performansların yakın olması sebebiyle, dört metrik üzerinden değerlendirmeler sağlanmaktadır. Bu metrikler; sırasıyla Eşitlik (3.12) – Eşitlik (3.15) arası tanımlanan *doğruluk*, *İşlem Karakteristik Eğrisi Altında Kalan Alan (AUC)*, *duyarlılık* ve *özgüllük* olarak sıralanabilir (Fawcett, 2006; Sakalli ve Koyuncu, 2022).

$$\text{Doğruluk} = \frac{DP + DN}{DP + DN + YP + YN} \quad (3.12)$$

$$\text{AUC} = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n 1_{p_i > p_j} \quad (3.13)$$

$$\text{Duyarlılık} = \frac{DP}{DP + YN} \quad (3.14)$$

$$\text{Özgüllük} = \frac{DN}{DN + YP} \quad (3.15)$$

Eşitlik (3.12) – Eşitlik (3.15) kapsamında; Doğru Pozitif (DP) değeri pozitif sınıf olarak atfedilen ve doğru sınıflanan örnek sayısını, Doğru Negatif (DN) değeri negatif sınıfta olan ve doğru atanan örnek sayısını, Yanlış Pozitif (YP) değeri normalde negatif olup pozitif sınıflanan örnek sayısını ve Yanlış Negatif (YN) değeri ise normalde pozitif olup negatif sınıfa atanan örnek sayısını belirtir. AUC eldesi için bir i değişkeni tanımlanır ve değişken “1” etiketli m data noktasını tarar. Yine tanımlı bir diğer değişken olan j ise “0” etiketli n data noktasını tarar. Böylelikle, p_i ve p_j değişkenleri sırasıyla i ve j noktalarına sınıflama algoritması tarafından atanan çıkış, yani olasılık olarak tanımlanabilmektedir. Burada AUC metriği ile sınıflayıcının pozitif işaretli verilere yaptığı atamaların, negatif işaretli verilere verdiği cevaplardan ne derece farklı olduğunu niteler (Fawcett, 2006; Sakalli ve Koyuncu, 2022).

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Tez ön çalışması ve tez çalışmaları kapsamında, termal kamera görüntülerinin sınıflandırılması için üç çalışma gerçekleştirilmektedir. Bu çalışmalarda;

- Tez ön çalışması olarak, istatistiksel bir model ile asenkron motor durumlarının sınıflandırılması,
- Tez çalışması ilk bölümünde, ResNet50 mimarisi ile asenkron motor durumlarının kategorizasyonu,
- Tez çalışması ikinci bölümünde ise beş farklı transfer öğrenme mimarisi ile hem asenkron motor hem de transformatöre ait – birleştirilmiş 20 durumun ayırımı ele alınmaktadır.

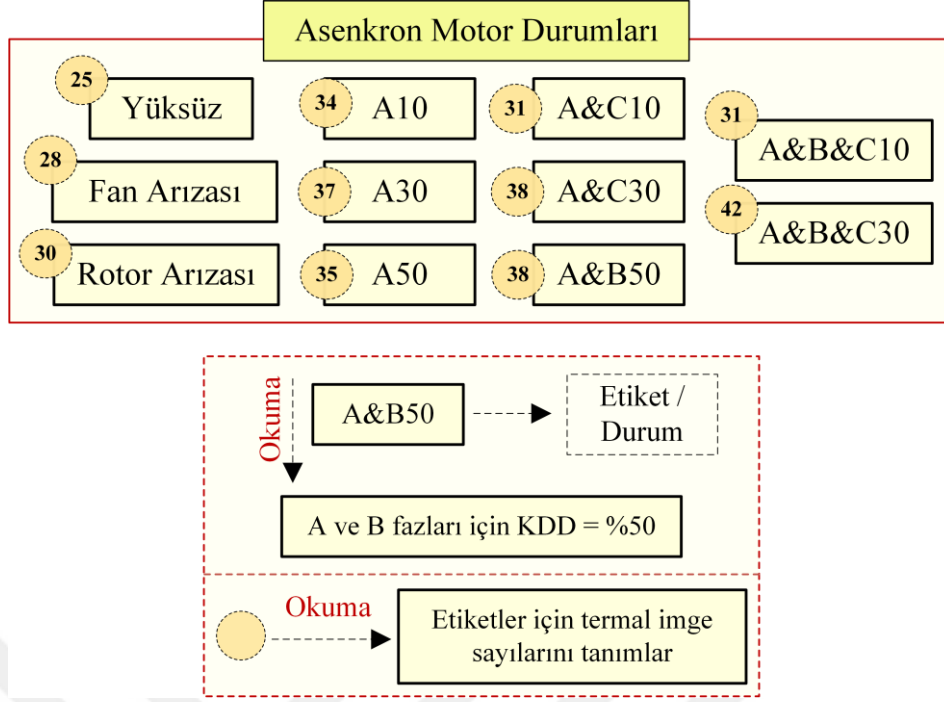
Tez ön çalışması kapsamında *Orange 3.32.0* yazılımı üzerinden değerlendirmeler sağlanmaktadır. Tez çalışmalarında ise *MATLAB Deep Learning Toolbox* işletilerek, sistemde kayıtlı varsayılan tercihler ve detaylıca araştırılan sistem parametreleri üzerinden analizler gerçekleştirilmektedir.

4.1. Tez Ön Çalışması

Tez ön çalışması kapsamında elde edilen BDİ özellikleri, beş farklı sınıflama algoritması üzerinden kategorize edilmiştir. Bu bağlamda değerlendirilen veri seti, Najafi ve ark. (2020)'nin çalışmasında olduğu gibi yalnızca asenkron motor durumlarıdır. Bu durumlar Bölüm 3.6 kapsamında mevcuttur ve Şekil 4.1 kapsamında özetlenmektedir (Sakalli ve Koyuncu, 2022).

Kullanılan veri setinde, 11 farklı senaryoya ait 369 adet termal kamera görüntüsü bulunmaktadır. Bu durumlar; asenkron motorun stator sargılarında 8 farklı kısa devre arızası, rotor arızası, soğutma fan arızası ve yüksüz olarak tanımlanmaktadır. Kısa devre arızaları iki kavrama göre belirlenir: 1-) Faz sayısı, 2-) Faz(lar)da gözlenen kısa devre oranı (Sakalli ve Koyuncu, 2022).

Önerilen model kapsamında; termal görüntüler arasındaki ayırt edici bilgilerin eldesi için BDİ özellikleri kullanılmıştır. Modele dair sınıflandırma aşaması için beş etkin algoritma (YSA, k-EYK, RO, LR, DVM) incelenmiştir. Gerçeklenen deneylerde test yöntemi olarak 10-kat çapraz doğrulama seçilmiş ve performansı değerlendirmek için dört metrik (doğruluk, AUC, özgüllük, duyarlılık) dikkate alınmıştır.



Şekil 4.1. Asenkron motor durumları (Sakalli ve Koyuncu, 2022)

Çizelge 4.2’de YSA temelli deney sonuçları ele alınmıştır. Burada 4 farklı aktivasyon fonksiyonu üzerine farklı düğüm sayılarında karşılaştırmalar yapılarak, en yüksek sonuçların gözlendiği seçenekler sunulmuştur. Buna göre aktivasyon fonksiyonu *Tanh* seçildiği durumda, üç metrik (doğruluk, duyarlılık, özgüllük) üzerine en yüksek performanslar gözlemlenmiştir. İkinci en iyi performans, *ReLU* fonksiyonu işletildiği durumda kaydedilmiştir. Burada diğer fonksiyonlara dair başarı oranları, *ReLU* ile kaydedilen başarılarla eşit veya daha düşük olmaktadır.

Çizelge 4.2. YSA Temelli Deney Sonuçları

Aktivasyon Fonksiyonu	Gizli Düğüm Sayısı	Doğruluk	AUC	Duyarlılık	Özgüllük
Identity	4	0,9485	0,9938	0,9485	0,9950
Logistic	16	0,9431	0,9900	0,9431	0,9947
Tanh	16	0,9539	0,9945	0,9539	0,9958
ReLU	18	0,9485	0,9966	0,9485	0,9951

Çizelge 4.3 kapsamında k-EYK temelli deneyler sunulmaktadır. Burada algoritmaya dair *ağırlık* ve *metrik* parametrelerine dair farklı opsiyonlar incelenmiş, her bir opsiyonda en yüksek performans eldesi için komşuluk sayısı araştırılmıştır. Sonuçlara göre; ağırlık opsiyonu *uniform* seçildiğinde, *distance* opsiyonuna göre daha

düşük doğruluk oranları gözlenmiştir. Ağırlık parametresi *distance* seçildiği durumda ise *Mahalanobis* metriği kullanımı ile üç kıstas (doğruluk, duyarlılık, özgüllük) üzerine en yüksek performanslar kaydedilmiştir. Metrik temelli incelemede ise *Mahalanobis* opsiyonu ile daha yüksek başarıların sağlandığı görülmüştür. *Mahalanobis* opsiyonu sonrasında ikinci en iyi seçeneğin, elde edilen başarı oranlarına göre *Manhattan* olduğu tespit edilmiştir.

Çizelge 4.3. k-EYK Temelli Deney Sonuçları

Ağırlık	Metrik	Komşuluk Sayısı	Doğruluk	AUC	Duyarlılık	Özgüllük
Uniform	Euclidean	5	0,9485	0,9874	0,9485	0,9953
	Manhattan	6	0,9539	0,9916	0,9539	0,9958
	Chebyshev	3	0,9512	0,9840	0,9512	0,9955
	Mahalanobis	2	0,9621	0,9907	0,9621	0,9965
Distance	Euclidean	3	0,9648	0,9843	0,9648	0,9968
	Manhattan	3	0,9675	0,9856	0,9675	0,9970
	Chebyshev	3	0,9675	0,9843	0,9675	0,9971
	Mahalanobis	2	0,9729	0,9909	0,9729	0,9976

Çizelge 4.4 kapsamında LR temelli deney sonuçları sunulmaktadır. Burada regülarizasyon (düzenleyici) tipi *Ridge (L2)* seçildiğinde, bütün metrikler üzerine en iyi performansların elde edildiği görülmüştür.

Çizelge 4.4. LR Temelli Deney Sonuçları

Regülarizasyon Tipi	Doğruluk	AUC	Duyarlılık	Özgüllük
Lasso (R1)	0,8482	0,9837	0,8482	0,9845
Ridge (L2)	0,9295	0,9956	0,9295	0,9931

Çizelge 4.5'te DVM temelli deney sonuçları ele alınmıştır. Burada üç farklı çekirdek (*kernel*) tipi incelenmiş, her bir metrik üzerine en yüksek sonuçların *Polynomial* kerneli ile elde edildiği görülmüştür. Bu opsiyon sonrası en iyi seçenek, diğer opsiyondan daha yüksek başarılar sağladığından *Linear* kerneli olmuştur.

Çizelge 4.5. DVM Temelli Deney Sonuçları

Kernel Tipi	Doğruluk	AUC	Duyarlılık	Özgüllük
Polynomial	0,9702	0,9964	0,9702	0,9971
Linear	0,9350	0,9954	0,9350	0,9941
RBF	0,9295	0,9942	0,9295	0,9936

Çizelge 4.6’da beş sınıflayıcıya dair en yüksek başarılar temel alınarak, genel bir karşılaştırma tablosu sunulmaktadır. Yapılan incelemelere göre k-EYK ve RO algoritmalarının, üç metrik (doğruluk, duyarlılık, özgüllük) üzerine en yüksek değerleri sağladığı görülmüştür. En yüksek AUC değeri ise DVM metodu ile kaydedilmiştir. Burada k-EYK ve RO algoritmalarına dair AUC skorları, en iyi teknik seçimi için belirleyici rol üstlenmiştir. RO algoritması, k-EYK tekniğinden daha yüksek AUC başarısı sağlamış, böylece en optimum sınıflayıcı olarak ön plana çıkmıştır. İkinci en iyi seçenek ise k-EYK metodu olmuştur.

Çizelge 4.6. Algoritmaların optimum sonuçlar üzerine karşılaştırılması

Algoritma	Doğruluk	AUC	Duyarlılık	Özgüllük
YSA	0,9539	0,9945	0,9539	0,9958
k-EYK	0,9729	0,9909	0,9729	0,9976
RO	0,9729	0,9960	0,9729	0,9976
LR	0,9295	0,9956	0,9295	0,9931
DVM	0,9702	0,9964	0,9702	0,9971

Tez ön çalışması neticesinde, BDİ özellik çıkarma yaklaşımına en uygun sınıflayıcının RO metodu olduğu ve önerilecek istatistiksel modelin bu iki yaklaşımı işletmesi gerektiği tespit edilmiştir. Burada önerilen modelin, 11 asenkron motor durumunu %97,29 doğruluk ile kategorize edebildiği gözlemlenmiştir.

