Biyomedikal Görüntülerin Sınıflandırılmasında Swish Aktivasyon Fonksiyonunun Performans Analizi

Sama Mohamed Elsayed Mosilhy ABDALLA

T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİYOMEDİKAL GÖRÜNTÜLERİN SINIFLANDIRILMASINDA SWİSH AKTİVASYON FONKSİYONUNUN PERFORMANS ANALİZİ

Sama Mohamed Elsayed Mosilhy ABDALLA
0009-0005-4656-9766

Doç. Dr. Neyir OZCAN SEMERCI
(Danışman)

YÜKSEK LİSANS
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2025
Her Hakkı Saklıdır

ÖZET

BİYOMEDİKAL GÖRÜNTÜLERİN SINIFLANDIRILMASINDA SWİSH AKTİVASYON FONKSİYONUNUN PERFORMANS ANALİZİ

Sama Mohamed Elsayed Mosilhy ABDALLA

Yüksek Lisans

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik Müdenisliği Anabilim Dalı

Danışman: Doç. Dr. Neyir OZCAN SEMERCI

Aktivasyon fonksiyonları, yapay sinir ağları ve derin öğrenme modellerinde nöron karakteristiğini belirleyen temel bileşen olarak önemli bir rol oynamaktadırlar. Derin öğrenme modellerinin performansını artırmak için uygun bir aktivasyon fonksiyonu seçmek kritik bir öneme sahiptir. Bu fonksiyonlar, ağa verilen girdilere karşılık gelen çıktıları belirleyerek modelin öğrenme kapasitesini doğrudan etkilemektedirler. Ağın doğrusal olmayan ilişkileri öğrenebilmesi için modelin de doğrusal olmayan bir özellik göstermesi önemlidir. Böylece ağ karmaşık ve çok boyutlu ilişkileri öğrenebilir. Bu doğrusal olmama durumu, aktivasyon fonksiyonları aracılığıyla modele dahil edilir. Literatürde farklı uygulamalar için kullanılan doğrusal olmayan aktivasyon fonksiyonları mevcuttur: ReLU, Leakly ReLU, Swish, P-Swish, SaRa …vs. Swish aktivasyon fonksiyonu; düzgün (smooth) yapısı, alt sınıra sahip olması ve monoton olmama özelliği sayesinde sinir ağlarının daha karmaşık ilişkileri öğrenmesini kolaylaştırır ve öğrenme sürecinde daha stabil ve esnek bir performans sunmaktadır. ReLU fonksiyonu nöron girişine uygulanan negatif değerler için nöronun inaktif olmasına neden olurken Swish aktivasyon fonksiyonu nöronun çıktı üretebilmesine olanak tanır. Böylece bu değerler için gradyen ölmesi problemin ile daha az karşılaşılır. Bu durum bilgi kaybını azaltır ve modelin daha fazla ayrıntıyı öğrenmesine olanak tanır. Aynı amaçla üretilmiş olan Leakly ReLU fonksiyonu ise monoton bir yapıya sahiptir. Swish fonksiyonunun monoton olmayan yapısı ise modelin karmaşık ilişkileri daha iyi öğrenmesini sağlar. Gerçekleştirilen tez çalışmasında biyomedikal veriler kullanılarak swish fonksiyonunun farklı türleri için farklı derin öğrenme mimarileri kullanılarak fonksiyonun diğer aktivasyon fonksiyonları ile performans kıyaslaması yapılmıştır. Çalışmada kullanılan veri seti 3 farklı tür beyin tümörüne (meningioma, glioma ve pituitary) ait manyetik rezonans (MR) görüntülerini içermektedir. Bu görüntüler farklı derin öğrenme mimarileri ve farklı aktivasyon fonksiyonları kullanılarak sınıflandırılmıştır. Elde edilen sonuçlar yüksek doğruluk ve detayların korunmasını gerektiren bu tip problemlerde swish fonksiyonlarının literatürde kullanılan diğer aktivasyon fonksiyonlarına göre daha yüksek bir sınıflandırma performansına sahip olduğunu açıkça göstermektedir.

Anahtar Kelimeler: Derin Öğrenme, Aktivasyon Fonksiyonları, Swish Fonksiyonları
2015, vii + 149 pages.

ABSTRACT

PERFORMANCE ANALYSİS OF SWISH FUNCTION ON CLASSIFICATION OF BIOMEDICAL IMAGES

Sama Mohamed Elsayed Mosilhy ABDALLA

MSc Thesis

Bursa Uludağ University
Graduate School of Natural and Applied Sciences
Department of Electronic Engineering

Supervisor: Assoc. Prof. Dr. Neyir OZCAN SEMERCI

Activation functions play a crucial role as fundamental components that determine neuron characteristics in artificial neural networks and deep learning models. Selecting an appropriate activation function is critically important for enhancing the performance of deep learning models. These functions directly affect the model's learning capacity by determining the outputs corresponding to the inputs provided to the network. For the network to learn nonlinear relationships, it is essential for the model to exhibit nonlinear characteristics. In this way, the network can learn complex and multidimensional relationships. This nonlinearity is incorporated into the model through activation functions.

There are various nonlinear activation functions used in the literature for different applications, such as ReLU, Leaky ReLU, Swish, P-Swish, and SaRa. The Swish activation function, with its smooth structure, boundedness from below, and non-monotonic nature, facilitates neural networks in learning more complex relationships and offers a more stable and flexible performance during the learning process. While the ReLU function causes neurons to become inactive for negative input values, the Swish activation function allows neurons to produce outputs, thus reducing the likelihood of encountering the vanishing gradient problem for these values. This reduces information loss and enables the model to learn more detailed features. Although the Leaky ReLU function, designed for a similar purpose, has a monotonic structure, the non-monotonic nature of the Swish function allows the model to better capture complex relationships.

In this thesis study, biomedical data were used to compare the performance of different types of Swish functions with other activation functions across various deep learning architectures. The dataset used in the study consists of magnetic resonance (MR) images of three different types of brain tumors: meningioma, glioma, and pituitary. These images were classified using different deep learning architectures and activation functions. The obtained results clearly demonstrate that Swish functions achieve higher classification performance compared to other activation functions commonly used in the literature, particularly in problems requiring high accuracy and the preservation of fine details.

Key words: Deep Learning, Artificial Neural Networks, Activation Functions, Swish Function

2015, vii + 149 pages.

TEŞEKKÜR

Öncelikle, lisans ve yüksek lisans eğitimim süresince bilgi, deneyim ve desteğini esirgemeyen, aynı zamanda yüksek lisans yapma fırsatını bana sunarak akademik gelişimime katkıda bulunan ve bu süreçte bilgi birikimlerini benimle cömertçe paylaşan ve akademik yolculuğumda yol gösterici olan danışmanım Sayın Doç. Dr. Neyir OZCAN SEMERCI'a en içten teşekkürlerimi sunarım.

Hayatım boyunca bana kendimden çok daha fazla inanan, her zaman yanımda olan, destekleriyle beni motive eden sevgili babam Mohamed Gaballa ve annem Wesam Metwally teşekkürlerimi sunarım.

Tez sürecim boyunca her zaman yanımda olan, sabrıyla, sevgisiyle ve anlayışıyla bana güç veren kıymetli eşim Khaled Taha'ya en içten teşekkürlerimi sunarım.

Hayatım boyunca yanımda olan, bana koşulsuz sevgi ve destek veren sevgili kardeşlerim; Gana Gaballa, Mostafa Gaballa ve Basmala Gaballa'ya teşekkürlerimi sunarım.

Sama Mohamed Elsayed Mosilhy
ABDALLA
22/06/2025

İÇİNDEKİLER

Sayfa

ÖZET vi
ABSTRACT vii
TEŞEKKÜR viii
SİMGELER ve KISALTMALAR DİZİNİ x
1. GİRİŞ 1
2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI 4
3. MATERYAL ve YÖNTEM 10
3.1. Veri Seti 10
3.2. Aktivasyon fonksiyonları 11
3.2.1. Sigmoid aktivasyon fonksiyonu 13
3.2.2. Tanh aktivasyon fonksiyonu 14
3.2.3. ReLU aktivasyon fonksiyonu 15
3.2.4. LeakyRelu aktivasyon fonksiyonu 17
3.2.5. Swish aktivasyon fonksiyonu 17
3.2.6. P-Swish aktivasyon fonksiyonu 19
3.2.7. SaRa aktivasyon fonksiyonu 21
3.2.8. Softmax aktivasyon fonksiyonu 22
3.3. Evrişimli Sinir Ağı 23
3.3.1. VGG16 mimari 24
3.3.2. DenseNet mimari 24
3.3.3. Evrişimli nöral ağ modeli V3 mimari 26
3.4. Yöntem 28
3.5. Değerlendirme Metrikleri 31
3.5.1. Model Performansının Değerlendirilmesi 31
3.5.2. Grafiksel Analiz 32
3.5.3. Kullanılan Metrikler 33
4. BULGULAR 34
4.1. VGG16 Modelin Bulguları ve Tartışması 34
4.2. DenseNet121 Modelin Bulguları ve Tartışması 46
4.3. Inception-V3 Modelin Bulguları ve Tartışması 58
5. SONUÇ 70
5.1. VGG16 Modelin Sonuçları 70
5.2. DenseNet121 Modelin Sonuçları 72
5.3. Inception-V3 Modelin Sonuçları 74
5.4. Sonuç 76
KAYNAKLAR 78
ÖZGEÇMİŞ 80

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler	**Açıklama**
%	Yüzde

Kısaltmalar	**Açıklama**
vb	ve benzerleri

ŞEKİLLER DİZİNİ

Sayfa

Şekil 3.1. Veri Setinin dört sınıfı A) Tümör yok B) Glioma C) Meningioma D)Pituitar 10
Şekil 3.2. Girişler, Toplama fonksiyonu, Aktivasyon fonksiyonu 11
Şekil 3.3. Yapay sinir ağnın blok diyagramı 12
Şekil 3.4. Sigmoid fonksiyonun matematiksel grafiği 13
Şekil 3.5. Tanh fonksiyonun matematiksel grafiği 14
Şekil 3.6. ReLU fonksiyonun matematiksel grafiği 15
Şekil 3.7. ReLU fonksiyonun türevinin matematiksel grafiği 16
Şekil 3.8. Swish fonksiyonun matematiksel grafiği 17
Şekil 3.9. Swish fonksiyonun birinci ve ikinci türevlerinin matematiksel grafiği .. 18
Şekil 3.10. P-Swish fonksiyonun matematiksel grafiği 20
Şekil 3.11. SaRa fonksiyonun matematiksel grafiği 22
Şekil 3.12. CNN blok diyagramı 23
Şekil 3.13. VGG16 mimarisinin blok diyagramı 24
Şekil 3.14. DanseNet mimarisinin blok diyagramı 25
Şekil 3.15. Nöral Yapısı 27
Şekil 3.16. Model ile eklenen dense katmanları ve nöron sayıları..... 29
Şekil 4.1. VGG16 modelin eğitim ve doğrulama doğruluk grafiği 35
Şekil 4.2. VGG16 modelin eğitim ve doğrulama kayıp grafiği 35
Şekil 4.3. ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama doğruluk grafiği 37
Şekil 4.4. ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama kayıp grafiği 37
Şekil 4.5. Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama doğruluk grafiği 39
Şekil 4.6. Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama kayıp grafiği 39
Şekil 4.7. Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama doğruluk grafiği 41
Şekil 4.8. Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama kayıp grafiği 41
Şekil 4.9. P-Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama doğruluk grafiği 43
Şekil 4.10. P-Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama kayıp grafiği 43
Şekil 4.11. SaRa aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama doğruluk grafiği 45
Şekil 4.12. SaRa aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama kayıp grafiği 45
Şekil 4.13. DenseNet121 modelin eğitim ve doğrulama doğruluk grafiği 47
Şekil 4.14. DenseNet121 modelin eğitim ve doğrulama kayıp grafiği 47
Şekil 4.15. ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama doğruluk grafiği 49
Şekil 4.16. ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama kayıp grafiği 49