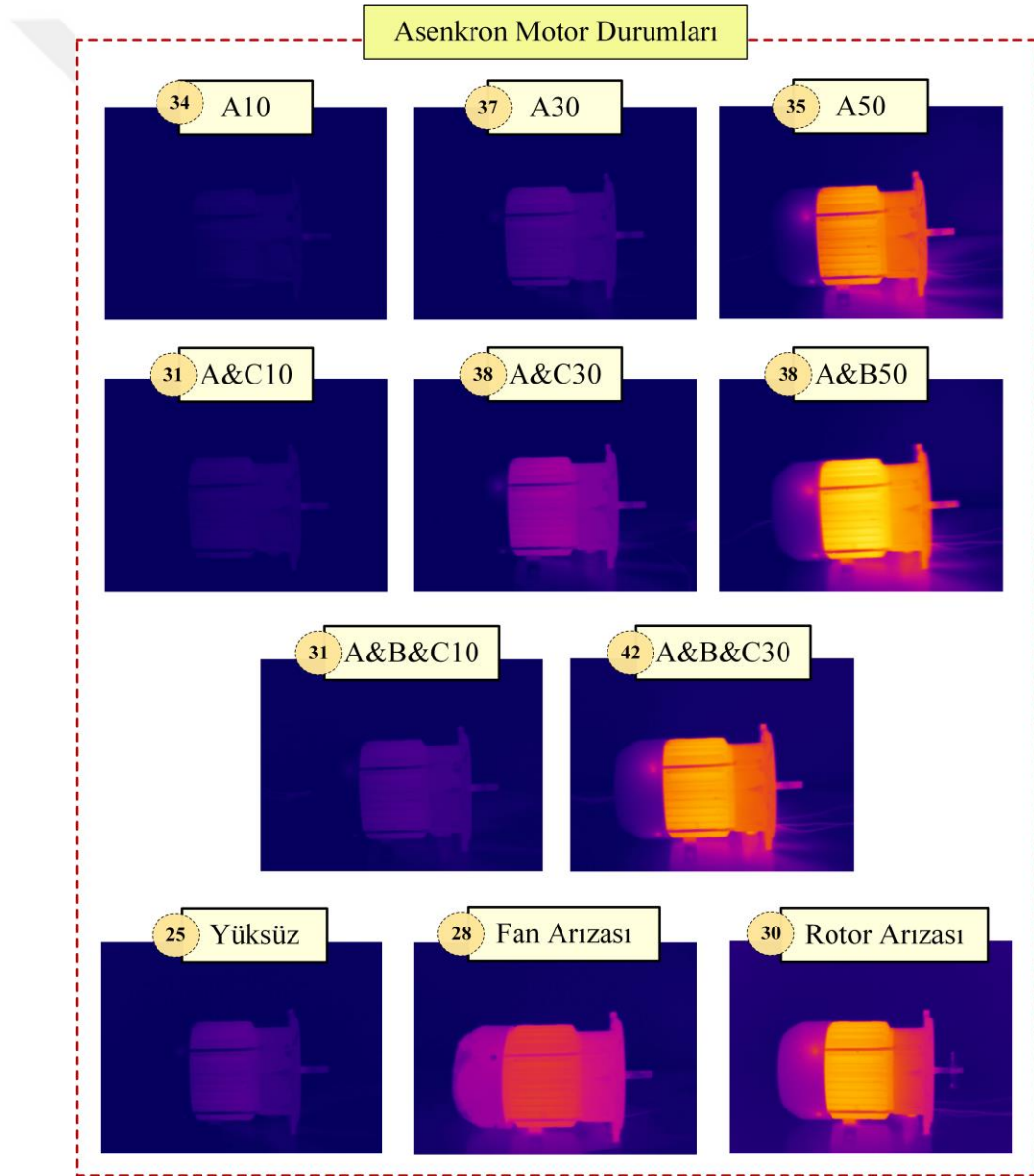
4.2. ResNet50 Mimarisi İle Asenkron Motor Durum Tespiti

Tez çalışması ilk bölümünde, tez ön çalışmasında olduğu gibi asenkron motor durumlarının (11 sınıfın) kategorizasyonu ele alınmıştır. Bu amaçla, literatürde etkin kullanımı nedeniyle ResNet50 mimarisi ile deneyler gerçekleştirilmiştir. Sistem deneyleri, %70-%30 ve %80-%20 eğitim-test ayrımları üzerinden ele alınmıştır. Buradaki temel amaç; eğitim verisinin de bir mimari hiperparametresi olarak görülmesi sebebiyle, mimariye gerekli verinin sunulmasını sağlamaktır. Mimariye dair genel hiperparametre ayarlamaları ise Çizelge 4.7’de görüldüğü gibidir. Bu kapsamda ele alınan hiperparametreler; transfer öğrenme algoritmalarının genel mimarisini etkilemeyen, ancak literatür çalışmalarında sıklıkla ele alınan ve sistem performansında artış sağlayan parametreler ve opsiyonları göz önüne alınarak belirlenmiştir.

Çizelge 4.7. Hiperparametre düzenlemeleri

Hiperparametreler	Değerler / Seçenekler
Epok	100
Mini-yığın boyutu	16, 32
Öğrenme oranı	0,01, 0,001, 0,0001
Öğrenme oranı düşüm faktörü (LRDF)	0,2, 0,4, 0,6, 0,8
Optimize edici / En iyileyici	Adam, Rmsprop, Sgdm

Şekil 4.2’de kullanılan termal görüntü veri seti görsel olarak ifade edilmektedir. Burada asenkron motor durumlarına dair etiket değerlerinin hemen yanında, her bir etikete dair termal görüntü sayıları sunulmaktadır.



Şekil 4.2. Asenkron motor durumları (Sakalli ve Koyuncu, 2023)

ResNet50 mimarisi ile gerçekleştirilen deneyler, %70-%30 ve %80-%20 eğitim-test ayrımları için sırasıyla Çizelge 4.8 ve Çizelge 4.9’da sunulmaktadır.

Çizelge 4.8’e göre; *sgdm* optimize edicinin yüksek öğrenme oranları ile kullanımı, diğer opsiyonlara göre daha yüksek performans sağlamıştır. Öğrenme oranı ‘0,0001’ seçilmesi durumunda; *adam* opsiyonunun, en iyileyciler arasında en optimum seçenek olduğu ve bu seçim ile neredeyse her durumda %100 doğruluk kaydedildiği gözlemlenmiştir. LRDF ve mini-yığın boyutu seçimleri konusunda belirgin bir değişim yoktur ve en yüksek doğruluğu sağlayan değerler bir denemeden diğerine değişebilmektedir. ‘0,0001’ öğrenme oranının, %100 doğruluk tekrarı ile ilgili olarak diğer opsiyonlardan daha etkin olduğu gözlemlenmiştir.

Çizelge 4.8. %70-%30 eğitim-test ayrımı temelli sonuçlar

Mini-yığın Boyutu	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk
16	0,01	0,2	Adam	85,19	0,001	0,2	Adam	99,07	0,0001	0,2	Adam	100
			Sgdm	99,07			Sgdm	99,07			Sgdm	100
			Rmsprop	86,11			Rmsprop	90,74			Rmsprop	99,07
			Adam	86,11			Adam	94,44			Adam	100
		0,4	Sgdm	98,15		0,4	Sgdm	100		0,4	Sgdm	98,15
		Rmsprop	80,56	Rmsprop		83,33	Rmsprop	100				
		0,6	Adam	89,81		0,6	Adam	99,07		0,6	Adam	100
		Sgdm	99,07	Sgdm		100	Sgdm	99,07				
		Rmsprop	86,11	Rmsprop		92,59	Rmsprop	100				
		0,8	Adam	87,04		0,8	Adam	97,22		0,8	Adam	99,70
		Sgdm	99,07	Sgdm		100	Sgdm	99,07				
		Rmsprop	83,33	Rmsprop		78,70	Rmsprop	98,15				
32	0,01	0,2	Adam	88,89	0,001	0,2	Adam	95,37	0,0001	0,2	Adam	100
			Sgdm	100			Sgdm	98,15			Sgdm	98,15
			Rmsprop	88,89			Rmsprop	77,78			Rmsprop	99,07
			Adam	77,78			Adam	100			Adam	100
		0,4	Sgdm	99,07		0,4	Sgdm	96,30		0,4	Sgdm	98,15
		Rmsprop	92,59	Rmsprop		93,52	Rmsprop	100				
		0,6	Adam	88,89		0,6	Adam	94,44		0,6	Adam	99,07
		Sgdm	100	Sgdm		99,07	Sgdm	97,22				
		Rmsprop	93,52	Rmsprop		92,59	Rmsprop	98,15				
		0,8	Adam	75,93		0,8	Adam	98,15		0,8	Adam	100
		Sgdm	100	Sgdm		98,15	Sgdm	98,15				
		Rmsprop	88,89	Rmsprop		78,70	Rmsprop	99,07				

Çizelge 4.9’a göre; *sgdm* optimize edicinin, öğrenme oranı yüksek seçilmesi halinde (Çizelge 4.8’de olduğu gibi) diğer opsiyonlardan daha uygulanabilir olduğu ortaya çıkmıştır. Öğrenme oranı ‘0,0001’ olarak tercih edilirse, tüm optimize edicilerin %100 doğruluk sağlayabildiği farklı seçenekler mevcuttur. Burada ‘0,0001’ öğrenme oranı, alınan en yüksek skor sayıları açısından en optimum seçenek olmuştur. Ayrıca

mini-yığın boyutunun '16' atanması, en yüksek skorların tekrarı açısından uygun bir tercih olmaktadır.

İki test yöntemi karşılaştırıldığında; %80-%20 eğitim-test ayrımı ile gözlemlenen sonuçların, daha fazla eğitim örneği işletilmesi sebebiyle daha yüksek başarı sağladığı görülmüştür. Burada eğitim verisinin, transfer öğrenme mimarisine dair etkin bir hiperparametre olduğu daha açık görülmüştür. ResNet50 mimarisinde; genel olarak düşük öğrenme oranlarıyla daha iyi performans kaydedildiği, *sgdm* en iyileycisinin yüksek öğrenme oranıyla uyumlu olduğu, *adam* ve *rmsprop* optimize edicilerinin ise düşük öğrenme oranıyla daha yüksek skorlar sağladığı tespit edilmiştir.

Özetle; ResNet50 mimarisinin termal görüntülerden 11 asenkron motor durumunu %100 başarı ile ayırt edebildiği, bu nedenle 20 sınıflı (birleştirilmiş) veri seti için farklı mimarilere dair kapsamlı deneylerin anlamlı olacağı tespit edilmiştir.

Çizelge 4.9. %80-%20 eğitim-test ayrımı temelli sonuçlar

Mini-yığın Boyutu	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk
16	0,01	0,2	Adam	97,30	0,001	0,2	Adam	100	0,0001	0,2	Adam	100
			Sgdm	100			Sgdm	100			Sgdm	100
			Rmsprop	83,78			Rmsprop	93,24			Rmsprop	100
			Adam	90,54			Adam	97,30			Adam	100
		0,4	Sgdm	100		0,4	Sgdm	100		0,4	Sgdm	100
			Rmsprop	95,95			Rmsprop	93,24			Rmsprop	91,89
			Adam	94,59			Adam	100			Adam	100
			Sgdm	100			Sgdm	100			Sgdm	100
		0,6	Rmsprop	87,84		0,6	Rmsprop	97,30		0,6	Rmsprop	98,65
			Adam	86,49			Adam	100			Adam	100
			Sgdm	100			Sgdm	100			Sgdm	98,65
			Rmsprop	82,43			Rmsprop	91,89			Rmsprop	100
32	0,01	0,2	Adam	89,19	0,001	0,2	Adam	98,65	0,0001	0,2	Adam	98,65
			Sgdm	100			Sgdm	98,65			Sgdm	98,65
			Rmsprop	85,14			Rmsprop	87,84			Rmsprop	100
			Adam	94,59			Adam	98,65			Adam	100
		0,4	Sgdm	100		0,4	Sgdm	100		0,4	Sgdm	100
			Rmsprop	90,54			Rmsprop	85,14			Rmsprop	100
			Adam	87,84			Adam	98,65			Adam	100
			Sgdm	100			Sgdm	100			Sgdm	100
		0,6	Rmsprop	79,73		0,6	Rmsprop	82,43		0,6	Rmsprop	100
			Adam	90,54			Adam	98,65			Adam	100
			Sgdm	100			Sgdm	98,65			Sgdm	98,65
			Rmsprop	95,95			Rmsprop	95,95			Rmsprop	100

4.3. Transfer Öğrenme Mimarileri İle Kapsamlı Durum Sınıflama

Tez çalışması kapsamında ikinci çalışma, üç fazlı asenkron motor ve tek fazlı transformatör için bütün durumları (20 farklı etiketi) içeren veri seti üzerine, etkin

transfer öğrenme mimarilerinin (DenseNet201, MobileNetV2, ResNet50, ShuffleNet, Xception) karşılaştırılmasıdır. Tez çalışması için genel veri setini oluşturan ekipman durumları, Bölüm 3.6 kapsamında detaylı izah edilmektedir.

Derin öğrenme mimarilerinin hiperparametreleri kapsamlı bir şekilde incelenmiştir. Tez çalışmasının ilk bölümünde önerilen ResNet50 hiperparametre ayarlamaları (Bölüm 4.2'deki Çizelge 4.7), tez çalışmasının ikinci bölümünde de referans olarak kabul edilmiş, böylece beş transfer öğrenme mimarisinin analizleri sağlanmıştır. Gerçeklenen deneylerde, mimarileri veri artırma olmadan ve yeterli eğitim verisi üzerinden karşılaştırmak için %80-20 eğitim-test ayrımı işletilmiştir. Termal görüntü sınıflamasında en uygun mimariyi bulmak için, doğruluk metriği temelli analizlerin yanı sıra derinlemesine bir analiz de gerçekleştirilmiştir.

Tez çalışmasındaki derin öğrenme temelli deneyler; 2.60 GHz CPU, 16GB RAM ve GeForce GTX 1650Ti ekran kartına sahip bir kişisel bilgisayarda, *MATLAB* yazılımında yer alan *Deep Network Designer* araç kutusunda gerçekleştirilmiştir.

Çizelge 4.10 – Çizelge 4.14 arası analizler sırasıyla DenseNet201, MobileNetV2, ResNet50, ShuffleNet ve Xception mimarileri için, hiperparametre değişikliklerine göre sistem performanslarını göstermektedir. Burada her parametre ayarının ve mimarinin önemini yorumlamak için, %100 doğrulukteki tekrar sayısı Toplam İstatistiksel Başarı (TİB) olarak tanımlanmıştır. Ayrıca bir mimarinin; tüm denemelerdeki doğruluk oranlarına dair ortalama alınarak, Ortalama Başarı Oranı (OBO) elde edilmiştir. Derinlemesine analiz kapsamında mimarilere dair TİB, OBO ve hesaplama süresi temelli analizler sunulurken, mimarilerdeki hiperparametre opsiyonları da TİB metriği üzerinden değerlendirilmiştir. Bu sayede, derin öğrenme mimarilerinin etkin bir şekilde karşılaştırılması sağlanmıştır.