Şekil 4.17. Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama doğruluk grafiği..... 51
Şekil 4.18. Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelim eğitim ve doğrulama kayıp grafiği 51
Şekil 4.19. Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama doğruluk grafiği..... 53
Şekil 4.20. Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama kayıp grafiği 53
Şekil 4.21. P-Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama doğruluk grafiği..... 55
Şekil 4.22. P-Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama kayıp grafiği 55
Şekil 4.23. SaRa aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama doğruluk grafiği..... 57
Şekil 4.24. SaRa aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama kayıp grafiği 57
Şekil 4.25. Inception-V3 modelin eğitim ve doğrulama doğruluk grafiği..... 59
Şekil 4.26. Inception-V3 modelin eğitim ve doğrulama kayıp grafiği 59
Şekil 4.27. ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama doğruluk grafiği 61
Şekil 4.28. ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama kayıp grafiği..... 61
Şekil 4.29. Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama doğruluk grafiği 63
Şekil 4.30. Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama kayıp grafiği..... 63
Şekil 4.31. Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama doğruluk grafiği 65
Şekil 4.32. Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama kayıp grafiği..... 65
Şekil 4.33. P-Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama doğruluk grafiği 67
Şekil 4.34. P-Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama kayıp grafiği..... 67
Şekil 4.35. SaRa aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama doğruluk grafiği 69
Şekil 4.36. SaRa aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelim eğitim ve doğrulama doğruluk grafiği…… 69
Şekil 5.1. VGG16 mimarisinin sonuçları..... 71
Şekil 5.2. DenseNet121 mimarisinin sonuçları..... 73
Şekil 5.3. Inception-V3 mimarisinin sonuçları 75

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 5. 1. VGG16 modelin sonuçları 71
Çizelge 5. 2. DenseNet121modelin sonuçları 73
Çizelge 5. 3. Inception-V3 modelin sonuçları 75

1. GİRİŞ

Derin öğrenmenin algılama ve sınıflandırma konularındaki yüksek başarımı, özellikle biyomedikal uygulamalarda oldukça değerli hale gelmiştir. Bu bağlamda, Evrişimsel Sinir Ağları (Convolutional Neural Networks – CNN), görüntü tanıma, analiz ve sınıflandırma gibi görevlerde etkin bir yöntem olarak öne çıkmaktadır. CNN mimarisinin başarısını belirleyen temel bileşenlerden biri de aktivasyon fonksiyonlarıdır.

Aktivasyon fonksiyonları herhangi bir sinir ağı mimarisinde önemli bir rol oynar. Aktivasyon fonksiyonları, verilen girdilere karşılık çıktıyı belirleyen matematiksel eşleme kurallarıdır. En basit sinir ağı mimarisi olan doğrusal algılayıcı (perceptron), biyolojik nöronun davranışını taklit ederek, nöronları belirli bir eşik değerine göre aktive eder ya da pasif bırakır. Bu süreçte aktivasyon, bir nöronun "ateşlenip ateşlenmeyeceğine" karar verilmesini ifade eder ve bu karar, modelin çıktıyı tahmin etme başarısında kritik bir rol oynar.

Yapay sinir ağlarında, her nöron gelen ağırlıklı girdilerin toplamını hesaplar ve bu değere bir yanlılık (bias) ekler. Elde edilen toplam değer, bir sonraki katmana aktarılmadan önce bir aktivasyon fonksiyonundan geçirilir. Bu adım gerçekleştirilmeden değer doğrudan aktarılırsa, sistem yalnızca girdilerle çıktılar arasında doğrusal bir ilişki kurabilir. Bu durum, sinir ağlarının karmaşık örüntüleri öğrenme kapasitesini ciddi şekilde sınırlar; çünkü doğrusal bir eşleme, çok katmanlı yapılarda bile, ağın yalnızca doğrusal ayrılabilir problemleri çözmesine neden olur.

Bu sorunu aşmak için, sinir ağlarına doğrusalsızlık (non-linearity) kazandırmak gereklidir. Aktivasyon fonksiyonları, bu doğrusal olmayanlığı sağlayarak, ağın giriş verileri ile hedef çıktı arasındaki karmaşık ilişkileri modellemesini mümkün kılar. Aynı zamanda, çıktı değerlerini belirli aralıklarda sınırlayarak (örneğin 0 ile 1 arasında ya da -1 ile 1 arasında) öğrenme sürecinin daha kararlı ve kontrollü olmasına katkıda bulunurlar. Ayrıca, modelin öğrenmeye devam edebilmesi için ağ içerisindeki ağırlıklar (weights) ve biaslar her yinelemede güncellenmelidir. Bu güncellemeler, geri yayılım algoritması aracılığıyla gerçekleştirilir. Aktivasyon fonksiyonları, bu süreçte

gradyanların hesaplanmasına ve geri yayılım sırasında katmanlar arasında iletilmesine olanak tanıyarak ağırlıkların etkin bir şekilde güncellenmesini sağlar.

İdeal bir aktivasyon fonksiyonu, yalnızca doğrusalsızlık sağlamakla kalmaz; aynı zamanda gradyanların kaybolmasını (vanishing gradient) önleyerek derin ağların daha sağlıklı öğrenmesini destekler. Bununla birlikte, uygun bir aktivasyon fonksiyonu, daha az sayıda nöronla bile daha hızlı ve daha verimli öğrenme süreci sağlayabilir. Bu sayede hem modelin eğitim süresi kısalır hem de genel performans artar.

Beyin tümörleri, her biri kendine özgü klinik sunum, tedavi yaklaşımları ve prognoza sahip 100'den fazla primer (birincil) tümör türünü kapsar. Bu tümörler, tüm onkolojik hastalıklar arasında, hastaların yaşam kalitesini en fazla etkileyenler arasında yer alır. Fiziksel, bilişsel ve psikolojik alanlarda kalıcı ve yaşamı kökten değiştiren etkiler oluşturabilirler. Beyin tümörlerinin bu çok yönlü etkileri, erken teşhis ve etkili tedavi yaklaşımlarını hayati kılmaktadır.

CNN tabanlı modeller beyin tümörlerinin tespitinde oldukça başarılı sonuçlar elde etmiştir. Bu amaçla, beyin tümörü görüntülerini sınıflandırmak için çeşitli modeller geliştirilmiştir. Bu CNN mimarilerinde en yaygın olarak kullanılan aktivasyon fonksiyonu ReLU (Rectified Linear Unit) olmuştur. Ancak, ReLU aktivasyon fonksiyonu ölü nöronlar ve bias kayması gibi bazı sorunlara sahiptir. ReLU fonksiyonu nöron girişine uygulanan negatif değerler için nöronun inaktif olmasına neden olurken Swish aktivasyon fonksiyonu nöronun çıktı üretebilmesine olanak tanır. Böylece bu değerler için gradyen ölmesi problemin ile daha az karşılaşılır. Bu durum bilgi kaybını azaltır ve modelin daha fazla ayrıntıyı öğrenmesine olanak tanır. Aynı amaçla üretilmiş olan Leakly ReLU fonksiyonu ise monoton bir yapıya sahiptir. Swish fonksiyonunun monoton olmayan yapısı ise modelin karmaşık ilişkileri daha iyi öğrenmesini sağlar.

Bu tez çalışmasında çeşitli CNN mimarilerinin farklı Swish türevi aktivasyon fonksiyonları ile çeşitli görüntü veri setleri üzerinde performansları değerlendirilmiştir. Elde edilen sonuçlar bu fonksiyonların biyomedikal görüntüler üzerinde daha başarılı sonuçlar verdiğini göstermektedir. Bu değerlendirmeler sonucunda, önerilen CNN

mimarisiyle uyumlu ve etkili çalışan aktivasyon fonksiyonlarının tespiti mümkün olmuştur.

Gerçekleştirilen tez çalışmasında dört sınıflı beyin tümörü veri seti üzerinde, ReLU, Leaky-ReLU, Swish, P-Swish ve SaRa aktivasyon fonksiyonları, VGG16, DenseNet121 ve Inception-V3 mimarileri kullanılarak test edilmiş ve performansları analiz edilmiştir.

VGG16, derinliğiyle öne çıkan ancak mimari olarak oldukça sade bir yapıya sahip evrişimli sinir ağı modelidir. DenseNet121 ise her katmanın önceki tüm katmanlardan aldığı girdilerle yoğun bağlantılar kurduğu bir mimari yaklaşımı benimser; bu yapı, bilgi akışını güçlendirirken parametre verimliliğini artırır. Inception-V3, konvolüsyon işlemlerini daha küçük boyutlara ayırarak (faktörize ederek) hesaplama yükünü azaltan ve her blok içerisinde paralel yollar (paths) barındıran Inception modüllerini kullanır.

Bu üç mimari, evrişimli sinir ağları içerisinde farklı mimari yaklaşımları temsil etmeleri nedeniyle bu çalışma kapsamında seçilmiştir. Böylece, aktivasyon fonksiyonlarının farklı ağ yapıları üzerindeki performansının kapsamlı bir biçimde değerlendirilmesi hedeflenmiştir. Ayrıca, söz konusu mimariler, tıbbi görüntü analizi literatüründe yaygın olarak kullanılan ve köklü kıyas modelleri olarak kabul edilmektedir.

Gerçekleştirilen tez çalışmasında ayrıca swish aktivasyon fonksiyonlarının CNN mimarilerinin tüm katmanlarında kullanılmasının performansı her zaman artırmadığını göstermektedir Bu nedenle çalışmada, sinir ağlarında düşük hesaplama karmaşıklığına sahip etkili bir yöntem olarak mimari yeniden yapılandırılmış ve çıkış katmanından önce üç adet yoğun (dense) katman eklenip performansları analiz edilmiştir. Böylece en yüksek başarımlı sonucu elde etmek için en uyumlu mimari-aktivasyon fonksiyonu eşleştirmesi hedeflenmiştir.

2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI

Yapay zekâ temel olarak insan gibi düşünebilen, davranabilen bilgisayar sistemleri olarak tanımlanabilir (Russel ve Norving 2003). Literatürde kabul gören genel Yapay zekâ alt başlıkları aşağıda verilmiştir. Bu temel başlıkların alt kırınımları da oldukça popüler başlıkladır.

- Bulanık Mantık
- Genetik Algoritmalar
- Doğal Dil İşleme
- Uzman Sistemler
- Örüntü Tanıma
- Makine Öğrenmesi

Bu temel başlıklardan biri olan makine öğrenmesi, deneyimle birlikte kendiliğinden gelişebilen bilgisayar programlarının nasıl oluşturulacağını inceleyen bir yapay zekâ alandır (Nilsson 1998). Pek çok kaynakta makine öğrenmesinin bir alt başlığı olarak kabul edilen Yapay Sinir Ağları (YSA) ise insan beynine özgü idrak yetisini bilgisayarlara aktarmayı amaçlayan bir yapay zekâ yöntemidir. YSA'lar insan beynindeki sinir ağlarını taklit eden bilgisayar programları olarak geliştirilmiştir (Zurada 1992).

Derin öğrenme, bilgisayarların deneyim yoluyla öğrenmesini ve dünyayı kavramlar hiyerarşisiyle anlamasını sağlayan, çok katmanlı yapay sinir ağlarını kullanan bir makine öğrenmesi yöntemidir (Goodfellow, Bengio ve Courville, 2016). Bu çalışmada kullanılan Evrişimli Sinir Ağları ise verilerden otomatik olarak özellik çıkaran bir derin öğrenme mimarisidir ve özellikle görüntü sınıflandırma problemlerinde yaygın olarak kullanılmaktadır (Anumol, 2023).

Derin öğrenme modellerinin performansını doğrudan etkileyen temel unsurlardan biri de aktivasyon fonksiyonlarıdır**.** Bu fonksiyonlar, ağın her bir nöronunun çıktısını belirleyen matematiksel işlemleri ifade eder. Özellikle doğrusal olmayan aktivasyon fonksiyonları, sinir ağlarının karmaşık veri yapılarındaki örüntüleri öğrenebilmesine olanak tanır ve modelin daha hızlı yakınsamasına katkı sağlar (Gangadia, 2021).

Aktivasyon fonksiyonlarının derin öğrenme mimarilerindeki önemine yönelik araştırmalar doğrultusunda, bu fonksiyonların farklı versiyonlarını geliştirmeye yönelik çalışmalar da literatürde yer almaktadır. Mercioni ve Holban (2020) çalışmasında, Swish aktivasyon fonksiyonundan türetilmiş olan P-Swish aktivasyon fonksiyonu önerilmektedir. P-Swish, Swish'ten farklı olarak üç adet eğitilebilir parametreye sahiptir. Bu fonksiyon, parametrelerin durumuna göre dört farklı biçimde tanımlanabilmektedir: tamamen eğitilebilir, sabit, kısmen eğitilebilir I ve kısmen eğitilebilir II. P-Swish, sürekli bir fonksiyon olup monoton değildir, alt sınırı mevcutken üst sınırı bulunmamaktadır. Bu çalışmada, P-Swish fonksiyonunun performansı; ReLU, Sigmoid, Swish ve PReLU aktivasyon fonksiyonlarıyla karşılaştırılmıştır. Karşılaştırmalar çeşitli veri setleri (CIFAR-10, CIFAR-100 ve 2018 Data Science Bowl) ve farklı mimariler (LeNet-5, Network in Network (NiN), ResNet-34 ve U-Net) üzerinde gerçekleştirilmiştir.

- LeNet-5 mimarisiyle yapılan deneylerde:
    - CIFAR-10 veri setinde en yüksek eğitim başarı oranı kısmen eğitilebilir I P-Swish,
    - CIFAR-100 veri setinde ise sabit P-Swish ile elde edilmiştir.
- NiN mimarisiyle yapılan deneylerde:
    - CIFAR-10 için en yüksek başarı Swish,
    - CIFAR-100 için ise kısmen eğitilebilir II P-Swish ile elde edilmiştir.
- ResNet-34 mimarisi ile:
    - CIFAR-10 veri setinde en iyi sonuç kısmen eğitilebilir II P-Swish,
    - CIFAR-100 veri setinde ise yine sabit P-Swish vermiştir.

Son olarak, 2018 Data Science Bowl veri seti üzerinde yapılan deneyde, U-Net mimarisi kullanılmış ve en yüksek başarı oranı P-Swish aktivasyon fonksiyonu ile elde edilmiştir. Bu sonuçlar, P-Swish'in farklı mimariler ve veri setleri üzerinde güçlü ve esnek bir aktivasyon fonksiyonu olduğunu göstermektedir.

Qureshi ve Umar (2022) çalışmasında, ReLU ve Swish aktivasyon fonksiyonlarından esinlenerek geliştirilen SaRa aktivasyon fonksiyonu tanıtılmaktadır. SaRa fonksiyonu, alt sınıra sahiptir ve bu sınır değeri, α ve β olmak üzere iki hiperparametre aracılığıyla

kontrol edilebilmektedir. Fonksiyonun üst sınırı yoktur, bu da çıkış değerlerinin pozitif doğrultuda sınırsız olabileceği anlamına gelmektedir.

Melanom sınıflandırmasına yönelik olarak iki sınıflı bir veri seti üzerinde yapılan deneylerde, Google Xception mimarisine dayanan bir model kullanılmış ve ReLU, Leaky-ReLU, ELU, Swish ve SaRa aktivasyon fonksiyonları karşılaştırılmıştır. Elde edilen sonuçlara göre, en yüksek sınıflandırma performansı SaRa aktivasyon fonksiyonu ile elde edilmiştir.

Sunkari, Sangam, Venkata Sreeram, Suchetha, Raman, Rajalakshmi ve Tamilselvi (2023) çalışmasında, ResNet-18 Evrişimli sinir ağı mimarisi tasarlanmış ve bu mimaride Swish aktivasyon fonksiyonu kullanılmıştır. Modelin performansını değerlendirmek amacıyla, dört sınıflı Diyabetik Retinopati veri seti kullanılmıştır. Tasarlanan mimarinin başarımı, farklı CNN mimarileri ile karşılaştırılmıştır. Karşılaştırmalar sonucunda, Swish fonksiyonuna sahip ResNet-18 modeli, tüm modeller arasında en yüksek sınıflandırma performansını göstermiştir. Ayrıca modelin genelleme kabiliyeti değerlendirilmek üzere farklı veri setleri üzerinde de test edilmiş ve başarılı sonuçlar elde edilmiştir.

Cheng-Fa Tsai, Jinsakul, Wu ile Chia-En Tsai (2019) çalışmasında, modifiye edilmiş bir Xception modeli önerilmektedir. Önerilen model ile orijinal Xception modeli arasındaki temel fark, ReLU aktivasyon fonksiyonu yerine Swish aktivasyon fonksiyonunun kullanılmasıdır. Çalışmada ayrıca VGG16, ResNet50, Inception-V3, Inception-ResNetV2, NASNetLarge, NASNetMobile, MobileNetV2 ve Xception modellerindeki tüm ReLU aktivasyon fonksiyonları Swish ile değiştirilmiş ve bu değişikliğin performansa etkisi değerlendirilmiştir.
Model performansları, iki sınıflı ve üç sınıflı Kolorektal Polip veri setleri kullanılarak analiz edilmiştir.

- İki sınıflı veri seti için:
    - Orijinal modeller arasında en yüksek başarı Xception modeline aittir.
    - Modifiye edilmiş modeller arasında da en iyi sonuç Swish ile değiştirilmiş Xception modeli tarafından elde edilmiştir ve bu model, orijinal Xception modelinden daha iyi performans göstermiştir.

    - Ayrıca Swish ile güncellenmiş NASNetMobile ve MobileNetV2 modelleri, orijinal sürümlerine kıyasla daha yüksek performans sergilemiştir.
    - Diğer modellerde ise Swish kullanımı anlamlı bir performans artışı sağlamamıştır.
- Üç sınıflı veri seti için:
    - Orijinal modeller arasında yine en iyi performans Xception modelinde gözlemlenmiştir.
    - Modifiye modeller arasında da Swish ile güncellenmiş Xception, en yüksek doğruluk oranını elde etmiş ve orijinalinden daha üstün bir performans sergilemiştir.
    - Bu veri seti için Swish aktivasyonu kullanılan ResNet50 ve NASNetLarge modelleri, orijinal modellerine göre daha iyi sonuçlar vermiştir.
    - Diğer modellerde ise Swish fonksiyonu performansa anlamlı bir katkı sağlamamıştır.

Javid, Das, Skoglund ve Chatterjee (2021) çalışmasında, CNN modelinin performansını artırmak amacıyla, modelin çıkış katmanından önce ReLU aktivasyon fonksiyonuna sahip bir ReDense katmanı önerilmektedir. ReDense yapısı, eğitim kaybının azaltılmasını teorik olarak garanti etmek için rastgele ağırlıklar ve ReLU aktivasyon fonksiyonlarının bir kombinasyonunu kullanmaktadır. Modelin önceki tüm katmanları dondurularak, yalnızca ReDense katmanı eğitilmekte ve bu sayede daha düşük eğitim kaybı elde edilmektedir. ReDense katmanı, sığ (shallow) bir yapıya sahip olduğundan yerel minimum veya kaybolan gradyan problemi gibi zorluklardan etkilenmemektedir.

ReDense, MNIST, CIFAR-10 ve CIFAR-100 veri kümeleri üzerinde test edilmiştir. Elde edilen sonuçlar, ReDense katmanının MLP, CNN ve ResNet-50 gibi farklı mimarilere entegre edilmesi durumunda sınıflandırma performansında artış sağladığını göstermektedir. Ayrıca bu yöntem, hem eğitim süresinde verimlilik hem de genelleme başarımında iyileşme sunmaktadır.

Gangadia (2021) çalışmasında, ilk olarak çok sığ bir CNN modeli kullanılarak CIFAR-10 ve MNIST veri kümeleri üzerinde çeşitli aktivasyon fonksiyonlarının performansı analiz edilmiştir. Yapılan analiz sonucunda, en iyi sonuçları veren aktivasyon fonksiyonlarının ReLU, SMod ve 1.7955 × Swish olduğu belirlenmiştir. Bu başarılı fonksiyonlar daha sonra VGG-16, ResNet ve EffNet gibi derin mimarilerde test edilmiştir. Ancak bu fonksiyonlar ağın tüm katmanlarında doğrudan uygulanmamıştır. Araştırma, aktivasyon fonksiyonlarının tüm katmanlarda tek başına kullanılmasının performansı her zaman artırmadığını göstermektedir. Bu nedenle, çalışmada ara katmanlarda (Int Layer) yeni aktivasyon fonksiyonlarının kullanılması, daha derin katmanlarda ise varsayılan aktivasyon fonksiyonu olan ReLU'nun tercih edilmesi önerilmektedir. Deneyler, basit fonksiyonların karmaşık fonksiyonlara kıyasla daha derin katmanlarda daha iyi sonuçlar verebildiğini göstermiştir. Ayrıca yeni aktivasyon fonksiyonları tam bağlantılı (FC) katmanlarda kullanıldığında da başarılı sonuçlar elde edilmiştir. Çalışma, çeşitli veri kümeleri üzerinde önceden eğitilmiş derin sinir ağları ile gerçekleştirilen deneyleri sunmaktadır. Ara katmanlarda (Int) ve tam bağlantılı katmanlarda (FC) ReLU, SMod ve 1.7955 × Swish fonksiyonlarının en düşük hata oranlarını sağladığı görülmüştür. Özellikle SMod fonksiyonunun istikrarlı bir şekilde yüksek performans gösterdiği gözlemlenmiştir.

Khadim, Shah ve Wagan (2021) çalışmasında, sıtma paraziti sınıflandırması amacıyla tasarlanmış bir Evrişimli Sinir Ağı modeli üzerinde çeşitli aktivasyon fonksiyonlarının performansı test edilmiştir. Karşılaştırma amacıyla Sigmoid, Tanh, ReLU, Leaky ReLU ve Swish aktivasyon fonksiyonları kullanılmıştır. Deneyler sonucunda, çoğu durumda ReLU fonksiyonunun gizli katmanlar için uygun bir seçim olduğu gözlemlenmiştir. Bununla birlikte, Swish fonksiyonunun doğruluk ve kayıp değerleri açısından ReLU'ya benzer performans sergilediği tespit edilmiştir. Model sonuçları, Swish fonksiyonunun diğer aktivasyon fonksiyonlarına kıyasla biraz daha yüksek doğruluk sağladığını ortaya koymuştur. Dolayısıyla çalışma, Swish fonksiyonunun sıtma paraziti tespiti için geliştirilen CNN modelinde gizli katman aktivasyon fonksiyonu olarak etkili bir şekilde kullanılabileceğini göstermektedir.