4.3.1. DenseNet201 temelli analizler

Çizelge 4.10 incelendiğinde; *sgdm*'nin, genel performansı ve denemelerde kaydettiği TİB sayısı ile DenseNet201 mimarisi için işletilebilecek en uygun optimize edici olduğu görülmüştür. İki mini-yığın boyutu tercihi için TSS değerleri hemen hemen aynıdır ve her iki opsiyon da etkilidir. Öğrenme oranının '0,0001' olarak seçimi, bu tercihte kaydedilen TSS skoru sebebiyle daha uygun görülmektedir. LRDF değerleri üzerine detaylı bir inceleme yapılır ise '0,2' ve '0,4' seçeneklerinin, diğerlerine göre her koşul için en az bir kere % 100 doğruluk sağlayabildiği ve en uygun tercihler oldukları

tespit edilmiştir. Sonuç olarak; DenseNet201 mimarisinin her iki mini-yığın boyutu seçeneğinde iyi performans elde ettiği, düşük öğrenme oranı gerektirdiği ve genel olarak *sgdm* optimize edici ile daha yüksek sonuçlar ürettiği görülmüştür. DenseNet201 temelli değerlendirmelerde TİB ve OBO çıkışları sırasıyla ‘41’ ve %98,45 olarak gözlemlenmiştir.

Çizelge 4.10. DenseNet201 mimarisinin sonuçları

Mini-yığın Boyutu	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk
16	0,01	0,2	Adam	98,37	0,001	0,2	Adam	98,37	0,0001	0,2	Adam	100
			Sgdm	100			Sgdm	100			Sgdm	100
			Rmsprop	97,56			Rmsprop	99,19			Rmsprop	100
			Adam	94,31			Adam	99,19			Adam	100
		0,4	Sgdm	100		0,4	Sgdm	100		0,4	Sgdm	100
			Rmsprop	91,87			Rmsprop	96,75			Rmsprop	100
			Adam	86,99			Adam	100			Adam	100
			Sgdm	100			Sgdm	99,19			Sgdm	100
		0,6	Rmsprop	90,24		0,6	Rmsprop	100		0,6	Rmsprop	100
			Adam	89,43			Adam	100			Adam	99,19
			Sgdm	99,19			Sgdm	100			Sgdm	100
			Rmsprop	95,93			Rmsprop	100			Rmsprop	98,37
32	0,01	0,2	Adam	93,50	0,001	0,2	Adam	99,19	0,0001	0,2	Adam	100
			Sgdm	100			Sgdm	100			Sgdm	100
			Rmsprop	92,68			Rmsprop	100			Rmsprop	99,19
			Adam	98,37			Adam	100			Adam	100
		0,4	Sgdm	100		0,4	Sgdm	100		0,4	Sgdm	100
			Rmsprop	94,31			Rmsprop	100			Rmsprop	99,19
			Adam	97,56			Adam	93,50			Adam	100
			Sgdm	100			Sgdm	99,19			Sgdm	100
		0,6	Rmsprop	100		0,6	Rmsprop	98,37		0,6	Rmsprop	100
			Adam	98,37			Adam	100			Adam	100
			Sgdm	99,19			Sgdm	100			Sgdm	100
			Rmsprop	100			Rmsprop	97,56			Rmsprop	94,31

4.3.2. MobileNetV2 temelli analizler

Çizelge 4.11’deki analizlere göre; *sgdm* en iyileyici, genel performansı ve elde edilen TİB değeri açısından MobileNetV2 ile kullanılabilir en uyumlu optimize edicidir. Mini-yığın boyutu ‘16’ ve öğrenme oranı ‘0,0001’ olarak seçilir ise her seçenekte %100 doğruluk elde edilmiştir. Ayrıca mini-yığın boyutu ‘32’ olduğu durumda öğrenme oranının ‘0,001’ olarak seçimi, TİB temelli sonuçlara göre diğer öğrenme oranı tercihlerinden daha uygundur. LRDF oranları için iki değer (0,2, 0,8) dikkat çekicidir ve bu opsiyonlar için her koşulda en az bir kere %100 doğruluk gözlemlenmiştir. Sonuç olarak; MobileNetV2’nin her iki mini-yığın boyutu tercihlerinde daha iyi performans için düşük bir öğrenme oranına ihtiyaç duyduğu ve *sgdm* optimize

edici ile genel olarak %100 doğruluk sağladığı tespit edilmiştir. MobileNetV2 temelli incelemelerde TİB değeri ve OBO skorunun sırasıyla ‘45’ ve %97,99 olarak elde edildiği görülmüştür.

Çizelge 4.11. MobileNetV2 mimarisinin sonuçları

Mini-yığın Boyutu	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk
16	0,01	0,2	Adam	86,99	0,001	0,2	Adam	99,19	0,0001	0,2	Adam	100
			Sgdm	100			Sgdm	100			Sgdm	100
			Rmsprop	89,43			Rmsprop	100			Rmsprop	100
			Adam	96,75			Adam	100			Adam	100
		0,4	Sgdm	100		0,4	Sgdm	100		0,4	Sgdm	100
			Rmsprop	93,50			Rmsprop	95,93			Rmsprop	100
			Adam	95,12			Adam	95,93			Adam	100
			Sgdm	100			Sgdm	99,19			Sgdm	100
		0,6	Rmsprop	96,75		0,6	Rmsprop	96,75		0,6	Rmsprop	100
			Adam	95,93			Adam	100			Adam	100
			Sgdm	100			Sgdm	100			Sgdm	100
			Rmsprop	89,43			Rmsprop	95,93			Rmsprop	100
32	0,01	0,2	Adam	93,50	0,001	0,2	Adam	100	0,0001	0,2	Adam	100
			Sgdm	100			Sgdm	100			Sgdm	100
			Rmsprop	89,43			Rmsprop	100			Rmsprop	100
			Adam	91,87			Adam	100			Adam	100
		0,4	Sgdm	99,19		0,4	Sgdm	100		0,4	Sgdm	100
			Rmsprop	89,43			Rmsprop	100			Rmsprop	100
			Adam	93,50			Adam	100			Adam	100
			Sgdm	100			Sgdm	100			Sgdm	99,19
		0,6	Rmsprop	91,87		0,6	Rmsprop	98,37		0,6	Rmsprop	99,19
			Adam	96,75			Adam	100			Adam	100
			Sgdm	100			Sgdm	100			Sgdm	96,75
			Rmsprop	89,43			Rmsprop	100			Rmsprop	100

4.3.3. ResNet50 temelli analizler

Çizelge 4.12’de görüldüğü gibi; *sgdm*, genel performansı ve deneylerde elde edilen TİB sayısı nedeniyle kullanımı daha uygun optimize edicidir. Mini-yığın boyutu ‘16’ olarak tercih edilir ise elde edilen TİB değerinin, ‘32’ opsiyonu ile kaydedilen TİB değerinden daha fazla olduğu görülmektedir. Öğrenme oranlarının değerlendirilmesinde elde edilen TİB değerine göre ‘0,0001’ seçimi daha güvenilir görünmektedir. Üç LRDF seçeneği (0,2, 0,6, 0,8) ile her koşulda en az bir kere %100 doğruluk elde edilmiştir. Ayrıca ‘0,0001’ öğrenme oranı ve ‘16’ mini-yığın boyutu seçildiğinde, LRDF değeri 0,2 ve 0,6 olarak sabitlenirse tüm optimize ediciler için %100 doğruluk gözlemlenmektedir. Sonuç olarak; ResNet50’nin düşük mini-yığın boyutu ile çalışma eğiliminde olduğu, çoğunlukla düşük öğrenme oranıyla başarılı sağladığı ve genel olarak *sgdm* en iyileyici kullanımı ile yüksek performans elde ettiği görülmüştür.

ResNet50 temelli deneylerde TİB değeri ve OBO skoru için sırasıyla ‘29’ değeri ve %97,43 oranı kaydedilmiştir.

Çizelge 4.12. ResNet50 mimarisinin sonuçları

Mini-yığın Boyutu	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk
16	0,01	0,2	Adam	98,37	0,001	0,2	Adam	95,93	0,0001	0,2	Adam	100
			Sgdm	100			Sgdm	100			Sgdm	100
			Rmsprop	98,37			Rmsprop	95,93			Rmsprop	100
		0,4	Adam	88,62		Adam	92,68	Adam		98,37		
			Sgdm	99,19		Sgdm	100	Sgdm		98,37		
			Rmsprop	92,68		Rmsprop	96,75	Rmsprop		100		
		0,6	Adam	77,24		Adam	96,75	Adam		100		
			Sgdm	100		Sgdm	100	Sgdm		100		
			Rmsprop	91,87		Rmsprop	87,80	Rmsprop		100		
		0,8	Adam	98,37		Adam	97,56	Adam		100		
			Sgdm	100		Sgdm	100	Sgdm		99,19		
			Rmsprop	91,87		Rmsprop	87,80	Rmsprop		97,56		
32	0,01	0,2	Adam	90,24	0,001	0,2	Adam	100	0,0001	0,2	Adam	100
			Sgdm	100			Sgdm	99,19			Sgdm	99,19
			Rmsprop	99,19			Rmsprop	95,12			Rmsprop	95,93
		0,4	Adam	95,93		Adam	97,56	Adam		100		
			Sgdm	100		Sgdm	100	Sgdm		97,56		
			Rmsprop	98,37		Rmsprop	95,12	Rmsprop		100		
		0,6	Adam	96,75		Adam	95,93	Adam		100		
			Sgdm	100		Sgdm	100	Sgdm		99,19		
			Rmsprop	95,93		Rmsprop	99,19	Rmsprop		96,75		
		0,8	Adam	96,75		Adam	100	Adam		100		
			Sgdm	100		Sgdm	100	Sgdm		98,37		
			Rmsprop	98,37		Rmsprop	93,50	Rmsprop		99,19		

4.3.4. ShuffleNet temelli analizler

Çizelge 4.13’te görüldüğü üzere; denemelerdeki genel performans ve TİB değerine göre en iyi optimize edicinin *sgdm* olduğu görülmüştür. Mini-yığın boyutuna dayalı değerlendirmelerde ise opsiyonlar için neredeyse aynı TİB değerlerinin elde edildiği ve bu nedenle her iki tercihin de kullanılabilir olduğu gözlenmiştir. ‘0,0001’ öğrenme oranı ile mini-yığın boyutu ‘16’ seçilir ise deneylerde daha iyi TİB değeri elde edilmiştir. İki LRDF oranı (0,2, 0,4) için her koşulda en az bir kere %100 doğruluk kaydedilmiştir. Mini-yığın boyutu ‘16’ ve ‘32’ olarak seçilir ise ‘0,001’ öğrenme oranı ile kullanmak için sırasıyla ‘0,2’ ve ‘0,6’ LRDF oranları ön plana çıkmaktadır. Mini-yığın boyutu ‘16’ ve öğrenme oranı ‘0,0001’ olduğu durumda, üç LRDF tercihi (0,2, 0,6, 0,8) için optimize edici seçimi gerekli değildir. Sonuç olarak; ShuffleNet’in genelde düşük öğrenme oranları ve *sgdm* optimize edici kullanımı ile en yüksek sonuçları

sağladığı görülmüştür. ShuffleNet mimarisinde TİB ve OBO metrikleri için sırasıyla ‘41’ değeri ve %98,92 skoru gözlemlenmiştir.

Çizelge 14 ShuffleNet mimarisinin sonuçları

Mini-yığın Boyutu	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk
16	0,01	0,2	Adam	99,19	0,001	0,2	Adam	100	0,0001	0,2	Adam	100
			Sgdm	100			Sgdm	100			Sgdm	100
			Rmsprop	95,12			Rmsprop	100			Rmsprop	100
		0,4	Adam	96,75		Adam	96,75	Adam		100		
			Sgdm	100		Sgdm	100	Sgdm		99,19		
			Rmsprop	98,37		Rmsprop	100	Rmsprop		98,37		
		0,6	Adam	92,68		Adam	98,37	Adam		100		
			Sgdm	97,56		Sgdm	100	Sgdm		100		
			Rmsprop	98,37		Rmsprop	100	Rmsprop		100		
		0,8	Adam	96,75		Adam	95,93	Adam		100		
			Sgdm	99,19		Sgdm	100	Sgdm		100		
			Rmsprop	99,19		Rmsprop	100	Rmsprop		100		
32	0,01	0,2	Adam	95,12	0,001	0,2	Adam	100	0,0001	0,2	Adam	99,19
			Sgdm	100			Sgdm	99,19			Sgdm	100
			Rmsprop	100			Rmsprop	95,93			Rmsprop	100
		0,4	Adam	100		Adam	100	Adam		100		
			Sgdm	100		Sgdm	100	Sgdm		100		
			Rmsprop	95,93		Rmsprop	95,12	Rmsprop		100		
		0,6	Adam	96,75		Adam	100	Adam		99,19		
			Sgdm	100		Sgdm	100	Sgdm		100		
			Rmsprop	97,56		Rmsprop	100	Rmsprop		100		
		0,8	Adam	94,31		Adam	100	Adam		98,37		
			Sgdm	100		Sgdm	99,19	Sgdm		98,37		
			Rmsprop	97,56		Rmsprop	99,19	Rmsprop		99,19		

4.3.5. Xception temelli analizler

Çizelge 4.14 incelemeleri kapsamında; *sgdm* optimize edicinin, TİB başarısına göre diğer opsiyonlardan daha uygun olduğu görülmüştür. Mini-yığın boyutu değerlendirmesinde; bütün denemelerde %100 doğruluk sağlandığından, ‘16’ mini-yığın boyutu ve ‘0,0001’ öğrenme oranı kullanımı dikkat çekicidir. Mini-yığın boyut ‘32’ seçilir ise düşük öğrenme oranlarında verimli sonuçlar alınabilmektedir. Bütün LRDF oranları için en az bir kere %100 başarı sağlanmıştır. Ancak, üç LRDF oranı (0,2, 0,6, 0,8) için TİB temelli daha iyi performans elde edilmiş ve bu değer ‘12’ olarak gözlenmiştir. Sonuç olarak; Xception’ın genelde düşük öğrenme oranı ve *sgdm* en iyileyci çalıştırarak en yüksek performanslara ulaştığı ortaya çıkmıştır. Xception mimarisinin, TİB ve OBO metrikleri üzerine sırasıyla ‘45’ değeri ve %98,66 skorunu sağladığı tespit edilmiştir.