Sun (2021) çalışmasında, çeşitli aktivasyon fonksiyonlarının performans farklarını değerlendirmek amacıyla, ablasyon doğrulaması ile birlikte istatistiksel karşılaştırma yöntemleri kullanılmıştır. Her bir aktivasyon fonksiyonunun performansı, nicel deneysel sonuçlarla analiz edilmiştir. Deneylerde, MNIST veri seti kullanılarak görüntü sınıflandırma işlemi gerçekleştirilmiş ve model olarak klasik bir iki katmanlı Evrişimli sinir ağı olan LeNet tercih edilmiştir. Her bir aktivasyon fonksiyonunun ortalama test doğruluğu (avg acc), en yüksek test doğruluğu (top-1 acc) ve ortalama test kaybı kaydedilmiştir. Model, başlangıç öğrenme oranı 0.005 olan Adam optimizasyon algoritması ile eğitilmiş ve kayıp fonksiyonu olarak Çapraz Entropi Kaybı kullanılmıştır. Batch boyutu 32 olarak belirlenmiş ve her model, 800 iterasyon boyunca 5 kez tekrarlanarak eğitilmiştir. Deneyler süresince yalnızca aktivasyon fonksiyonu değiştirilmiş, diğer tüm hiperparametreler sabit tutulmuştur.

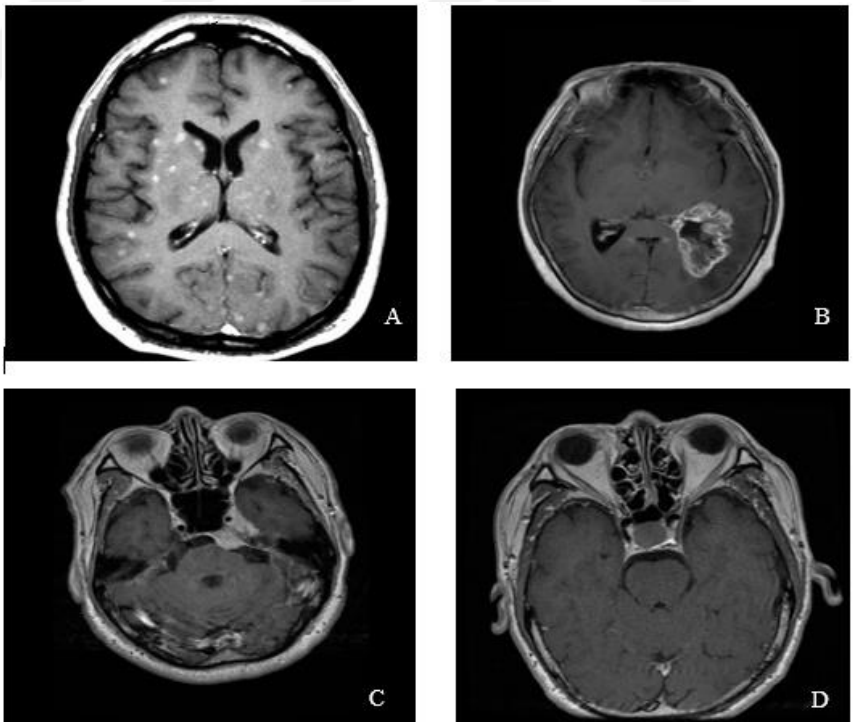
Sonuçlara göre:

- Swish fonksiyonu, en yüksek ortalama test doğruluğunu ve top-1 doğruluk oranını elde etmiştir.
- Mish, en düşük ortalama test kaybını göstermiştir.
- Sigmoid fonksiyonunun ise, görüntü sınıflandırma görevlerinde zayıf performans sergilediği gözlemlenmiştir.

3. MATERYAL ve YÖNTEM

3.1. Veri Seti

Bu çalışmada aktivasyon fonksiyonlarının performanslarını karşılaştırmak için biyomedikal bir veri seti kullanılmıştır. Veri seti, açık erişimli veri setleri sunan Kaggle platformundan alınmış ve iki uyumlu farklı veri setinin birleşiminden oluşturulmuştur. Veri seti, "Tümör Yok", "Glioma", "Meningioma" ve "Pituitary" olmak üzere dört sınıftan oluşan beyin tümörü görüntülerini içermektedir. Şekil 3.1'de her sınıftan birer örnek gösterilmektedir. "Tümör Yok" sınıfında 1202, "Glioma" sınıfında 2715, "Meningioma" sınıfında 2279 ve "Pituitary" sınıfında 2490 olmak üzere toplam 8704 resim bulunmaktadır. Veri seti eğitim, doğrulama ve test olmak üzere üçe ayrılmıştır. Veri setinin %80'i eğitim ve doğrulama için kullanılmış, kalan %20'si ise test için ayrılmıştır. Eğitim ve doğrulama verisi %90 eğitim, %10 doğrulama olacak şekilde bölünmüştür. Böylece, eğitim için 6266, doğrulama için 697 ve test için 1741 resim ayrılmıştır.



Şekil 3.1. Veri Setinin dört sınıfı A) Tümör yok B) Glioma C) Meningioma D) Pituitar

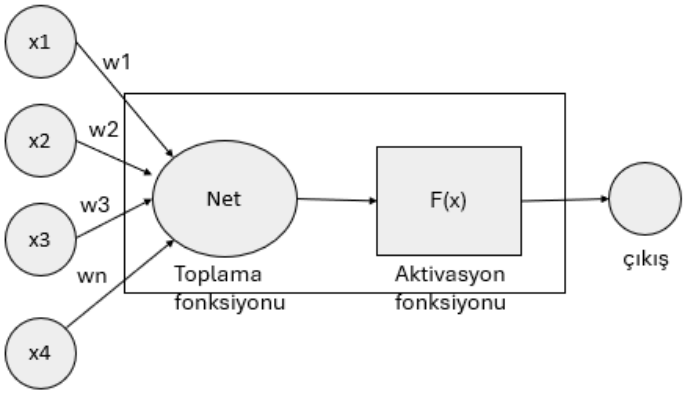
Modelin daha iyi ve kapsamlı özellik çıkarımı yapabilmesi amacıyla, ağın giriş verisi olarak renkli (RGB) görüntüler kullanılmıştır. Renkli görüntüler, gri tonlamalı

görüntülere kıyasla daha fazla bilgi içerdiğinden, modelin sınıflandırma başarısını artırmaya katkı sağlar. Ayrıca, hesaplama kaynaklarının verimli kullanımı ve eğitim süresinin makul seviyede tutulması amacıyla, görüntü boyutları 150x150 piksel olarak yeniden boyutlandırılmıştır. Böylece, modelin giriş katmanına beslenen veri boyutu 150x150x3 (yükseklik x genişlik x renk kanalları) şeklinde belirlenmiştir.

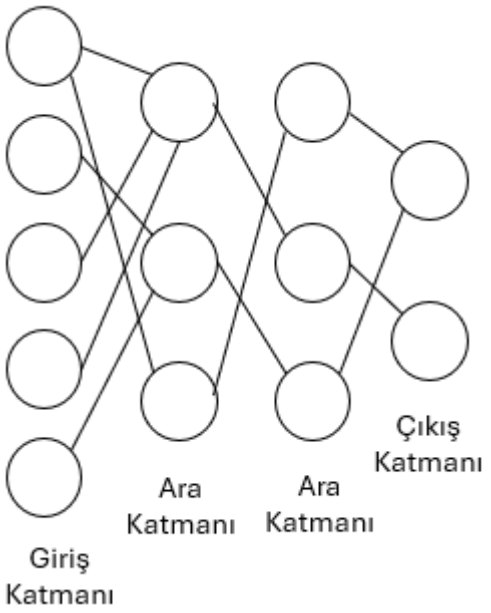
3.2. Aktivasyon fonksiyonları

Aktivasyon fonksiyonları, yapay sinir ağları ve derin öğrenme modellerinin temel bileşenlerinden biridir. Derin öğrenme modeli için uygun bir aktivasyon fonksiyonu seçmek, modelin performansını önemli ölçüde arttırabilir. Aktivasyon fonksiyonu, bir ağda verilen girdilere karşılık gelen çıktıyı belirleyen matematiksel bir işlemdir. Biyolojik nöronlara benzer şekilde, en basit doğrusal modellerde nöronlar, belirli eşik değerlerine göre aktif hale gelir. Yapay sinir ağlarında ise bir nöronun aktif olup olmayacağına aktivasyon fonksiyonu karar verir.

Şekil 3.2'de gösterildiği gibi yapay sinir ağlarında bir nöron giriş verileri ağırlıklarla çarpılıp toplanarak bir sonraki gizli katmana veya çıkış katmanına iletilir (Şekil 3.3). Eğer bu değerler katmanlar arasında doğrudan iletilseydi, bu yalnızca X'ten Y'ye doğrusal bir eşleme olurdu. Bu durumda çıkış sınırlı olmaz, $-\infty$ ile $+\infty$ arasında herhangi bir değeri alabilirdi. Ancak ağın belirli girişler için alt ve üst sınırlar oluşturabilmesi ve verilerdeki karmaşık, anlamlı desenleri yakalayabilmesi için doğrusal olmayan bir yapı gerekir. Bu doğrusal olmayanlık, aktivasyon fonksiyonu sayesinde ağa kazandırılır.



Şekil 3.2. Girişler, Toplama fonksiyonu, Aktivasyon fonksiyonu



Şekil 3.3. Yapay sinir ağnın blok diyagramı

Ayrıca, modelin sürekli öğrenebilmesi için ağırlık ve eşik parametrelerinin güncellenmesi gerekir. Aktivasyon fonksiyonları, geri yayılım (backpropagation) sürecinde bu parametrelerin güncellenmesine olanak tanır. İyi bir aktivasyon fonksiyonu, yalnızca gerekli olan az sayıdaki nöronun aktifleşmesini sağlayarak daha hızlı ve verimli bir öğrenme süreci sağlar.

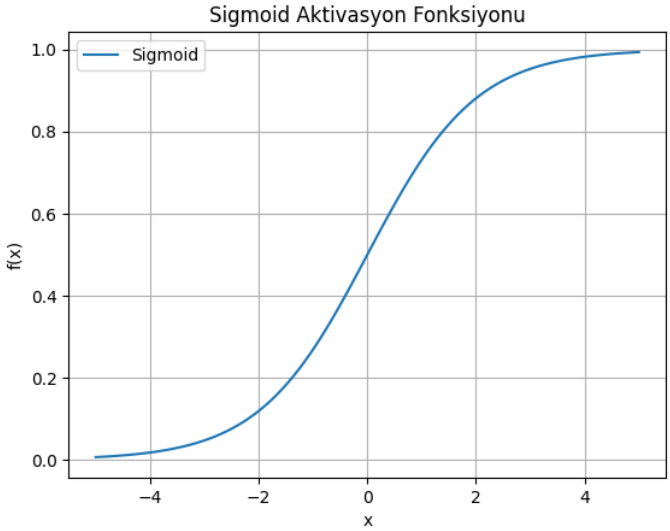
Zaman içinde birçok farklı aktivasyon fonksiyonu geliştirilmiştir. Bu bölümde, bu fonksiyonlardan bazıları ele alınacaktır.

3.2.1. Sigmoid aktivasyon fonksiyonu

Sigmoid fonksiyonu, türevlenebilir ve sürekli bir fonksiyondur; çıktısı her zaman 0 ile 1 arasında bir değer alır. Bu özelliği nedeniyle özellikle ikili sınıflandırma (binary classification) problemlerinde yaygın olarak kullanılmaktadır. Matematiksel olarak şu şekilde ifade edilir:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Fonksiyonun grafiği (Şekil 3.4) incelendiğinde, giriş değeri arttıkça fonksiyonun eğiminin azaldığı görülür. Bu durum, geri yayılım (backpropagation) sırasında gradyanların çok küçük değerlere ulaşmasına, yani gradyan sönmesi (vanishing gradient) sorununa yol açar. Gradyanların çok küçük olması, ağırlıkların güncellenmesini zorlaştırır ve modelin öğrenme hızını yavaşlatır. Bu nedenle, belli bir iterasyon sayısından sonra öğrenme durabilir veya çok yavaş ilerler.



Şekil 3.4. Sigmoid fonksiyonun matematiksel grafiği

Bir diğer dezavantajı ise sigmoid fonksiyonunun sıfır etrafında simetrik olmamasıdır. Çıkış değerleri her zaman pozitif olduğu için (0 ile 1 arasında), ağırlık güncellemeleri sırasında her zaman aynı yönde bir gradyan sinyali oluşur. Bu da modelin dengesiz öğrenmesine sebep olabilir.

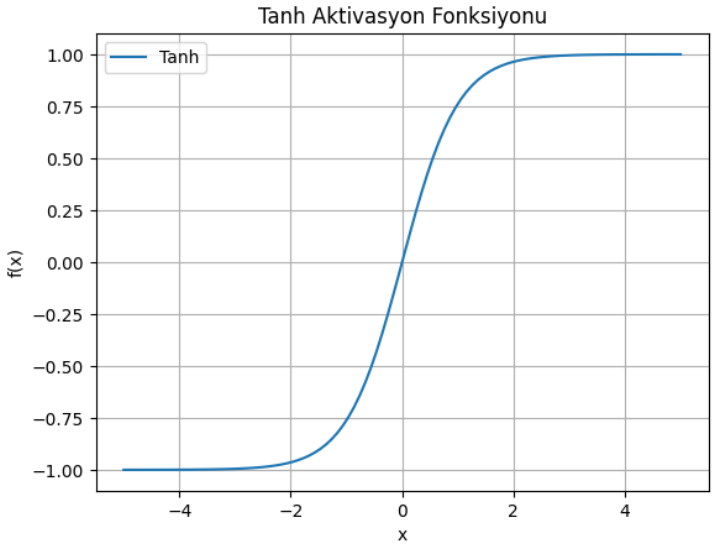
3.2.2. Tanh aktivasyon fonksiyonu

Tanh fonksiyonu, çıktısı −1 ile 1 arasında olan, sürekli ve türevlenebilir bir aktivasyon fonksiyonudur. Bu yönüyle sigmoid fonksiyonuna benzer, ancak tanh fonksiyonunun önemli bir avantajı, sıfır merkezli (*zero-centered*) olmasıdır. Matematiksel olarak şu şekilde tanımlanır:

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Fonksiyonun grafiği (Şekil 3.5) incelendiğinde, tanh fonksiyonunun orijine göre simetrik olduğu görülür. Bu özellik, çıktı değerlerinin hem pozitif hem de negatif olabilmesini sağlar; dolayısıyla ağırlık güncellemeleri sırasında gradyanlar pozitif ya da negatif yönde hareket edebilir. Bu, öğrenmeyi sigmoid fonksiyonuna kıyasla daha dengeli hâle getirir.

Ancak tanh fonksiyonu da yüksek mutlak giriş değerlerinde doygunluğa ulaşır; bu durumlarda fonksiyonun türevi çok küçük olur. Bu da yine gradyan sönmesi (vanishing gradient) problemine yol açar. Dolayısıyla, tanh fonksiyonu sigmoid'e göre bazı açılardan avantajlı olsa da, gradyan sönmesi sorunu tamamen ortadan kalkmamaktadır.



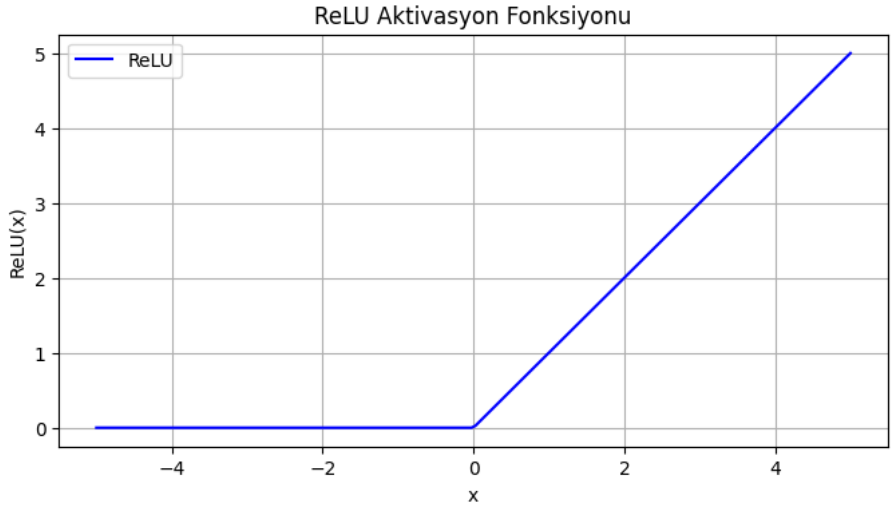
Şekil 3.5. Tanh fonksiyonun matematiksel grafiği

3.2.3. ReLU aktivasyon fonksiyonu

ReLU, günümüzde en yaygın ve en başarılı şekilde kullanılan aktivasyon fonksiyonlarından biridir. Matematiksel olarak şu şekilde tanımlanır:

$$\mathrm{ReLU} = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

Fonksiyonun grafiğini (Şekil 3.6) incelendiğinde, giriş değeri 0 veya daha küçük olduğunda çıkışın 0 olduğu, giriş 0'dan büyük olduğunda ise çıkışın giriş değerine eşit olduğu görülmektedir.

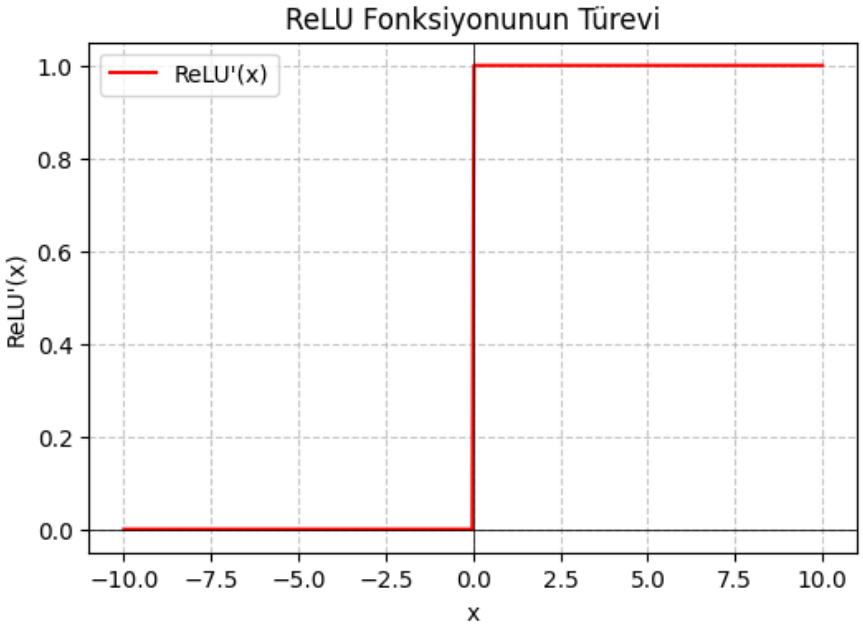


Şekil 3.6. ReLU fonksiyonun matematiksel grafiği

ReLU fonksiyonunun türevi ise şu şekildedir:

$$\mathrm{ReLU}' = \begin{cases} 1, & x > 0 \\ 0, & x < 0 \end{cases}$$

Bu türev fonksiyona ait grafik Şekil 3.7'de gösterilmektedir. Buradan da görülebileceği üzere, negatif giriş değerleri için türev sıfırdır; pozitif değerler içinse türev 1'dir.



Şekil 3.7. ReLu fonksiyonun türevinin matematiksel grafiği

ReLU fonksiyonu yalnızca 0 veya pozitif çıktılar üretir. Bu özelliği sayesinde yapay sinir ağlarında seyreklik (sparsity) oluşturur. Seyreklik, aktivasyon matrisinde çok sayıda sıfır değerin bulunması anlamına gelir. Örneğin, aktivasyonların yaklaşık %50'sinin sıfır olduğu durumda seyreklikten söz edilir. Seyrek matrisler, bellek kullanımı ve işlem maliyeti açısından daha verimlidir; böylece zaman ve alan karmaşıklığı azaltılmış olur. ReLU fonksiyonunun en önemli avantajlarından biri, tüm nöronları aynı anda aktif hale getirmemesidir. Sadece doğrusal dönüşüm sonucu pozitif olan nöronlar aktifleşir. Bu da daha verimli bir öğrenme süreci sağlar.

Tanh gibi önceki aktivasyon fonksiyonlarına göre ReLU'nun üstünlüğü, pozitif girişler için üst sınırının olmamasıdır. Bu sayede daha güçlü ve etkili modeller geliştirilebilir. Ancak ReLU fonksiyonunun bir dezavantajı da vardır: Negatif girişler için türevi sıfır olduğundan, bu bölgedeki nöronların ağırlık ve eşik değerleri geri yayılım sırasında güncellenemez. Bu durum, ölü nöronlar (dead neurons) olarak bilinen bir probleme yol açar. Ölü nöronlar, giriş değeri 0'dan küçük olduğunda sürekli olarak 0 çıktısı üretir ve artık öğrenmeye katkı sağlamaz.

3.2.4. LeakyRelu aktivasyon fonksiyonu

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0.1x, & x < 0 \end{cases}$$

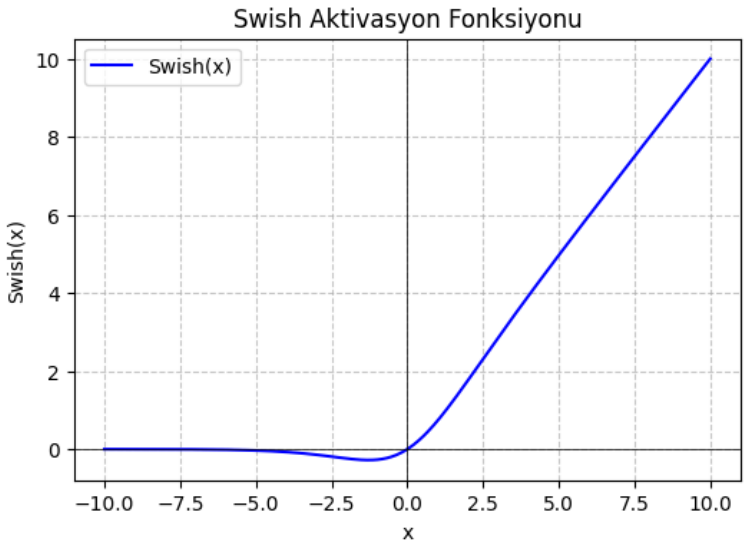
Leaky ReLU fonksiyonu, gradyan kaybı (gradient vanishing) sorununa karşı geliştirilmiştir. Ancak bu fonksiyon monoton bir şekilde artış gösterir; yani, negatif ve pozitif değerler için farklı eğimlere sahip olmasına rağmen, fonksiyon x değeri arttıkça her zaman artar.

3.2.5. Swish aktivasyon fonksiyonu

Swish'nin matimatiksel formülü

$$\text{Swish} = x \,.\, \sigma(x), \sigma(x) = \frac{1}{1+e^{-x}}$$

Swish fonksiyonunun grafiği Şekil 3.8'de gösterilmiştir. ReLU fonksiyonu gibi Swish'in de alt sınırı vardır; ancak üst sınırı yoktur. ReLU'dan farklı olarak Swish fonksiyonu düzgün (smooth) ve monoton değildir. Bu monoton olmama özelliği, Swish'i diğer aktivasyon fonksiyonlarından ayıran en önemli farklardan biridir.