Çizelge 4.14. Xception mimarisinin sonuçları

Mini-yığıın Boyutu	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk	Öğrenme Oranı	LRDF	Optimize Edici	Doğruluk
16	0,01	0,2	Adam	98,37	0,001	0,2	Adam	100	0,0001	0,2	Adam	100
			Sgdm	100			Sgdm	100			Sgdm	100
			Rmsprop	97,56			Rmsprop	100			Rmsprop	100
32	0,01	0,2	Adam	97,56	0,001	0,2	Adam	100	0,0001	0,2	Adam	100
			Sgdm	100			Sgdm	99,19			Sgdm	99,19
			Rmsprop	91,87			Rmsprop	100			Rmsprop	100
32	0,01	0,4	Adam	98,37	0,001	0,4	Adam	94,31	0,0001	0,4	Adam	99,19
			Sgdm	100			Sgdm	100			Sgdm	100
			Rmsprop	95,93			Rmsprop	99,19			Rmsprop	100
32	0,01	0,6	Adam	100	0,001	0,6	Adam	100	0,0001	0,6	Adam	100
			Sgdm	100			Sgdm	100			Sgdm	100
			Rmsprop	90,24			Rmsprop	100			Rmsprop	97,56
32	0,01	0,8	Adam	97,56	0,001	0,8	Adam	100	0,0001	0,8	Adam	100
			Sgdm	100			Sgdm	100			Sgdm	96,75
			Rmsprop	89,43			Rmsprop	100			Rmsprop	100

4.4. Detaylı Analiz ve Tartışma

Bölüm 4.1’de belirtilen analizlere göre; önerilen model için BDİ özellikleri ve RO sınıflayıcısı kullanılmış, 11 asenkron motor durumunun %97,29 sınıflama doğruluğu ile kategorize edildiği görülmüştür. Burada derin öğrenme mimarilerinin kullanım fikri ortaya çıkmış, Bölüm 4.2’de ResNet50 mimarisi ile motor durumları sınıflandırılmıştır. Sonuç olarak, iki farklı veri seti düzenlemesi üzerine %100 doğruluk elde edilmiştir. Böylece, tez çalışmasının temelini teşkil eden ikinci analizin (birleştirilmiş 20 durum ayrımının) değerlendirilmesi sağlanmıştır. Bu amaçla, beş etkin transfer öğrenme mimarisi ile gerçekleştirilen deneyler Bölüm 4.3’de kapsamlı analiz edilmiştir.

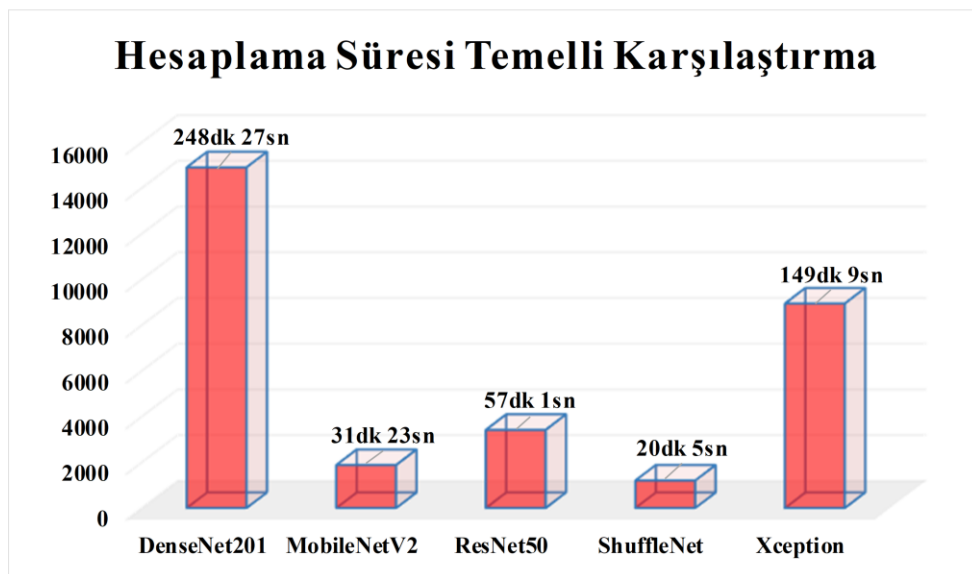
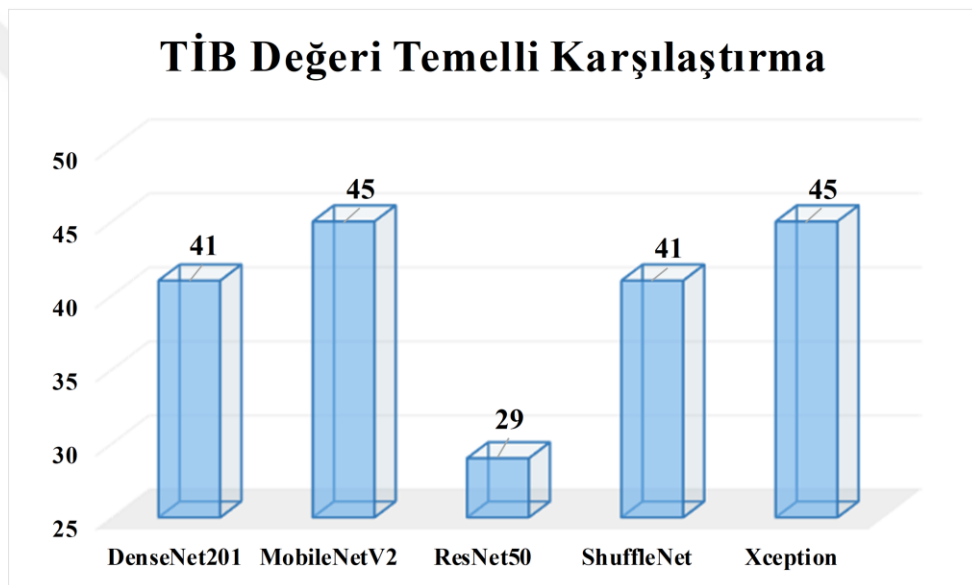
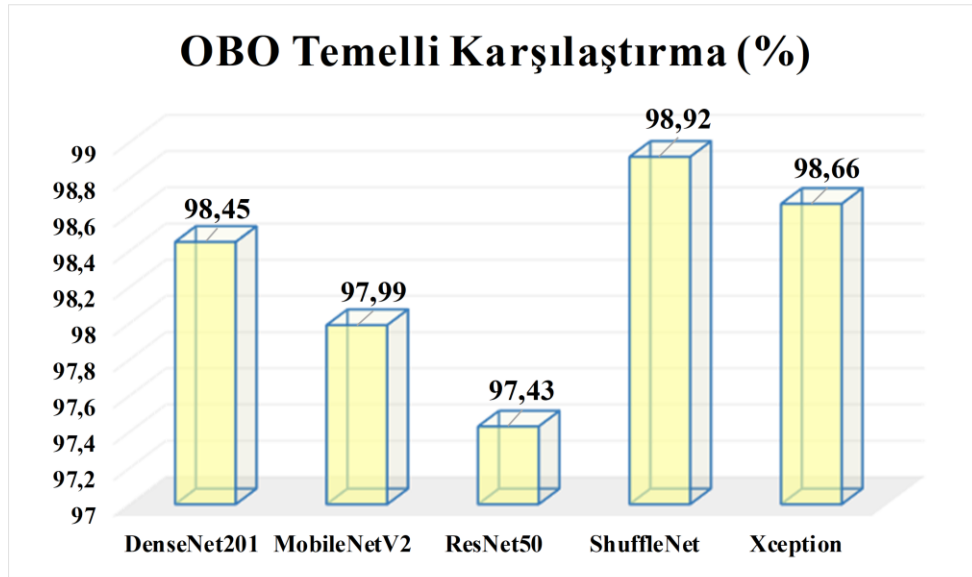
Bölüm 4.4’de, (temel çalışmayı teşkil ettiğinden) tez çalışmasının ikinci kısmı üzerine irdemeler sunulmaktadır. Hâlihazırda diğer bölümlerin (Bölüm 4.1 ve Bölüm 4.2) irdemeleri, bu bölümlerin sonlarında değerlendirilmiştir. Bu bölümde mimarilere dair OBO, TİB ve hesaplama süresi temelli analizler ile mimarilerin hiperparametre ayarlamaları ve literatür karşılaştırması ele alınmıştır.

Eđitim verileri, transfer öğrenme mimarileri için bağımsız bir hiperparametre olarak görülmektedir. Ayrıca derin öğrenme yöntemlerini ele alan literatür çalışmalarına göre veri artırma ve %90-%10 eğitim-test ayrımı sıklıkla kullanılmaktadır. Tez çalışmasında veri artırma tercih edilmemiş, deneyler kapsamında test yöntemi olarak %80-%20 eğitim-test ayrımı değerlendirilmiştir. Diğer bir deyişle buradaki amaç, transfer öğrenme algoritmalarını veri artırma olmadan uygulamak ve kendi başına bir hiperparametre olarak görüldüğü için yeterli eğitim verisinden yararlanmaktır.

Tez çalışmasının ikinci bölüm analizlerinde görüldüğü üzere; her bir derin öğrenme mimarisi, asenkron motor ve transformatörün durumlarına dair yirmi farklı opsiyon için %100 doğruluk sağlamaktadır. Bu durum ise derinlemesine bir analizin gerekliliğini ortaya çıkarmıştır.

Şekil 4.3'te beş mimarinin performanslarına dair derinlemesine bir analiz yapmak için OBO, TİB ve hesaplama süresine dayalı karşılaştırmalar sunulmaktadır. Burada TİB değerleri ve OBO skorları, mimarilerin sınıflandırma performansını açıklamaktadır. Bu metriklerin incelenmesinde; OBO metriđi, 72 deneme üzerine ortalama başarıyı gösteren metrik olup, mimarilerin başarısını ölçmede TİB metriđinden daha fazla önem arz etmektedir. Çünkü ortalama performans, sistemlerin termal görüntülerde durum sınıflama için hangi derece kararlı olduğunu ortaya koymaktadır. Bu sayede mimarilerin performans analizleri iki farklı bakış açısı üzerinden değerlendirilmektedir. Ayrıca, mühendislik alanında bir sistem daha kısa sürede iyi sonuçlar veriyor ise dikkat çekici olarak tanımlanabilir. Bu aşamada hesaplama zamanı temelli değerlendirmeler bu konuya ışık tutmaktadır. Bölüm 4.3'teki kapsamlı analizlere ve Şekil 4.4'deki sonuçlara göre;

- Transfer öğrenme mimarileri; çok sınıflı termal görüntü veri setinin sınıflandırılmasında genel olarak düşük bir öğrenme oranı, *sgdm* optimize edici ve küçük mini-yığın boyutu kullanarak daha yüksek performans sergilemektedir.
- Tüm mimariler, ayrıntılı olarak analiz edilen hiperparametre düzenlemeleri ve eğitim setinin de bir hiperparametre olarak görüldüğü %80-%20 eğitim-test ayrımı ile %100 doğruluk elde etmektedir.



Şekil 4.4. OBO, TİB ve hesaplama süresi temelli derinlemesine analiz

- OBO temelli değerlendirmeler sonucunda ShuffleNet mimarisinin, deneylerde kararlı bir performans sağlayarak diğer mimarilerden daha yüksek başarı elde ettiği görülmektedir. Xception ise ikinci en iyi mimari olarak karşımıza çıkmaktadır.
- TİB temelli değerlendirmeler sonucunda MobileNetV2 ve Xception mimarileri, 72 denemenin 45'inde %100 doğruluk elde ederek ön plana çıkmaktadır. ShuffleNet ve DenseNet201 mimarileri ise ikinci en yüksek başarıyı sağlamaktadır.
- Birleştirilmiş eğitim ve test süreleri temelli analizlerde ShuffleNet en hızlı mimari olarak ortaya çıkarken, MobileNetV2 ile ikinci en iyi zamanlama elde edilmektedir.

Bahsi geçen çıkarımlar ve irdelemeler ışığında; ShuffleNet mimarisinin, elde edilen OBO skoruna ilişkin en kararlı mimari olduğu ve 20dk 5sn'de sonuç vererek en hızlı sistem olarak tanımlandığı görülmektedir. Ayrıca ShuffleNet mimarisi, ikinci en iyi TİB değerine sahip olmaktadır ve bu TİB değeri (41) en iyi TİB değerine (45) çok yakındır. Sonuç olarak ShuffleNet mimarisi; termal görüntüde, asenkron motor ve transformatör durumlarının sınıflandırılması için en uygun transfer öğrenme yöntemi olarak ön plana çıkmaktadır.

Tez kapsamında gerçekleştirilen çalışmalar ile literatürdeki etkin ve son zamanlarda ele alınan çalışmalar Çizelge 4.15'de karşılaştırılmaktadır. Burada önerilen tez çalışması; birçok çoklu sınıf durumu içeren birden fazla elektrikli ekipman için, termal görüntüler üzerinden sınıflamanın yüksek performans ile sağlanabileceğini göstermektedir. Böylece transfer öğrenme mimarilerinin; KDD sınıflamasını iki ekipmana göre ayırabildiği, bu KDD arızalarını rotor ve fan arızalarından ayırt edilebildiği, ayrıca iki yüksüz durumu birbirine karıştırmadan bütün durumları %100 başarı ile kategorize edilebildiği kanıtlanmıştır. Ayrıca termal görüntü analizlerinin, farklı ekipmanlara dair durumları belirlemede etkili bir araç olarak kullanılabileceği görülmüştür. Termal görüntüleme ile birden fazla ekipmana dair çoklu sınıf durum kategorizasyonu, görüntüleme ile uyumlu olan ekipmanların analizinde dikkate değer ilerleme sağlayabilecek ve literatürde halen araştırılması gereken ümit vadeden bir konudur.

Çizelge 4.15. Literatür karşılaştırması

Çalışma	Mimari / Model	Temel	Ekipman	Durumlar	Test Yöntemi	Doğruluk (%)
Lozanov ve ark. (2020)	<u>Segmentasyon:</u> Otsu <u>Özellik çıkarma:</u> Histogram, GDEM ve bileşen temelli özellikler <u>Sınıflama:</u> DVM	Termal görüntü analizi	Motor	Soğutma fan arızası, motor yatağı arızası, arızasız (3 sınıf)	62.5%-37.5% eğitim-test ayrımı	83,3
Najafi ve ark. (2020)	<u>Segmentasyon:</u> RO ve Adaboost <u>Özellik çıkarma:</u> Histogram temelli özellikler <u>Sınıflama:</u> KA	Termal görüntü analizi	Motor	Yüksüz, farklı fazlar için sekiz KDD, rotor arızası, soğutma fan arızası (11 sınıf)	50%-50% eğitim-test ayrımı (10-kat çapraz doğrulama ile eğitim)	93,8
Choudhary ve ark. (2020)	<u>Gürültü giderme:</u> 2B-ADD <u>Özellik çıkarma:</u> TBA <u>Özellik seçme:</u> MM <u>Sınıflama:</u> DVM	Termal görüntü analizi	Motor	Normal, iç halka arızası, dış halka arızası, yağlama arızası (4 sınıf)	5-kat çapraz doğrulama	100
Granados-Lieberman ve ark. (2021)	<u>Özellik çıkarma:</u> HFÖÜ <u>Sınıflama:</u> BM	Harmonik analizi	Trafo	Normal, beş farklı yüzdelerde KDD (6 sınıf)	Belirtilmemiş	100
Fang ve ark. (2021)	<u>Özellik çıkarma:</u> Bölgesel sıcaklık, GDEM ve Hu moment temelli özellikler <u>Sınıflama:</u> ÇÜA	Termal görüntü analizi	Trafo	Ekipman hatası ve aşırı ısınma (2 sınıf)	Belirtilmemiş	84,2
Jorkesh ve Poshtan (2021)	<u>Gürültü giderme:</u> BBA <u>Özellik çıkarma:</u> BDİ <u>Sınıflama:</u> YSA	3-fazda gerilim ve akım sinyalleri	Motor	Arızasız, %8 ve %12,5 KDD, yatak arızası, kırık rotor çubuğu, pompanın kuru çalışması, dengesiz güç kaynağı (7 sınıf)	60%-20%-20% eğitim-validasyon- test ayrımı	98
Yaman (2021)	<u>Özellik çıkarma:</u> ADD & YİÖ <u>Özellik seçme:</u> KBA <u>Sınıflama:</u> k-EYK	Ses sinyalleri	Motor	Bir kırık rotor çubuğu, iki kırık rotor çubuğu, üç kırık rotor çubuğu, yatak arızaları, arızasız (5 sınıf)	10-kat çapraz doğrulama	99,93
Saucedo-Dorantes ve ark. (2021)	<u>Özellik çıkarma:</u> Zaman & frekans domeni özellikleri <u>Özellik seçme:</u> GA-TBA & LDA <u>Sınıflama:</u> YSA	Stator akım ve titreşim sinyalleri	Motor	Arızasız, kırık rotor çubuğu, dengesizlik, hiza kaçıklığı, dengesizlik + hiza kaçıklığı, kırık rotor çubuğu + dengesizlik, kırık rotor çubuğu + hiza kaçıklığı, kırık rotor çubuğu + dengesizlik + hiza kaçıklığı (8 sınıf)	5-kat çapraz doğrulama	98,25
Khanjani ve Ezoji (2021)	<u>Özellik çıkarma:</u> AlexNet <u>Sınıflama (1):</u> KOA <u>Sınıflama (2):</u> DVM	Termal görüntü analizi	Motor	Arızasız, fan arızası, bir fazda %10 KDD, bir fazda %30 KDD, iki fazda %10 KDD, üç fazda %10 KDD (6 sınıf) Arızasız, rotor blokesi, fan arızası, bir fazda %10 KDD, bir fazda %30 KDD, bir fazda %50 KDD, iki fazda %10 KDD, iki fazda %30 KDD, iki fazda %50 KDD, üç fazda %10 KDD, üç fazda %30 KDD (11 sınıf)	70%-30% eğitim-test ayrımı	100
Yi ve ark. (2022)	<u>Gürültü giderme:</u> SR <u>Özellik seçme:</u> KAO <u>Sınıflama:</u> KAO-ÜÇDVVT	Akım sinyalleri	Motor	Arızalı, arızasız (2 sınıf)	50%-50% eğitim-test ayrımı	98
Martinez-Herrera ve ark. (2022)	<u>Özellik çıkarma:</u> Homojenlik ve Basıklık <u>Sınıflama:</u> YSA	Başlangıç geçici akım sinyalleri	Motor	Arızasız, bir kırık rotor çubuğu, iki kırık rotor çubuğu, rulman dış bilezik hasarı, dengesiz mekanik yük (5 sınıf)	70%-30% eğitim-test ayrımı	100

Ibrahim ve ark. (2022(a))	<u>Özellik çıkarma:</u> GDEM <u>Özellik seçme:</u> YOA <u>Sınıflama:</u> XGBoost	Termal görüntü analizi	Motor	Normal, iç yatak arızası, dış yatak arızası, rulman arızası, bir kırık rotor çubuğu, beş kırık rotor çubuğu, sekiz kırık rotor çubuğu (7 sınıf)	10-kat çapraz doğrulama	93
Ibrahim ve ark. (2022(b))	<u>Özellik çıkarma:</u> EfficientNetB0 <u>Sınıflama:</u> RO	Termal görüntü analizi	Motor	Normal, iç yatak arızası, dış yatak arızası, rulman arızası, bir kırık rotor çubuğu, beş kırık rotor çubuğu, iç yatak arızası + bir kırık rotor çubuğu, dış yatak arızası + beş kırık rotor çubuğu, bilyeli yatak arızası + sekiz kırık rotor çubuğu (10 sınıf)	10-kat çapraz doğrulama	97
<u>Tez Ön Çalışması:</u> Sakallı ve Koyuncu (2022(a))	<u>Özellik çıkarma:</u> BDI <u>Sınıflama:</u> RO	Termal görüntü analizi	Motor	Yüksüz, farklı fazlar için sekiz KDD, rotor arızası, soğutma fan arızası (11 sınıf)	10-kat çapraz doğrulama	97,29
<u>Tez Çalışması:</u> Sakallı ve Koyuncu (2022(b))	<u>Özellik çıkarma & Sınıflama:</u> ResNet50	Termal görüntü analizi	Motor	Yüksüz, farklı fazlar için sekiz KDD, rotor arızası, soğutma fan arızası (11 sınıf)	80%-20% eğitim-test ayrımı	100
<u>Tez Çalışması:</u> Sakallı ve Koyuncu (2023(a))	<u>Özellik çıkarma & Sınıflama:</u> DenseNet201 <u>Özellik çıkarma & Sınıflama:</u> MobileNetV2 <u>Özellik çıkarma & Sınıflama:</u> ResNet50 <u>Özellik çıkarma & Sınıflama:</u> ShuffleNet <u>Özellik çıkarma & Sınıflama:</u> Xception	Termal görüntü analizi	Trafo ve Motor	<u>Trafo:</u> Yüksüz, sekiz farklı oranda KDD <u>Motor:</u> Yüksüz, farklı fazlar için sekiz KDD, rotor arızası, soğutma fan arızası (20 sınıf)	80%-20% eğitim-test ayrımı	100

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Tez çalışması kapsamında; birçok duruma sahip iki elektrikli ekipmanı içeren birleşik veri seti dikkate alınarak, kapsamlı bir araştırma ve analiz sunulmaktadır. Etkin beş derin öğrenme mimarisi, ekipman durumlarını ayırt etmek için değerlendirilmektedir. Burada termal görüntüleri doğru bir şekilde kategorize etmek için, transfer öğrenme mimarilerine dair derinlemesine analizler gerçekleştirilmektedir. Bu amaçla, hiperparametre düzenlemeleri kapsamlı bir şekilde incelenmekte ve sistem üzerindeki etkileri detaylı bir şekilde yorumlanmaktadır.

Gerçeklenen deneylerde; bütün transfer öğrenme mimarilerinin, asenkron motor ve trafo ekipmanlarına ait yirmi sınıfı %100 doğrulukla kategorize edebildiği ortaya çıkmıştır. Ayrıca bir mimarinin hiperparametre seçimi de mimariden mimariye farklılık gösterebilmektedir. Bununla birlikte mimariler; çok sınıflı termal veri setinin kategorizasyonu için genel olarak düşük bir öğrenme oranı, *sgdm* optimize edici ve küçük bir mini-yığın boyutu kullanmakta ve yüksek başarılar sağlamaktadır. ShuffleNet mimarisi; derinlemesine analiz ve değerlendirmelerde gözlenen OBO, TİB ve hesaplama süresi sonuçlarına göre, diğer mimarilerden daha umut verici bir performans sağlamakta ve ön plana çıkmaktadır.

Çok sınıflı veri analizinde termal görüntülere dair ısı farklılıklarının, transfer öğrenme mimarileri için gerekli bilgileri içerdiği görülmektedir. Termal görüntülemenin ölçüm temeli; diğer ölçüm yaklaşımlarına göre daha uyumlu, daha verimli ve daha ergonomik (herhangi bir müdahale olmaksızın / uzaktan) olduğu gözlenmektedir. Burada ikiden fazla ekipmana dair durum ayrımının, termal görüntüler üzerinden incelenebilecek kapsamlı bir araştırma teşkil ettiği ve bu konunun literatüre yön verebileceği öngörülmektedir.

Çalışma neticesinde, termal görüntülerde sınıflama analizi için ShuffleNet veya ShuffleNet temelli mimarilerin kullanımları tavsiye edilir. Ayrıca derin öğrenme modellerinin; temel bir sınıflayıcı modele rahatlıkla üstünlük sağlayabildiği, termal temelli bir sınıflamada derin öğrenme yapılarına dair kullanımın çok daha uygun olacağı öngörülmektedir.

5.2 Öneriler

Elde edilen bulgular ışığında üç önemli çıkarım bulunmaktadır;

- Termal görüntüleme; ekipmanın görünümünde bir ısı değişikliği varsa, birden çok elektrikli ekipmanın çok sınıflı durumlarını ayırt etmek için kullanılacak etkili bir araçtır. Burada termal görüntülemenin, ekipman durumlarının ısı değişimine sebebiyet vermesi halinde ikiden fazla ekipman için kullanılacağı öngörülmektedir. Bu konu ise halen araştırmaya ve incelemeye açık bir durumdadır.
- Transfer öğrenme mimarileri; çoklu sınıf etiketi içeren ve farklı işlevlere sahip ekipmanlar üzerine, termal görüntüleri yüksek doğrulukla sınıflandırabilmektedir. Ayrıca tez çalışması ikinci bölümünde gözlemlendiği gibi bu mimariler; ısı değişimleri birbirine benzeyen KDD'leri ekipmana göre ayırabilmekte, bu ısı değişimlere yakın görünümdeki rotor ve fan arızalarını sınıflayabilmekte ve iki farklı yüksüz durumu da kategorize edebilmektedir. Burada önemli bir diğer husus da eğitim veri setinin bir hiperparametre olarak görülmesi gerekliliğidir. Transfer öğrenme mimarilerinin yeterli eğitim verisi ile yüksek performans sağladığı bilinmektedir. Bu husus, tez çalışmasının ilk bölümü olan ResNet50 temelli iki test senaryosu üzerinde ispat edilmiştir. Derin öğrenme mimarileri ile literatürde önerilen modellerden daha yüksek başarıların eldesi, tez ön çalışması ve tez çalışması ilk bölümünde gözlemlenmiştir. Bu durum ayrıca literatür karşılaştırmasında (Çizelge 4.15) da sunulmaktadır. Kısaca, transfer öğrenme mimarilerinin termal görüntüleme ile ekipman durum tespiti için kullanılacak uygun sistemler olduğu önerilmektedir.
- ShuffleNet, termal görüntülerde elektrikli ekipmanların durumlarını belirlemek için en kararlı ve başarılı mimari olarak ön plana çıkmaktadır. Burada ShuffleNet, ShuffleNet temelli bir türev, ShuffleNet içerebilecek bir model veya ShuffleNet'ten ilham alarak üretilebilecek bir mimari ile termal görüntülerde ekipman durumu tespiti için yüksek başarıların gözleneceği öngörülmektedir.

KAYNAKLAR

- Apaydin, H., Feizi, H., Sattari, M. T., Colak, M. S., Shamsirband, S. and Chau, K. W., 2020, Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting, *Water*, 12 (5), 1500.
- Benbouzid, M., 1999, Bibliography on induction motors faults detection and diagnosis, *IEEE Transactions on Energy Conversion*, 14 (4), 1065-1074.
- Bhandare, A. and Kaur, D., 2021, Designing convolutional neural network architecture using genetic algorithms, *International Journal of Advanced Network, Monitoring and Controls*, 6 (3), 26-35.
- Cao, Q., Che, Y., Yang, J., Mi, M. and Men, Y., 2021, Short-circuit and open-circuit faults monitoring of IGBTs in solid-state-transformers using collector-emitter voltage, *Journal of Power Electronics*, 21 (7), 1052-1060.
- Chollet, F., 2017, Xception: Deep learning with depthwise separable convolutions, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR-2017, Honolulu, 1251-1258.
- Choudhary, A., Goyal, D. and Letha, S. S., 2020, Infrared thermography-based fault diagnosis of induction motor bearings using machine learning, *IEEE Sensors Journal*, 21 (2), 1727-1734.
- Cireşan, D. C., Meier, U. and Schmidhuber, J., 2012, Transfer learning for Latin and Chinese characters with deep neural networks, *The 2012 International Joint Conference on Neural Networks*, IJCNN-2012, Brisbane, 1-6.
- Es-Sabery, F., Hair, A., Qadir, J., Sainz-De-Abajo, B., García-Zapirain, B. and De La Torre-Díez, I., 2021, Sentence-level classification using parallel fuzzy deep learning classifier, *IEEE Access*, 9, 17943-17985.

- Fang, J., Yang, F., Tong, R., Yu, Q. and Dai, X., 2021, Fault diagnosis of electric transformers based on infrared image processing and semi-supervised learning, *Global Energy Interconnection*, 4 (6), 596-607.
- Fawcett, T., 2006, An introduction to ROC analysis, *Pattern Recognition Letters*, 27 (8), 861-874.
- Granados-Lieberman, D., Razo-Hernandez, J. R., Venegas-Rebollar, V., Olivares-Galvan, J. C. and Valtierra-Rodriguez, M., 2021, Harmonic PMU and fuzzy logic for online detection of short-circuited turns in transformers, *Electric Power Systems Research*, 190, 106862.
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. and Chen, T., 2018, Recent advances in convolutional neural networks, *Pattern recognition*, 77, 354-377.
- Gupta, J., Pathak, S. and Kumar, G., 2022, Deep learning (CNN) and transfer learning: A review. *Journal of Physics: Conference Series*, 2273 (1), 012029.
- He, K., Zhang, X., Ren, S. and Sun, J., 2016, Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, CVPR-2016, Las Vegas, 770-778.
- Hijazi, S., Kumar, R. and Rowen, C., 2015, Using convolutional neural networks for image recognition, *Cadence Design Systems Inc.*, San Jose, CA, USA, 9.
- Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K. Q., 2017, Densely connected convolutional networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR-2017, Honolulu, 4700-4708.
- Huerta-Rosales, J. R., Valtierra-Rodriguez, M., Amezcua-Sanchez, J. P. and Granados-Lieberman, D., 2021, Detection of short-circuited turns in transformer vibration signals using MUSIC-empirical wavelet transform and fractal

dimension, *IEEE International Autumn Meeting on Power, Electronics and Computing*, ROPEC-2021, Ixtapa, 1-7.