Şekil 3.8. Swish fonksiyonun matematiksel grafiği

Swish fonksiyonunun türevi şu şekilde elde edilir:

$$\begin{aligned} \text{Swish}' &= \sigma(x) + x.\sigma(x).\big(1 - \sigma(x)\big) \\ &= \sigma(x) + x.\sigma(x) - x.\sigma(x)^2 \\ &= x.\sigma(x) + \sigma(x)\big(1 - x.\sigma(x)\big) \\ &= \text{Swish} + \sigma(x)(1 - \text{Swish}) \end{aligned}$$

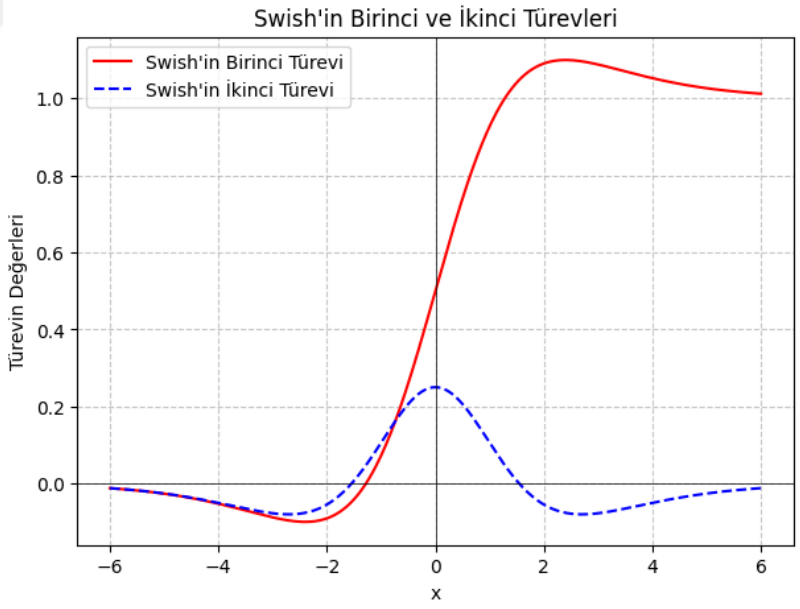
Swish fonksiyonunun birinci ve ikinci türevleri, fonksiyonun 1.25'ten küçük giriş değerleri için türev büyüklüğünün 1'den küçük olduğunu göstermektedir.

Swish fonksiyonu ayrıca parametreli biçimiyle de tanımlanabilir:

$$f(x;\ \beta) = x.\sigma(\beta.x)$$

Bu formda:

- β=0 olduğunda, fonksiyon doğrusal hale gelir: f(x)=x.
- $\beta \to \infty$ durumunda, fonksiyon ReLU'ya benzer şekilde davranır



Şekil 3.9. Swish fonksiyonun birinci ve ikinci türevlerinin matematiksel grafiği

Swish fonksiyonu, β parametresi sonsuza yaklaştığında ReLU fonksiyonu gibi davranmakta, β sıfıra eşit olduğunda ise doğrusal bir fonksiyon haline gelmektedir. Bu özelliği sayesinde Swish, doğrusal fonksiyon ile ReLU fonksiyonu arasında doğrusal olmayan bir şekilde interpolasyon yapan, sürekli ve türevlenebilir bir aktivasyon

fonksiyonu olarak düşünülebilir. Eğer β parametresi öğrenilebilir hâle getirilirse, model bu interpolasyon derecesini öğrenme süreci boyunca optimize edebilir. Bu parametreli versiyon, genellikle β-Swish olarak adlandırılır (2 çarpanı olmadan).

Swish fonksiyonunun en önemli avantajlarından biri, sınırsız (özellikle üst sınırsız) bir fonksiyon olmasıdır. Bu sayede, sigmoid veya tanh gibi fonksiyonlarda karşılaşılan doygunluk (saturation) problemi büyük ölçüde engellenmiş olur. Doygunluk, sıfıra yakın türevler nedeniyle öğrenme sürecini yavaşlatabilir. Swish bu sorunu aşmakta başarılıdır. Swish ayrıca küçük negatif girişler için negatif çıktılar üretebilme özelliğine sahiptir. Bu durum, monoton olmama özelliğinden kaynaklanır ve öğrenme sürecinde daha esnek gradyan akışı sağlar. Bu yönüyle, ReLU ve Softplus gibi sadece pozitif çıktılar üreten fonksiyonlardan ayrılır.

Swish fonksiyonu düzgün (smooth) bir yapıya sahip olduğu için optimizasyon süreçlerinde daha kararlı davranır. Aynı zamanda genelleme (modelin eğitim dışındaki verileri doğru tahmin edebilme yeteneği) açısından da önemli avantajlar sunar. Öğrenme hızını artırır, başlangıç ağırlık değerlerine duyarlılığı azaltır ve daha etkili bir eğitim süreci sağlar.

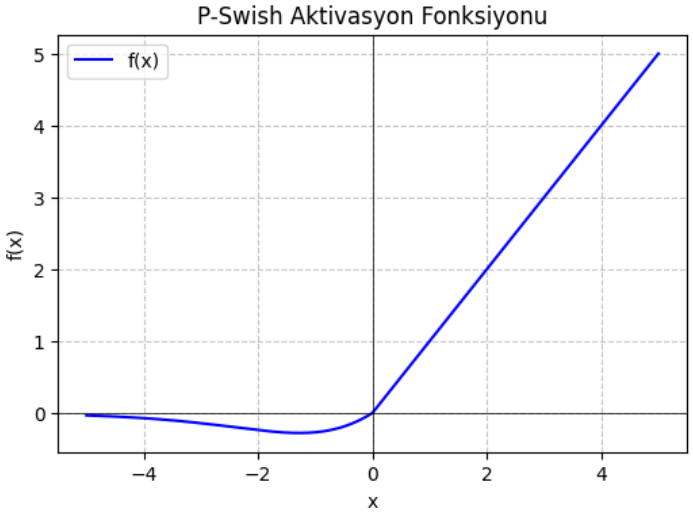
3.2.6. P-Swish aktivasyon fonksiyonu

P-Swish, matematiksel formülü şu şekildedir:

$$f(x) = \begin{cases} \alpha \cdot x \cdot \sigma(\delta \cdot x), & x \leq \beta \\ x, & x > \beta \end{cases}$$

Burada α, β ve δ parametreleri ya sabit olarak tanımlanabilir ya da eğitim sürecinde öğrenilebilir. Yaygın olarak kullanılan değerler α=0, β=0 ve δ=1 şeklindedir. Bu durumda P-Swish fonksiyonunun grafiği Şekil'de gösterildiği gibi olur.

P-Swish fonksiyonu, β≤0 durumunda klasik Swish fonksiyonu ile aynıdır. Ancak β>0 durumunda, eğitilebilir parametrelerin etkisiyle Swish'e göre daha fazla esneklik sunar. Bu esneklik, özellikle karmaşık desenleri modellemede önemli bir avantaj sağlar.



Şekil 3.10. P-Swish fonksiyonun matematiksel grafiği

P-Swish fonksiyonu şu temel özelliklere sahiptir:

- Süreklidir: Fonksiyonun her noktada kesintisiz olması, optimizasyon süreçlerinde kararlılık sağlar. (Şekil 3.10.)
- Monoton değildir: Bu özellik, modelin öğrenme yeteneğini artırabilir.
- Üst sınırı yoktur: Girdiler büyüdükçe çıkış da sınırsız şekilde artabilir.
- Alt sınıra sahiptir: Bu özellik, negatif girdiler için düzenleme (regülasyon) etkisi sağlar.

P-Swish'in temel avantajları:

- Basit bir yapıya sahip olmasına rağmen yüksek doğruluk sağlar.
- Kaybolan gradyan (vanishing gradient) problemini büyük ölçüde önler.
- Geri ve ileri yayılım süreçlerinde (backpropagation ve forward pass) etkili gradyan akışı sunar.

P-Swish fonksiyonu, parametre yapılarına göre dört şekilde sınıflandırılabilir:

1. Eğitilebilir P-Swish: α ve β eğitilebilir, δ=1 sabittir.
2. Sabit P-Swish: Tüm parametreler sabittir; α=0, β=0, δ=1.
3. Kısmen Eğitilebilir P-Swish (I): α\alphaα eğitilebilir (başlangıç değeri genellikle 1), β=0, δ=1 sabittir.
4. Kısmen Eğitilebilir P-Swish (II): α ve δ eğitilebilir (başlangıçta genellikle 1), β=0 sabittir.

Bu parametre esnekliği sayesinde P-Swish fonksiyonu, farklı görev ve veri yapıları için özelleştirilebilir bir aktivasyon fonksiyonu olarak öne çıkmaktadır.
Bu çalışmada, literatürdeki tıbbi veri seti üzerinde dört form arasında en iyi performansı gösterdiği için 2. Şekli seçilmiştir.

3.2.7. SaRa aktivasyon fonksiyonu

SaRa aktivasyon fonksiyonu, ReLU ve Swish fonksiyonlarından esinlenerek geliştirilmiş bir fonksiyondur. Matematiksel olarak şu şekilde tanımlanır:

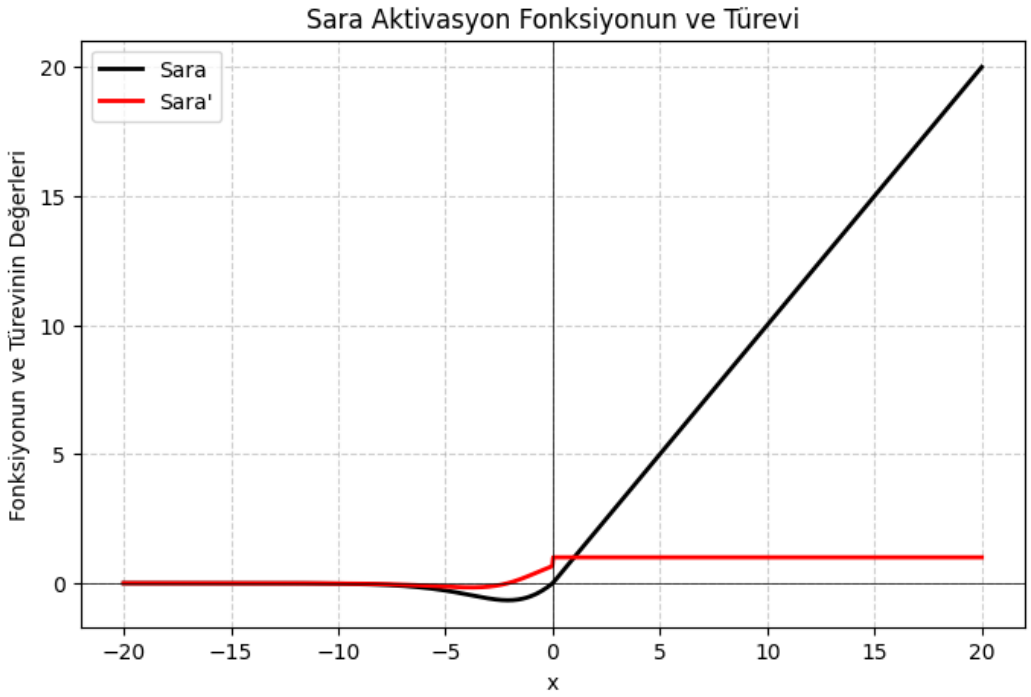
$$\text{SaRa} = \begin{cases} x, & x \geq 0 \\ f(x), & x < 0 \end{cases},$$

$$f(x) = \frac{x}{1 + \alpha \, . \, e^{-\beta . x}}$$

Burada α ve β hiperparametrelerdir. Literatürde en iyi performansın elde edildiği durum için bu parametreler α=0.5 ve β=0.7 olarak belirlenmiştir.
SaRa fonksiyonunun türevi ise şu şekilde ifade edilir:

$$f'(x) = \frac{1 + \alpha . e^{-\beta . x}(1 + \beta . x)}{(1 + \alpha . e^{-\beta . x})^2}$$

SaRa aktivasyon fonksiyonu ve türevinin grafikleri şekil 3.11'de sunulmuştur. Grafiklerden de görüleceği üzere, SaRa fonksiyonu alt sınıra sahiptir ve bu sınır değeri, α\alphaα ve β hiperparametrelerinin değiştirilmesiyle kontrol edilebilir. Bu yönüyle Swish fonksiyonundan ayrılır. Ayrıca, ReLU ve Swish gibi üst sınırı bulunmamaktadır.
SaRa, özellikle Evrişimli sinir ağı modellerinde genelleme performansı açısından başarılı sonuçlar vermektedir. Swish fonksiyonunda, ağırlık değerlerinin çoğu sıfır olduğunda gradyan sıfırlanması riski bulunur; çünkü Swish de alt sınıra sahip bir fonksiyondur. Ancak SaRa'da, hiperparametreler sayesinde negatif değerli girişlerin etkisi ayarlanabilir. Böylece gradyan sıfırlanması riski azaltılmış olur ve daha kararlı bir öğrenme süreci sağlanabilir.



Şekil 3.11. SaRa fonksiyonun matematiksel grafiği

3.2.8. Softmax aktivasyon fonksiyonu

Softmax, çoklu çıkışa sahip yapay sinir ağı modellerinin çıktı katmanında yaygın olarak kullanılan bir aktivasyon fonksiyonudur. Softargmax veya normalize edilmiş üstel fonksiyon olarak da adlandırılmaktadır. Bu fonksiyonun temel amacı, modelin çıktı vektöründeki her bir değeri olasılık dağılımına dönüştürmektir. Yani, Softmax fonksiyonu bir sınıflandırma probleminde her bir sınıfa ait olasılıkları hesaplayarak toplamlarının 1 olmasını sağlar. Bu yönüyle sigmoid fonksiyonunun çok sınıflı problemler için bir genellemesi olarak da değerlendirilebilir.
Softmax fonksiyonunun matematiksel ifadesi şu şekildedir:

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \text{ , i: Çıkış Sayısı}$$

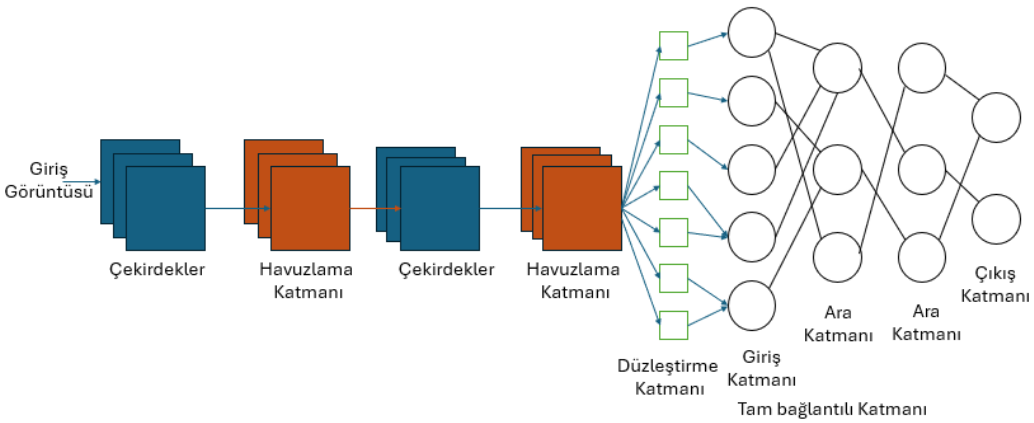
burada:

- x_i. i.çıkış nöronuna ait giriş değeri,
- n: Çıkış nöronu sayısıdır.

Softmax aktivasyon fonksiyonu, tüm çıktı birimlerinin olasılıklarını hesaplar ve bu çıktıların toplamının 1 olmasını sağlar. Bu sayede model, tahmin edilen sınıfa ilişkin en yüksek olasılık değerine sahip olan nöronu aktive eder. Bu özellik, çok sınıflı sınıflandırma problemlerinde son derece yararlı bir yapı sunar.

3.3. Evrişimli Sinir Ağı

Sınıflandırma işleminden önce, sinir ağına giriş olarak verilen görüntülerin boyutsallığını azaltmak ve anlamlı özellikleri etkin şekilde çıkarmak amacıyla, CNN kullanılan temel işlemler uygulanmıştır. Yüksek boyutlu görüntüler doğrudan sınıflandırma modellerine verildiğinde hem hesaplama maliyetini artırmakta hem de eğitim süresini uzatmaktadır. Bu nedenle, evrişimli katmanların sunduğu avantajlardan faydalanılarak, giriş görüntülerinden sınıflandırma için gerekli olan temel özellikler seçilmiştir. Özellik çıkarımı sürecinde (Şekil 3.12), filtreler (çekirdekler), adımlama (stride), dolgu (padding), havuzlama (pooling) ve düzleştirme (flattening) gibi CNN yapı taşları kullanılmış; ardından bu öznitelikler, sınıflandırma işlemi için tam bağlantılı (fully connected) sinir ağı katmanlarına aktarılmıştır. Şu ana kadar birçok farklı CNN mimarisi geliştirilmiştir. Bu mimariler, tasarım prensipleri, mimari bileşenleri, çeşitli bilgisayarla görme görevlerine uygunlukları, başarıyla uygulandıkları alanlar ve sağladıkları avantajlar açısından eleştirel olarak değerlendirilmektedir. Aynı zamanda, her mimarinin sınırlamaları da göz önünde bulundurulmaktadır. Bu bölümde, literatürde öne çıkan bazı CNN mimarilerine yer verilmektedir.



Şekil 3.12. CNN blok diyagramı (Zingaretti et al.,2020)

3.3.1. VGG16 mimari

VGG16, küçük boyutlu konvolüsyon filtrelerinden oluşan derin bir mimaridir. Bu mimaride kullanılan konvolüsyon filtreleri genellikle 3×3 boyutundadır; bu, sağ/sol, üst/alt ve merkez bilgilerini yakalayabilmek için tercih edilen en küçük filtre boyutudur. Ayrıca, giriş bilgileri üzerinde doğrusal bir geçiş sağlamak amacıyla 1×1 boyutunda konvolüsyon filtreleri de kullanılabilir. Konvolüsyon filtrelerinin adım (stride) büyüklüğü genellikle 1 pikseldir. 3×3 boyutundaki filtrelerin görüntü boyutlarını küçültmesini önlemek için 1 piksel dolgu (padding) uygulanmaktadır. Belirli konvolüsyon katmanlarından sonra beş adet havuzlama katmanı kullanılmaktadır (Şekil 3.13). Bu havuzlama filtreleri 2×2 boyutunda olup, adım büyüklükleri de 2 pikseldir. 13 konvolüsyon katmanını takiben, 3 adet tam bağlantılı (fully connected) katman bulunmaktadır. Çıkış katmanı ise Softmax katmanıdır. Mimarideki tüm gizli katmanlarda ReLU aktivasyon fonksiyonu kullanılmaktadır.



Şekil 3.13. VGG16 mimarisinin blok diyagramı (Prasanth., 2024)

3.3.2. DenseNet mimari

DenseNet mimarisi, giriş ve çıkış katmanlarına yakın olan katmanlar arasında kısa bağlantılar kurularak daha derin, daha başarılı ve eğitim sürecinde daha etkili olacak şekilde tasarlanmıştır. DenseNet'te her katman, ağdaki tüm önceki katmanlarla doğrudan bağlantılıdır. Geleneksel evrişimsel sinir ağlarında, L katmanlı bir yapıda sadece L bağlantı bulunurken (her katman yalnızca bir önceki katmanla bağlantılıdır), DenseNet mimarisinde bu sayı $\frac{L(L+1)}{2}$ olmaktadır.

Her bir katman, önceki tüm katmanlardan gelen özellik haritalarını giriş olarak alır ve ürettiği özellik haritasını da sonraki tüm katmanlara aktarır. Bu yapı sayesinde, DenseNet daha az parametreye sahip olur çünkü gereksiz özellik haritalarının tekrar öğrenilmesine gerek kalmaz.

DenseNet'in en önemli avantajlarından biri, bilgi ve gradyan akışını önemli ölçüde iyileştirmesidir; bu da modelin öğrenmesini kolaylaştırır. Ayrıca her katmanın hem orijinal girişe hem de kayıp fonksiyonuna doğrudan erişimi olduğundan, derin öğrenme süreci daha verimli hale gelir. DenseNet aynı zamanda düzenleştirici (regularization) bir etkiye de sahiptir ve bu sayede küçük veri setleriyle çalışırken aşırı öğrenmenin (overfitting) önüne geçebilir.

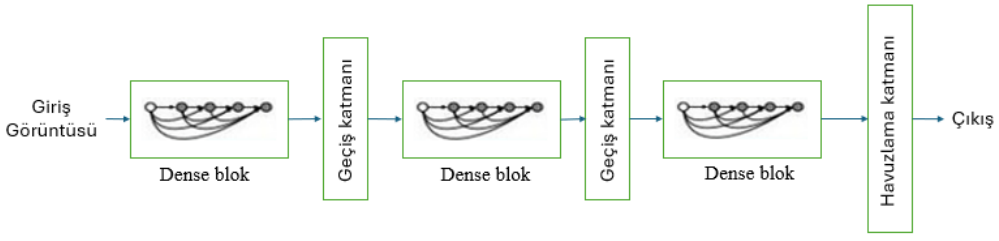
Her bir katmanın matematiksel gösterimi şu şekildedir:

$$x_i = H_i([x_0, x_1, \ldots, x_{i-1}])$$

Burada:

- x_i: i. katmanın çıktı özellik haritasıdır.
- H_i (.): toplu normalizasyon (batch normalization), ReLU aktivasyon fonksiyonu ve 3×3 konvolüsyon katmanından oluşan bir bileşik fonksiyondur.

Ancak H_i(.) fonksiyonu, farklı boyutlardaki özellik haritaları üzerinde doğrudan çalışamaz. Bu sorunu çözmek için ağ, çoklu Dense bloklara bölünür (Şekil 3.14). Bloklar arasında geçiş katmanları (transition layers) yer alır. Geçiş katmanları; toplu normalizasyon, 1×1 konvolüsyon ve ardından 2×2 boyutunda ortalama havuzlama (average pooling) katmanlarından oluşur.



Şekil 3.14. DanseNet mimarisinin blok diyagramı (Huang et al.,2018)

Her katman, k adet yeni özellik haritası üretir ve bu k değeri modelin hiperparametresi olup "büyüme oranı" (growth rate) olarak adlandırılır. Eğer her Hi fonksiyonu k adet özellik haritası üretiyorsa, i. katmanın giriş kanal sayısı şu şekilde olur:

$$K_0 + K.(İ - 1)$$

Burada K_0, giriş katmanındaki kanal sayısını ifade eder.

DenseNet'in en ayırt edici özelliklerinden biri, derinliği az tutarak yüksek performans sağlamasıdır. Örneğin, büyüme oranı olarak k=12 seçilebilir. Verimliliği artırmak amacıyla, 3×3 konvolüsyon katmanından önce darboğaz (bottleneck) olarak 1×1 konvolüsyon katmanları da eklenebilir. Aynı şekilde geçiş katmanlarında da özellik haritalarının sayısı azaltılabilir.