Ibrahim, A., Anayi, F. and Packianather, M., 2022(a), New machine learning model-based fault diagnosis of induction motors using thermal images, *2nd International Conference on Advance Computing and Innovative Technologies in Engineering*, ICACITE-2022, Greater Noida, 58-62.

Ibrahim, A., Anayi, F. and Packianather, M., 2022(b), Novel deep network-based transfer learning approach for fault detection of three-phase induction motor, *2nd International Conference on Advance Computing and Innovative Technologies in Engineering*, ICACITE-2022, Greater Noida, 655-659.

Jaiswal, A., Gianchandani, N., Singh, D., Kumar, V. and Kaur, M., 2021, Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning, *Journal of Biomolecular Structure and Dynamics*, 39 (15), 5682-5689.

Jamali, A., Mahdianpari, M., Brisco, B., Granger, J., Mohammadimanesh, F. and Salehi, B., 2021, Comparing solo versus ensemble convolutional neural networks for wetland classification using multi-spectral satellite imagery, *Remote Sensing*, 13 (11), 2046.

Jiang, X., Hu, B., Chandra Satapathy, S., Wang, S. H. and Zhang, Y. D., 2020, Fingerspelling identification for Chinese sign language via AlexNet-based transfer learning and Adam optimizer, *Scientific Programming*, 2020, 1-13.

Jorkesh, S. and Poshtan, J., 2021, Fault diagnosis of an induction motor using data fusion based on neural networks, *IET Science, Measurement & Technology*, 15 (8), 681-689.

Khanjani, M. and Ezoji, M., 2021, Electrical fault detection in three-phase induction motor using deep network-based features of thermograms, *Measurement*, 173, 108622.

- Kim, Y. H., Park, J. B., Chang, M. S., Ryu, J. J., Lim, W. H. and Jung, S. K., 2021, Influence of the depth of the convolutional neural networks on an artificial intelligence model for diagnosis of orthognathic surgery, *Journal of Personalized Medicine*, 11 (5), 356.
- Krizhevsky, A., Sutskever, I. and Hinton, G. E., 2017, Imagenet classification with deep convolutional neural networks, *Communications of the ACM*, 60.6, 84-90.
- Leonardo, M. M., Carvalho, T. J., Rezende, E., Zucchi, R. and Faria, F. A., 2018, Deep feature-based classifiers for fruit fly identification (Diptera: Tephritidae), *31st SIBGRAPI Conference on Graphics, Patterns and Images*, SIBGRAPI-2018, Foz do Iguaçu, 41-47.
- Li, L., Yu, Y., Hong, Q. and Wang, Z., 2020, Numerical investigation on faults diagnosis for AC induction machine by magnetic flux distribution, *Electromagnetic Non-Destructive Evaluation (XXIII)*, 45, 85.
- Li, Z. and Zhou, L., 2020, Simulation research on transformer short circuit, *12th IEEE PES Asia-Pacific Power and Energy Engineering Conference*, APPEEC-2020, Nanjing, 1-4.
- Lozanov, Y., Tzvetkova, S. and Petleshkov, A., 2020, Use of machine learning techniques for classification of thermographic images, *12th Electrical Engineering Faculty Conference*, BULEF-2020, Varna, 1-4.
- Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S. and Zhang, G., 2015, Transfer learning using computational intelligence: A survey, *Knowledge-Based Systems*, 80, 14-23.
- Marin, I., Mladenović, S., Gotovac, S. and Zaharija, G., 2021, Deep-feature-based approach to marine debris classification, *Applied Sciences*, 11 (12), 5644.
- Martinez-Herrera, A. L., Ferrucho-Alvarez, E. R., Ledesma-Carrillo, L. M., Mata-Chavez, R. I., Lopez-Ramirez, M. and Cabal-Yepez, E., 2022, Multiple fault

detection in induction motors through homogeneity and kurtosis computation, *Energies*, 15 (4), 1541.

Muraleedharan, R., Joseph, J. and Varghese, A. S., 2019, An FEM based approach for detection of short circuit faults in transformer windings by magnetic field measurement, *3rd International conference on Electronics, Communication and Aerospace Technology*, ICECA-2019, Coimbatore, 1024-1028.

Najafi, M., Baleghi, Y., Gholamian, S. A. and Mirimani, S. M., 2020, Fault diagnosis of electrical equipment through thermal imaging and interpretable machine learning applied on a newly-introduced dataset, *6th Iranian Conference on Signal Processing and Intelligent Systems*, ICSPIS-2020, Virtual, 1-7.

Ofir, A. and Ben-Artzi, G., 2022, SMM-Conv: Scalar matrix multiplication with zero packing for accelerated convolution, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (CVPR-2022), 3067-3075.

Orenstein, E. C. and Beijbom, O., 2017, Transfer learning and deep feature extraction for planktonic image data sets, *2017 IEEE Winter Conference on Applications of Computer Vision*, WACV-2017, Santa Rosa, 1082-1088.

O'Shea, K. and Nash, R., 2015, An introduction to convolutional neural networks, *arXiv preprint*, arXiv:1511.08458.

Ouyang, X., Zhou, Q., Luo, Z., Jiang, Q., Chen, M. and Li, Y., 2020, Analysis on the magnetic flux leakage distribution in the transformer under different winding deformation and typical working condition, *IEEE International Conference on High Voltage Engineering and Application*, ICHVE-2020, Beijing, 1-4.

Özkan, İ. N. İ. K. and Ülker, E., 2017, Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri, *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6 (3), 85-104.

- Rao, A. S., Nguyen, T., Palaniswami, M. and Ngo, T., 2021, Vision-based automated crack detection using convolutional neural networks for condition assessment of infrastructure, *Structural Health Monitoring*, 20 (4), 2124-2142.
- Sakalli, G. and Koyuncu, H., 2022(a), Discrimination of electrical motor faults in thermal images by using first-order statistics and classifiers, *International Congress on Human-Computer Interaction, Optimization and Robotic Applications*, HORA-2022, Ankara, 1-5.
- Sakalli, G. and Koyuncu, H., 2022(b), Categorization of asynchronous motor situations in infrared images: Analyses with ResNet50, *International Conference on Data Analytics for Business and Industry*, ICDABI-2022, Sakheer, 114-118.
- Sakalli, G. and Koyuncu, H., 2023, Identification of asynchronous motor and transformer situations in thermal images by utilizing transfer learning-based deep learning architectures, *Measurement*, 207, 112380.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L. C., 2018, Mobilenetv2: Inverted residuals and linear bottlenecks, *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, CVPR-2018, Salt Lake City, 4510-4520.
- Saucedo-Dorantes, J. J., Jaen-Cuellar, A. Y., Delgado-Prieto, M., de Jesus Romero-Troncoso, R. and Osornio-Rios, R. A., 2021, Condition monitoring strategy based on an optimized selection of high-dimensional set of hybrid features to diagnose and detect multiple and combined faults in an induction motor, *Measurement*, 178, 109404.
- Tabian, I., Fu, H. and Sharif Khodaei, Z., 2019, A convolutional neural network for impact detection and characterization of complex composite structures, *Sensors*, 19 (22), 4933.
- Wang, Y., Yan, J., Sun, Q., Li, J. and Yang, Z., 2019, A MobileNets convolutional neural network for GIS partial discharge pattern recognition in the ubiquitous

power internet of things context: Optimization, comparison, and application, *IEEE Access*, 7, 150226-150236.

Yaman, O., 2021, An automated faults classification method based on binary pattern and neighborhood component analysis using induction motor, *Measurement*, 168, 108323.

Yi, L., Xu, X., Zhao, J., Duan, R., Guo, Y. and Sun, T., 2022, Fault detection of induction motor based on ALO optimized TKSVD, *Journal of Electrical Engineering & Technology*, 17 (1), 381-393.

Yosinski, J., Clune, J., Bengio, Y. and Lipson, H., 2014, How transferable are features in deep neural networks?, *Advances in Neural Information Processing Systems*, 27.

Zhang, X., Zhou, X., Lin, M. and Sun, J., 2018, Shufflenet: An extremely efficient convolutional neural network for mobile devices, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR-2018*, Salt Lake City, 6848-6856.

EKLER

EK-1 ShuffleNet Mimarisi İçin MATLAB .mlx Uzantılı İşletim Kodu İçyapısı

```

clc;close all;clear;

%=====Loading Initial Parameters=====
trainingSetup =
load("C:\Users\user\Downloads\params_2023_01_01__20_23_48.mat");

%=====Import Data=====
imdsTrain =
imageDatastore("C:\Users\user\Desktop\veriseti","IncludeSubfolders",true,
"LabelSource","foldernames");
[imdsTrain, imdsValidation] = splitEachLabel(imdsTrain,0.8);

% Resize the images to match the network input layer.
augimdsTrain = augmentedImageDatastore([320 240 3],imdsTrain);
augimdsValidation = augmentedImageDatastore([320 240
3],imdsValidation);

%=====Set Training Options=====
opts = trainingOptions("adam",...
"ExecutionEnvironment","auto",...
"InitialLearnRate",0.0001,...
"LearnRateDropFactor",0.8,...
"MaxEpochs",100,...
"MiniBatchSize",16,...
"Shuffle","every-epoch",...
"Plots","training-progress",...
"ValidationData",augimdsValidation);

%=====Create Layer Graph=====
lgraph = layerGraph();

%=====Add Layer Branches=====
tempLayers = [
imageInputLayer([320 240 3],"Name","imageinput")
convolution2dLayer([3 3],24,"Name","node_1","Padding",[1 1 1],
"Stride",[2
2],"Bias",trainingSetup.node_1.Bias,"Weights",trainingSetup.node_1.Weights)

batchNormalizationLayer("Name","node_2","Offset",trainingSetup.node_2.
Offset,"Scale",trainingSetup.node_2.Scale,"TrainedMean",trainingSetup.
node_2.TrainedMean,"TrainedVariance",trainingSetup.node_2.TrainedVariance)
reluLayer("Name","node_3")
maxPooling2dLayer([3 3],"Name","node_4","Padding",[1 1 1],
"Stride",[2 2]);
lgraph = addLayers(lgraph,tempLayers);

tempLayers = [
groupedConvolution2dLayer([1
1],28,4,"Name","node_5","Bias",trainingSetup.node_5.Bias,"Weights",trainingSetup.
node_5.Weights)

batchNormalizationLayer("Name","node_6","Offset",trainingSetup.node_6.
Offset,"Scale",trainingSetup.node_6.Scale,"TrainedMean",trainingSetup.

```

```

node_6.TrainedMean, "TrainedVariance", trainingSetup.node_6.TrainedVariance)
    reluLayer("Name", "node_7")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_8to10", 4)
    groupedConvolution2dLayer([3
3], 1, 112, "Name", "node_11", "Padding", [1 1 1 1], "Stride", [2
2], "Bias", trainingSetup.node_11.Bias, "Weights", trainingSetup.node_11.Weights)

batchNormalizationLayer("Name", "node_12", "Offset", trainingSetup.node_12.Offset, "Scale", trainingSetup.node_12.Scale, "TrainedMean", trainingSetup.node_12.TrainedMean, "TrainedVariance", trainingSetup.node_12.TrainedVariance)
    groupedConvolution2dLayer([1
1], 28, 4, "Name", "node_13", "Bias", trainingSetup.node_13.Bias, "Weights", trainingSetup.node_13.Weights)

batchNormalizationLayer("Name", "node_14", "Offset", trainingSetup.node_14.Offset, "Scale", trainingSetup.node_14.Scale, "TrainedMean", trainingSetup.node_14.TrainedMean, "TrainedVariance", trainingSetup.node_14.TrainedVariance)];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = averagePooling2dLayer([3 3], "Name", "node_15", "Padding", [1 1 1 1], "Stride", [2 2]);
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    depthConcatenationLayer(2, "Name", "node_16")
    reluLayer("Name", "node_17")];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    groupedConvolution2dLayer([1
1], 34, 4, "Name", "node_18", "Bias", trainingSetup.node_18.Bias, "Weights", trainingSetup.node_18.Weights)

batchNormalizationLayer("Name", "node_19", "Offset", trainingSetup.node_19.Offset, "Scale", trainingSetup.node_19.Scale, "TrainedMean", trainingSetup.node_19.TrainedMean, "TrainedVariance", trainingSetup.node_19.TrainedVariance)
    reluLayer("Name", "node_20")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_21to23", 4)
    groupedConvolution2dLayer([3
3], 1, 136, "Name", "node_24", "Padding", [1 1 1 1], "Bias", trainingSetup.node_24.Bias, "Weights", trainingSetup.node_24.Weights)

batchNormalizationLayer("Name", "node_25", "Offset", trainingSetup.node_25.Offset, "Scale", trainingSetup.node_25.Scale, "TrainedMean", trainingSetup.node_25.TrainedMean, "TrainedVariance", trainingSetup.node_25.TrainedVariance)
    groupedConvolution2dLayer([1
1], 34, 4, "Name", "node_26", "Bias", trainingSetup.node_26.Bias, "Weights", trainingSetup.node_26.Weights)

batchNormalizationLayer("Name", "node_27", "Offset", trainingSetup.node_27.Offset, "Scale", trainingSetup.node_27.Scale, "TrainedMean", trainingSetup.node_27.TrainedMean, "TrainedVariance", trainingSetup.node_27.TrainedVariance)];
lgraph = addLayers(lgraph, tempLayers);

```

```

tempLayers = [
    additionLayer(2, "Name", "node_28")
    reluLayer("Name", "node_29")];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    groupedConvolution2dLayer([1
1], 34, 4, "Name", "node_30", "Bias", trainingSetup.node_30.Bias, "Weights", t
rainingSetup.node_30.Weights)

batchNormalizationLayer("Name", "node_31", "Offset", trainingSetup.node_3
1.Offset, "Scale", trainingSetup.node_31.Scale, "TrainedMean", trainingSet
up.node_31.TrainedMean, "TrainedVariance", trainingSetup.node_31.Trained
Variance)
    reluLayer("Name", "node_32")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_33to35", 4)
    groupedConvolution2dLayer([3
3], 1, 136, "Name", "node_36", "Padding", [1 1 1
1], "Bias", trainingSetup.node_36.Bias, "Weights", trainingSetup.node_36.W
eights)

batchNormalizationLayer("Name", "node_37", "Offset", trainingSetup.node_3
7.Offset, "Scale", trainingSetup.node_37.Scale, "TrainedMean", trainingSet
up.node_37.TrainedMean, "TrainedVariance", trainingSetup.node_37.Trained
Variance)
    groupedConvolution2dLayer([1
1], 34, 4, "Name", "node_38", "Bias", trainingSetup.node_38.Bias, "Weights", t
rainingSetup.node_38.Weights)

batchNormalizationLayer("Name", "node_39", "Offset", trainingSetup.node_3
9.Offset, "Scale", trainingSetup.node_39.Scale, "TrainedMean", trainingSet
up.node_39.TrainedMean, "TrainedVariance", trainingSetup.node_39.Trained
Variance)];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    additionLayer(2, "Name", "node_40")
    reluLayer("Name", "node_41")];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    groupedConvolution2dLayer([1
1], 34, 4, "Name", "node_42", "Bias", trainingSetup.node_42.Bias, "Weights", t
rainingSetup.node_42.Weights)

batchNormalizationLayer("Name", "node_43", "Offset", trainingSetup.node_4
3.Offset, "Scale", trainingSetup.node_43.Scale, "TrainedMean", trainingSet
up.node_43.TrainedMean, "TrainedVariance", trainingSetup.node_43.Trained
Variance)
    reluLayer("Name", "node_44")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_45to47", 4)
    groupedConvolution2dLayer([3
3], 1, 136, "Name", "node_48", "Padding", [1 1 1
1], "Bias", trainingSetup.node_48.Bias, "Weights", trainingSetup.node_48.W
eights)

batchNormalizationLayer("Name", "node_49", "Offset", trainingSetup.node_4
9.Offset, "Scale", trainingSetup.node_49.Scale, "TrainedMean", trainingSet
up.node_49.TrainedMean, "TrainedVariance", trainingSetup.node_49.Trained
Variance)

```

```

        groupedConvolution2dLayer([1
1],34,4,"Name","node_50","Bias",trainingSetup.node_50.Bias,"Weights",t
rainingSetup.node_50.Weights)

batchNormalizationLayer("Name","node_51","Offset",trainingSetup.node_5
1.Offset,"Scale",trainingSetup.node_51.Scale,"TrainedMean",trainingSet
up.node_51.TrainedMean,"TrainedVariance",trainingSetup.node_51.Trained
Variance)];
lgraph = addLayers(lgraph,tempLayers);

tempLayers = [
    additionLayer(2,"Name","node_52")
    reluLayer("Name","node_53")];
lgraph = addLayers(lgraph,tempLayers);

tempLayers = averagePooling2dLayer([3 3],"Name","node_64","Padding",[1
1 1 1],"Stride",[2 2]);
lgraph = addLayers(lgraph,tempLayers);

tempLayers = [
    groupedConvolution2dLayer([1
1],34,4,"Name","node_54","Bias",trainingSetup.node_54.Bias,"Weights",t
rainingSetup.node_54.Weights)

batchNormalizationLayer("Name","node_55","Offset",trainingSetup.node_5
5.Offset,"Scale",trainingSetup.node_55.Scale,"TrainedMean",trainingSet
up.node_55.TrainedMean,"TrainedVariance",trainingSetup.node_55.Trained
Variance)
    reluLayer("Name","node_56")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_57to59",4)
    groupedConvolution2dLayer([3
3],1,136,"Name","node_60","Padding",[1 1 1 1],"Stride",[2
2],"Bias",trainingSetup.node_60.Bias,"Weights",trainingSetup.node_60.W
eights)

batchNormalizationLayer("Name","node_61","Offset",trainingSetup.node_6
1.Offset,"Scale",trainingSetup.node_61.Scale,"TrainedMean",trainingSet
up.node_61.TrainedMean,"TrainedVariance",trainingSetup.node_61.Trained
Variance)
    groupedConvolution2dLayer([1
1],34,4,"Name","node_62","Bias",trainingSetup.node_62.Bias,"Weights",t
rainingSetup.node_62.Weights)

batchNormalizationLayer("Name","node_63","Offset",trainingSetup.node_6
3.Offset,"Scale",trainingSetup.node_63.Scale,"TrainedMean",trainingSet
up.node_63.TrainedMean,"TrainedVariance",trainingSetup.node_63.Trained
Variance)];
lgraph = addLayers(lgraph,tempLayers);

tempLayers = [
    depthConcatenationLayer(2,"Name","node_65")
    reluLayer("Name","node_66")];
lgraph = addLayers(lgraph,tempLayers);

tempLayers = [
    groupedConvolution2dLayer([1
1],68,4,"Name","node_67","Bias",trainingSetup.node_67.Bias,"Weights",t
rainingSetup.node_67.Weights)

batchNormalizationLayer("Name","node_68","Offset",trainingSetup.node_6
8.Offset,"Scale",trainingSetup.node_68.Scale,"TrainedMean",trainingSet

```

```

up.node_68.TrainedMean, "TrainedVariance", trainingSetup.node_68.Trained
Variance)
    reluLayer("Name", "node_69")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_70to72", 4)
    groupedConvolution2dLayer([3
3], 1, 272, "Name", "node_73", "Padding", [1 1 1
1], "Bias", trainingSetup.node_73.Bias, "Weights", trainingSetup.node_73.W
eights)

batchNormalizationLayer("Name", "node_74", "Offset", trainingSetup.node_7
4.Offset, "Scale", trainingSetup.node_74.Scale, "TrainedMean", trainingSet
up.node_74.TrainedMean, "TrainedVariance", trainingSetup.node_74.Trained
Variance)
    groupedConvolution2dLayer([1
1], 68, 4, "Name", "node_75", "Bias", trainingSetup.node_75.Bias, "Weights", t
rainingSetup.node_75.Weights)

batchNormalizationLayer("Name", "node_76", "Offset", trainingSetup.node_7
6.Offset, "Scale", trainingSetup.node_76.Scale, "TrainedMean", trainingSet
up.node_76.TrainedMean, "TrainedVariance", trainingSetup.node_76.Trained
Variance)];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    additionLayer(2, "Name", "node_77")
    reluLayer("Name", "node_78")];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    groupedConvolution2dLayer([1
1], 68, 4, "Name", "node_79", "Bias", trainingSetup.node_79.Bias, "Weights", t
rainingSetup.node_79.Weights)

batchNormalizationLayer("Name", "node_80", "Offset", trainingSetup.node_8
0.Offset, "Scale", trainingSetup.node_80.Scale, "TrainedMean", trainingSet
up.node_80.TrainedMean, "TrainedVariance", trainingSetup.node_80.Trained
Variance)
    reluLayer("Name", "node_81")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_82to84", 4)
    groupedConvolution2dLayer([3
3], 1, 272, "Name", "node_85", "Padding", [1 1 1
1], "Bias", trainingSetup.node_85.Bias, "Weights", trainingSetup.node_85.W
eights)

batchNormalizationLayer("Name", "node_86", "Offset", trainingSetup.node_8
6.Offset, "Scale", trainingSetup.node_86.Scale, "TrainedMean", trainingSet
up.node_86.TrainedMean, "TrainedVariance", trainingSetup.node_86.Trained
Variance)
    groupedConvolution2dLayer([1
1], 68, 4, "Name", "node_87", "Bias", trainingSetup.node_87.Bias, "Weights", t
rainingSetup.node_87.Weights)

batchNormalizationLayer("Name", "node_88", "Offset", trainingSetup.node_8
8.Offset, "Scale", trainingSetup.node_88.Scale, "TrainedMean", trainingSet
up.node_88.TrainedMean, "TrainedVariance", trainingSetup.node_88.Trained
Variance)];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    additionLayer(2, "Name", "node_89")
    reluLayer("Name", "node_90")];

```



```

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [
    groupedConvolution2dLayer([1
1],68,4,"Name","node_91","Bias",trainingSetup.node_91.Bias,"Weights",t
rainingSetup.node_91.Weights)

batchNormalizationLayer("Name","node_92","Offset",trainingSetup.node_9
2.Offset,"Scale",trainingSetup.node_92.Scale,"TrainedMean",trainingSet
up.node_92.TrainedMean,"TrainedVariance",trainingSetup.node_92.Trained
Variance)
    reluLayer("Name","node_93")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_94to96",4)
    groupedConvolution2dLayer([3
3],1,272,"Name","node_97","Padding",[1 1 1
1],"Bias",trainingSetup.node_97.Bias,"Weights",trainingSetup.node_97.W
eights)

batchNormalizationLayer("Name","node_98","Offset",trainingSetup.node_9
8.Offset,"Scale",trainingSetup.node_98.Scale,"TrainedMean",trainingSet
up.node_98.TrainedMean,"TrainedVariance",trainingSetup.node_98.Trained
Variance)
    groupedConvolution2dLayer([1
1],68,4,"Name","node_99","Bias",trainingSetup.node_99.Bias,"Weights",t
rainingSetup.node_99.Weights)

batchNormalizationLayer("Name","node_100","Offset",trainingSetup.node_
100.Offset,"Scale",trainingSetup.node_100.Scale,"TrainedMean",training
Setup.node_100.TrainedMean,"TrainedVariance",trainingSetup.node_100.Tr
ainedVariance)];
lgraph = addLayers(lgraph,tempLayers);

tempLayers = [
    additionLayer(2,"Name","node_101")
    reluLayer("Name","node_102")];
lgraph = addLayers(lgraph,tempLayers);

tempLayers = [
    groupedConvolution2dLayer([1
1],68,4,"Name","node_103","Bias",trainingSetup.node_103.Bias,"Weights"
,trainingSetup.node_103.Weights)

batchNormalizationLayer("Name","node_104","Offset",trainingSetup.node_
104.Offset,"Scale",trainingSetup.node_104.Scale,"TrainedMean",training
Setup.node_104.TrainedMean,"TrainedVariance",trainingSetup.node_104.Tr
ainedVariance)
    reluLayer("Name","node_105")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_106to108",4)
    groupedConvolution2dLayer([3
3],1,272,"Name","node_109","Padding",[1 1 1
1],"Bias",trainingSetup.node_109.Bias,"Weights",trainingSetup.node_109
.Weights)

batchNormalizationLayer("Name","node_110","Offset",trainingSetup.node_
110.Offset,"Scale",trainingSetup.node_110.Scale,"TrainedMean",training
Setup.node_110.TrainedMean,"TrainedVariance",trainingSetup.node_110.Tr
ainedVariance)
    groupedConvolution2dLayer([1
1],68,4,"Name","node_111","Bias",trainingSetup.node_111.Bias,"Weights"
,trainingSetup.node_111.Weights)

```

```

batchNormalizationLayer("Name", "node_112", "Offset", trainingSetup.node_
112.Offset, "Scale", trainingSetup.node_112.Scale, "TrainedMean", training
Setup.node_112.TrainedMean, "TrainedVariance", trainingSetup.node_112.Tr
ainedVariance)];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    additionLayer(2, "Name", "node_113")
    reluLayer("Name", "node_114")];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    groupedConvolution2dLayer([1
1], 68, 4, "Name", "node_115", "Bias", trainingSetup.node_115.Bias, "Weights"
, trainingSetup.node_115.Weights)

batchNormalizationLayer("Name", "node_116", "Offset", trainingSetup.node_
116.Offset, "Scale", trainingSetup.node_116.Scale, "TrainedMean", training
Setup.node_116.TrainedMean, "TrainedVariance", trainingSetup.node_116.Tr
ainedVariance)
    reluLayer("Name", "node_117")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_118to120", 4)
    groupedConvolution2dLayer([3
3], 1, 272, "Name", "node_121", "Padding", [1 1 1
1], "Bias", trainingSetup.node_121.Bias, "Weights", trainingSetup.node_121
.Weights)

batchNormalizationLayer("Name", "node_122", "Offset", trainingSetup.node_
122.Offset, "Scale", trainingSetup.node_122.Scale, "TrainedMean", training
Setup.node_122.TrainedMean, "TrainedVariance", trainingSetup.node_122.Tr
ainedVariance)
    groupedConvolution2dLayer([1
1], 68, 4, "Name", "node_123", "Bias", trainingSetup.node_123.Bias, "Weights"
, trainingSetup.node_123.Weights)

batchNormalizationLayer("Name", "node_124", "Offset", trainingSetup.node_
124.Offset, "Scale", trainingSetup.node_124.Scale, "TrainedMean", training
Setup.node_124.TrainedMean, "TrainedVariance", trainingSetup.node_124.Tr
ainedVariance)];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    additionLayer(2, "Name", "node_125")
    reluLayer("Name", "node_126")];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    groupedConvolution2dLayer([1
1], 68, 4, "Name", "node_127", "Bias", trainingSetup.node_127.Bias, "Weights"
, trainingSetup.node_127.Weights)

batchNormalizationLayer("Name", "node_128", "Offset", trainingSetup.node_
128.Offset, "Scale", trainingSetup.node_128.Scale, "TrainedMean", training
Setup.node_128.TrainedMean, "TrainedVariance", trainingSetup.node_128.Tr
ainedVariance)
    reluLayer("Name", "node_129")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_130to132", 4)
    groupedConvolution2dLayer([3
3], 1, 272, "Name", "node_133", "Padding", [1 1 1
1]

```

```

1], "Bias", trainingSetup.node_133.Bias, "Weights", trainingSetup.node_133
.Weights)

batchNormalizationLayer ("Name", "node_134", "Offset", trainingSetup.node_
134.Offset, "Scale", trainingSetup.node_134.Scale, "TrainedMean", training
Setup.node_134.TrainedMean, "TrainedVariance", trainingSetup.node_134.Tr
ainedVariance)
    groupedConvolution2dLayer ([1
1], 68, 4, "Name", "node_135", "Bias", trainingSetup.node_135.Bias, "Weights"
, trainingSetup.node_135.Weights)

batchNormalizationLayer ("Name", "node_136", "Offset", trainingSetup.node_
136.Offset, "Scale", trainingSetup.node_136.Scale, "TrainedMean", training
Setup.node_136.TrainedMean, "TrainedVariance", trainingSetup.node_136.Tr
ainedVariance)];
lgraph = addLayers (lgraph, tempLayers);

tempLayers = [
    additionLayer (2, "Name", "node_137")
    reluLayer ("Name", "node_138")];
lgraph = addLayers (lgraph, tempLayers);

tempLayers = [
    groupedConvolution2dLayer ([1
1], 68, 4, "Name", "node_139", "Bias", trainingSetup.node_139.Bias, "Weights"
, trainingSetup.node_139.Weights)

batchNormalizationLayer ("Name", "node_140", "Offset", trainingSetup.node_
140.Offset, "Scale", trainingSetup.node_140.Scale, "TrainedMean", training
Setup.node_140.TrainedMean, "TrainedVariance", trainingSetup.node_140.Tr
ainedVariance)
    reluLayer ("Name", "node_141")
    nnet.shufflenet.layer.ChannelShufflingLayer ("shuffle_142to144", 4)
    groupedConvolution2dLayer ([3
3], 1, 272, "Name", "node_145", "Padding", [1 1 1
1], "Bias", trainingSetup.node_145.Bias, "Weights", trainingSetup.node_145
.Weights)

batchNormalizationLayer ("Name", "node_146", "Offset", trainingSetup.node_
146.Offset, "Scale", trainingSetup.node_146.Scale, "TrainedMean", training
Setup.node_146.TrainedMean, "TrainedVariance", trainingSetup.node_146.Tr
ainedVariance)
    groupedConvolution2dLayer ([1
1], 68, 4, "Name", "node_147", "Bias", trainingSetup.node_147.Bias, "Weights"
, trainingSetup.node_147.Weights)

batchNormalizationLayer ("Name", "node_148", "Offset", trainingSetup.node_
148.Offset, "Scale", trainingSetup.node_148.Scale, "TrainedMean", training
Setup.node_148.TrainedMean, "TrainedVariance", trainingSetup.node_148.Tr
ainedVariance)];
lgraph = addLayers (lgraph, tempLayers);

tempLayers = [
    additionLayer (2, "Name", "node_149")
    reluLayer ("Name", "node_150")];
lgraph = addLayers (lgraph, tempLayers);

tempLayers = averagePooling2dLayer ([3
3], "Name", "node_161", "Padding", [1 1 1 1], "Stride", [2 2]);
lgraph = addLayers (lgraph, tempLayers);

```

```

tempLayers = [
    groupedConvolution2dLayer([1
1],68,4,"Name","node_151","Bias",trainingSetup.node_151.Bias,"Weights"
,trainingSetup.node_151.Weights)

batchNormalizationLayer("Name","node_152","Offset",trainingSetup.node_
152.Offset,"Scale",trainingSetup.node_152.Scale,"TrainedMean",training
Setup.node_152.TrainedMean,"TrainedVariance",trainingSetup.node_152.Tr
ainedVariance)
    reluLayer("Name","node_153")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_154to156",4)
    groupedConvolution2dLayer([3
3],1,272,"Name","node_157","Padding",[1 1 1 1],"Stride",[2
2],"Bias",trainingSetup.node_157.Bias,"Weights",trainingSetup.node_157
.Weights)

batchNormalizationLayer("Name","node_158","Offset",trainingSetup.node_
158.Offset,"Scale",trainingSetup.node_158.Scale,"TrainedMean",training
Setup.node_158.TrainedMean,"TrainedVariance",trainingSetup.node_158.Tr
ainedVariance)
    groupedConvolution2dLayer([1
1],68,4,"Name","node_159","Bias",trainingSetup.node_159.Bias,"Weights"
,trainingSetup.node_159.Weights)

batchNormalizationLayer("Name","node_160","Offset",trainingSetup.node_
160.Offset,"Scale",trainingSetup.node_160.Scale,"TrainedMean",training
Setup.node_160.TrainedMean,"TrainedVariance",trainingSetup.node_160.Tr
ainedVariance)];
lgraph = addLayers(lgraph,tempLayers);

tempLayers = [
    depthConcatenationLayer(2,"Name","node_162")
    reluLayer("Name","node_163")];
lgraph = addLayers(lgraph,tempLayers);

tempLayers = [
    groupedConvolution2dLayer([1
1],136,4,"Name","node_164","Bias",trainingSetup.node_164.Bias,"Weights"
,trainingSetup.node_164.Weights)

batchNormalizationLayer("Name","node_165","Offset",trainingSetup.node_
165.Offset,"Scale",trainingSetup.node_165.Scale,"TrainedMean",training
Setup.node_165.TrainedMean,"TrainedVariance",trainingSetup.node_165.Tr
ainedVariance)
    reluLayer("Name","node_166")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_167to169",4)
    groupedConvolution2dLayer([3
3],1,544,"Name","node_170","Padding",[1 1 1
1],"Bias",trainingSetup.node_170.Bias,"Weights",trainingSetup.node_170
.Weights)

batchNormalizationLayer("Name","node_171","Offset",trainingSetup.node_
171.Offset,"Scale",trainingSetup.node_171.Scale,"TrainedMean",training
Setup.node_171.TrainedMean,"TrainedVariance",trainingSetup.node_171.Tr
ainedVariance)
    groupedConvolution2dLayer([1
1],136,4,"Name","node_172","Bias",trainingSetup.node_172.Bias,"Weights"
,trainingSetup.node_172.Weights)

batchNormalizationLayer("Name","node_173","Offset",trainingSetup.node_
173.Offset,"Scale",trainingSetup.node_173.Scale,"TrainedMean",training

```

```

Setup.node_173.TrainedMean, "TrainedVariance", trainingSetup.node_173.Tr
ainedVariance)];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    additionLayer(2, "Name", "node_174")
    reluLayer("Name", "node_175")];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    groupedConvolution2dLayer([1
1], 136, 4, "Name", "node_176", "Bias", trainingSetup.node_176.Bias, "Weights
", trainingSetup.node_176.Weights)

    batchNormalizationLayer("Name", "node_177", "Offset", trainingSetup.node_
177.Offset, "Scale", trainingSetup.node_177.Scale, "TrainedMean", training
Setup.node_177.TrainedMean, "TrainedVariance", trainingSetup.node_177.Tr
ainedVariance)
    reluLayer("Name", "node_178")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_179to181", 4)
    groupedConvolution2dLayer([3
3], 1, 544, "Name", "node_182", "Padding", [1 1 1
1], "Bias", trainingSetup.node_182.Bias, "Weights", trainingSetup.node_182
.Weights)

    batchNormalizationLayer("Name", "node_183", "Offset", trainingSetup.node_
183.Offset, "Scale", trainingSetup.node_183.Scale, "TrainedMean", training
Setup.node_183.TrainedMean, "TrainedVariance", trainingSetup.node_183.Tr
ainedVariance)
    groupedConvolution2dLayer([1
1], 136, 4, "Name", "node_184", "Bias", trainingSetup.node_184.Bias, "Weights
", trainingSetup.node_184.Weights)

    batchNormalizationLayer("Name", "node_185", "Offset", trainingSetup.node_
185.Offset, "Scale", trainingSetup.node_185.Scale, "TrainedMean", training
Setup.node_185.TrainedMean, "TrainedVariance", trainingSetup.node_185.Tr
ainedVariance)];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    additionLayer(2, "Name", "node_186")
    reluLayer("Name", "node_187")];
lgraph = addLayers(lgraph, tempLayers);

tempLayers = [
    groupedConvolution2dLayer([1
1], 136, 4, "Name", "node_188", "Bias", trainingSetup.node_188.Bias, "Weights
", trainingSetup.node_188.Weights)

    batchNormalizationLayer("Name", "node_189", "Offset", trainingSetup.node_
189.Offset, "Scale", trainingSetup.node_189.Scale, "TrainedMean", training
Setup.node_189.TrainedMean, "TrainedVariance", trainingSetup.node_189.Tr
ainedVariance)
    reluLayer("Name", "node_190")
    nnet.shufflenet.layer.ChannelShufflingLayer("shuffle_191to193", 4)
    groupedConvolution2dLayer([3
3], 1, 544, "Name", "node_194", "Padding", [1 1 1
1], "Bias", trainingSetup.node_194.Bias, "Weights", trainingSetup.node_194
.Weights)

    batchNormalizationLayer("Name", "node_195", "Offset", trainingSetup.node_

```

```

195.Offset,"Scale",trainingSetup.node_195.Scale,"TrainedMean",training
Setup.node_195.TrainedMean,"TrainedVariance",trainingSetup.node_195.Tr
ainedVariance)
    groupedConvolution2dLayer([1
1],136,4,"Name","node_196","Bias",trainingSetup.node_196.Bias,"Weights
",trainingSetup.node_196.Weights)

batchNormalizationLayer("Name","node_197","Offset",trainingSetup.node_
197.Offset,"Scale",trainingSetup.node_197.Scale,"TrainedMean",training
Setup.node_197.TrainedMean,"TrainedVariance",trainingSetup.node_197.Tr
ainedVariance)];
lgraph = addLayers(lgraph,tempLayers);

tempLayers = [
    additionLayer(2,"Name","node_198")
    reluLayer("Name","node_199")
    globalAveragePooling2dLayer("Name","node_200")
    fullyConnectedLayer(20,"Name","fc")
    softmaxLayer("Name","node_203")
    classificationLayer("Name","classoutput")];
lgraph = addLayers(lgraph,tempLayers);

% clean up helper variable
clear tempLayers;

%=====Connect Layer Branches=====
lgraph = connectLayers(lgraph,"node_4","node_5");
lgraph = connectLayers(lgraph,"node_4","node_15");
lgraph = connectLayers(lgraph,"node_15","node_16/in2");
lgraph = connectLayers(lgraph,"node_14","node_16/in1");
lgraph = connectLayers(lgraph,"node_17","node_18");
lgraph = connectLayers(lgraph,"node_17","node_28/in2");
lgraph = connectLayers(lgraph,"node_27","node_28/in1");
lgraph = connectLayers(lgraph,"node_29","node_30");
lgraph = connectLayers(lgraph,"node_29","node_40/in2");
lgraph = connectLayers(lgraph,"node_39","node_40/in1");
lgraph = connectLayers(lgraph,"node_41","node_42");
lgraph = connectLayers(lgraph,"node_41","node_52/in2");
lgraph = connectLayers(lgraph,"node_51","node_52/in1");
lgraph = connectLayers(lgraph,"node_53","node_64");
lgraph = connectLayers(lgraph,"node_53","node_54");
lgraph = connectLayers(lgraph,"node_64","node_65/in2");
lgraph = connectLayers(lgraph,"node_63","node_65/in1");
lgraph = connectLayers(lgraph,"node_66","node_67");
lgraph = connectLayers(lgraph,"node_66","node_77/in2");
lgraph = connectLayers(lgraph,"node_76","node_77/in1");
lgraph = connectLayers(lgraph,"node_78","node_79");
lgraph = connectLayers(lgraph,"node_78","node_89/in2");
lgraph = connectLayers(lgraph,"node_88","node_89/in1");
lgraph = connectLayers(lgraph,"node_90","node_91");
lgraph = connectLayers(lgraph,"node_90","node_101/in2");
lgraph = connectLayers(lgraph,"node_100","node_101/in1");
lgraph = connectLayers(lgraph,"node_102","node_103");
lgraph = connectLayers(lgraph,"node_102","node_113/in2");
lgraph = connectLayers(lgraph,"node_112","node_113/in1");
lgraph = connectLayers(lgraph,"node_114","node_115");
lgraph = connectLayers(lgraph,"node_114","node_125/in2");
lgraph = connectLayers(lgraph,"node_124","node_125/in1");
lgraph = connectLayers(lgraph,"node_126","node_127");
lgraph = connectLayers(lgraph,"node_126","node_137/in2");
lgraph = connectLayers(lgraph,"node_136","node_137/in1");

```

```
lgraph = connectLayers(lgraph, "node_138", "node_139");
lgraph = connectLayers(lgraph, "node_138", "node_149/in2");
lgraph = connectLayers(lgraph, "node_148", "node_149/in1");
lgraph = connectLayers(lgraph, "node_150", "node_161");
lgraph = connectLayers(lgraph, "node_150", "node_151");
lgraph = connectLayers(lgraph, "node_161", "node_162/in2");
lgraph = connectLayers(lgraph, "node_160", "node_162/in1");
lgraph = connectLayers(lgraph, "node_163", "node_164");
lgraph = connectLayers(lgraph, "node_163", "node_174/in2");
lgraph = connectLayers(lgraph, "node_173", "node_174/in1");
lgraph = connectLayers(lgraph, "node_175", "node_176");
lgraph = connectLayers(lgraph, "node_175", "node_186/in2");
lgraph = connectLayers(lgraph, "node_185", "node_186/in1");
lgraph = connectLayers(lgraph, "node_187", "node_188");
lgraph = connectLayers(lgraph, "node_187", "node_198/in2");
lgraph = connectLayers(lgraph, "node_197", "node_198/in1");

%=====Train Network=====

[net, traininfo] = trainNetwork(augimdsTrain,lgraph,opts);
```



EK-2 Tez Çalışması Kapsamında Hazırlanan Eser Tanımları

Sakalli, G. and Koyuncu, H., 2022(a), Discrimination of electrical motor faults in thermal images by using first-order statistics and classifiers, *International Congress on Human-Computer Interaction, Optimization and Robotic Applications*, HORA-2022, Ankara, 1-5. (***Yüksek lisans tezinden yapılmıştır***)

Sakalli, G. and Koyuncu, H., 2022(b), Categorization of asynchronous motor situations in infrared images: Analyses with ResNet50, *International Conference on Data Analytics for Business and Industry*, ICDABI-2022, Sakheer, 114-118. (***Yüksek lisans tezinden yapılmıştır***)

Sakalli, G. and Koyuncu, H., 2023, Identification of asynchronous motor and transformer situations in thermal images by utilizing transfer learning-based deep learning architectures, *Measurement*, 207, 112380. (***Yüksek lisans tezinden yapılmıştır***)