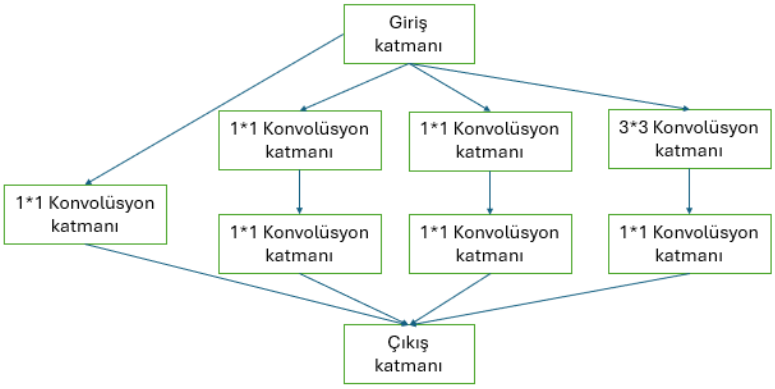
DenseNet121 yapısı şu katmanlardan oluşur:

- İlk katmanlar: 7×7 konvolüsyon katmanı ve 3×3 maksimum havuzlama (max pooling) katmanı.
- Ardından gelen 4 adet Dense blok:
    1. blok: Her biri 1×1 ve 3×3 filtrelerden oluşan 6 katman
    2. blok: Her biri 1×1 ve 3×3 filtrelerden 12 katman
    3. blok: Her biri 1×1 ve 3×3 filtrelerden 24 katman
    4. blok: Her biri 1×1 ve 3×3 filtrelerden16 katman
- Her Dense blok arasında, 1×1 konvolüsyon ve 2×2 ortalama havuzlama içeren geçiş katmanları yer alır.
- Tam bağlantılı (fully connected) katman yerine, 7×7 global ortalama havuzlama katmanı yer alır.
- Son katman: Softmax aktivasyon fonksiyonuna sahip çıkış katmanıdır.
- Tüm konvolüsyon katmanlarında batch normalization ve ReLU aktivasyon fonksiyonu kullanılmaktadır.

3.3.3. Evrişimli nöral ağ modeli V3 mimari

Evrişimli Sinir Ağı Modeli V3 (Inception-V3), temel olarak GoogleNet ve Evrişimli nöral ağ modeli V2 üzerine inşa edilmiştir. Bu modelin temelini oluşturan "inception" (nöral) yapısı (Şekil 3.15), ilk kez 2014 yılında GoogleNet mimarisinde kullanılmıştır.

Bu yapı, görüntünün küçük bölümlerini işlemek için en etkili yöntemleri uygularken, ağın genel performansını da yüksek tutmayı amaçlamaktadır.



Şekil 3.15. Nöral Yapısı (Szegedy et al.,2014)

İnception yapısının temel özelliği, tek bir boyutta filtre (örneğin 3×3) kullanmak yerine, farklı boyutlardaki filtrelerin paralel olarak kullanılmasıdır. Örneğin, 1×1 filtreler küçük detayları yakalarken, 5×5 filtreler daha büyük desenleri tanımlayabilir. Ancak, büyük filtrelerin doğrudan kullanımı yüksek hesaplama maliyetine yol açtığından, önce 1×1 filtreler uygulanır. Bu filtreler, hesaplama yükünü azaltırken önemli özellikleri korur, böylece ağın verimliliğini artırır.

İnception yapısı aynı zamanda havuzlama (pooling) katmanları da içerir. Bu katmanlar, görüntü boyutlarını küçülterek önemli bilgileri öne çıkarır ve gereksiz verilerin etkisini azaltır. Yapının çıkış katmanı ise, tüm bu paralel yapıların çıktısını birleştirerek sonraki katmana zengin bir özellik haritası iletir.

CNN Modeli V2 de inception yapısını içermektedir; ancak yapının tasarımı farklıdır. V2 modelinde, 5×5 filtreler yerine hesaplama maliyetini azaltmak amacıyla iki adet 3×3 filtre kullanılmaktadır. Ayrıca, V2 modelinde simetrik N×N filtreler yerine 1×N ve N×1 filtrelerin kullanılması, hesaplamaları daha verimli hale getirmektedir. Bu yaklaşım, özellikle büyük filtreler yerine daha küçük filtre kombinasyonlarıyla benzer etkiyi daha düşük maliyetle sağlamaktadır.

CNN Modeli V2'de ayrıca toplu normalizasyon (batch normalization) kullanılmıştır. Bu yöntem, modelin eğitim sürecini hızlandırmakta ve eğitimin daha kararlı bir şekilde ilerlemesini sağlamaktadır.

CNN Modeli V3'te bu iyileştirmelere ek olarak, küçük filtrelerde de simetrik N×N filtreler yerine 1×N ve N×1 filtreler tercih edilmektedir. Örneğin, 3×3 filtre yerine 1×3 ve 3×1 filtreler kullanılmaktadır. Ayrıca, alt örnekleme (downsampling) işlemi için maksimum havuzlama (max pooling) yerine konvolüsyon katmanları kullanılmaktadır. V3 mimarisi ayrıca etiket düzeltme (label smoothing) tekniğini de içermektedir; bu teknik, modelin ezberleme eğilimini azaltarak genelleme yeteneğini artırmaktadır.
Son olarak, CNN Modeli V3'te batch normalization, aktivasyon fonksiyonundan **önce** uygulanmaktadır. Bu yapılandırma, modelin başarı oranını ve kararlılığını daha da iyileştirmektedir.

3.4. Yöntem

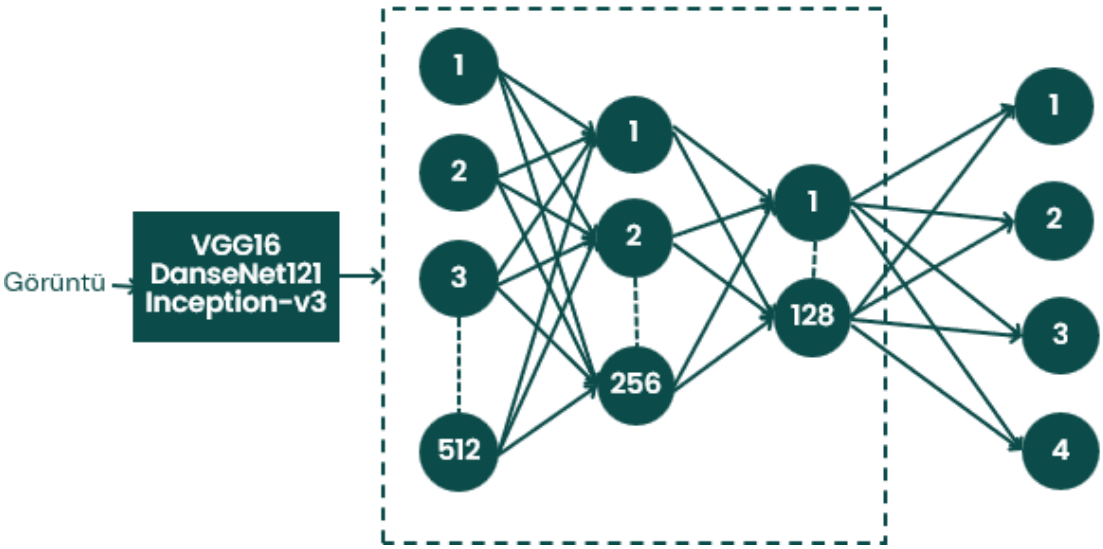
Aşırı uyumu engellemek ve bir sinir ağının performansını artırmak için genellikle üç standart yaklaşım tercih edilmektedir: aşırı uyumu engelleme, hiperparametre optimizasyonu ve veri artırma. Aşırı uyum, çoğunlukla L_1 veya L_2 düzenlemesi, bırakma (dropout) ve erken durdurma (early stopping) teknikleriyle ele alınır. Hiperparametre optimize etme ise çapraz doğrulama (cross validation) yöntemiyle gerçekleştirilir ve en uygun nöron sayısı, katman derinliği, öğrenme oranı gibi ağ mimarisi unsurlarının belirlenmesinin yanı sıra farklı optimize edicilerin ve kayıp fonksiyonlarının denenmesini içerir.

Veri artırma yöntemi ise eğitim veri setini yapay olarak çoğaltmaya amaçlar; bu, rastgele kırpma ve döndürme, yeniden ölçekleme ve örneklerin karıştırılması (mixup) gibi işlemlerle gerçekleştirilir. Ancak, yukarıda belirtilen tüm teknikler, teorik olarak performans artışını garanti etmemekle birlikte, genellikle ağın sıfırdan yeniden eğitilmesini gerektirir.

Araştırmaya göre aktivasyon fonksiyonlarının tüm katmanlarda tek başına kullanılmasının performansı her zaman artırmadığını göstermektedir. Bu nedenle,

çalışmamızda sinir ağlarında düşük hesaplama karmaşıklığına sahip etkili bir yöntem olarak, mimari yeniden yapılandırılmış ve çıkış katmanından önce üç adet yoğun (dense) katman eklenmiştir (Şekil 3.16). Üç katman tercih edilmesinin temel nedeni, yapının yapay sinir ağı (ANN) karakteristiğini yansıtmasının yanı sıra, içerdiği hiperparametre sayısını artırarak modelin öğrenme kapasitesini geliştirmektir. Her bir katmandaki nöron sayısı sırasıyla 512, 256 ve 128 olarak belirlenmiştir. Bu tercih, çok sayıda nöronun neden olabileceği hesaplama karmaşıklığından kaçınmak ve aktivasyon fonksiyonlarının performanslarını daha sağlıklı gözlemleyebilmek amacıyla yapılmıştır. Her bir denemede, tüm katmanlardaki aktivasyon fonksiyonları sistematik bir şekilde değiştirilerek farklı konfigürasyonların performansları karşılaştırılmıştır.

Sınıf sayısının dört olması nedeniyle, ağın çıkış katmanında Softmax aktivasyon fonksiyonu kullanılmıştır. Aşırı öğrenmeyi önlemek amacıyla, her katmandan sonra %30 oranında bırakma (dropout) yöntemi uygulanmış ve ayrıca 0.00001 oranı ile L2 normu ile düzenlileştirme (regularization) tekniği kullanılmıştır. Dropout oranı olarak 0.3 seçilmesinin nedeni; daha düşük oranların modelin öğrenme sürecini yavaşlatabilmesi, daha yüksek oranların ise çok sayıda nöronun ihmal edilmesine sebep olarak aktivasyon fonksiyonlarının performansının sağlıklı bir şekilde analiz edilmesini engelleyebileceği düşünülmesidir.



Şekil 3.16. Model ile eklenen dense katmanları ve nöron sayıları

VGG16, derinliğiyle öne çıkan ancak mimari olarak oldukça sade bir yapıya sahip bir evrişimli sinir ağı modelidir. DenseNet121 ise her katmanın önceki tüm katmanlardan aldığı girdilerle yoğun bağlantılar kurduğu bir mimari yaklaşımı benimser; bu yapı, bilgi akışını güçlendirirken parametre verimliliğini artırır. Inception-V3, konvolüsyon işlemlerini daha küçük boyutlara ayırarak (faktörize ederek) hesaplama yükünü azaltan ve her blok içerisinde paralel yollar (paths) barındıran Inception modüllerini kullanır.

Bu üç mimari, evrişimli sinir ağları içerisinde farklı mimari yaklaşımları temsil etmeleri nedeniyle bu çalışma kapsamında seçilmiştir. Böylece, aktivasyon fonksiyonlarının farklı ağ yapıları üzerindeki performansının kapsamlı bir biçimde değerlendirilmesi hedeflenmiştir. Ayrıca, söz konusu mimariler, tıbbi görüntü analizi literatüründe yaygın olarak kullanılan ve köklü kıyas modelleri olarak kabul edilmektedir.

Swish aktivasyon fonksiyonunun performansını daha sağlıklı değerlendirebilmek amacıyla, ReLU ve Leaky ReLU fonksiyonları ile karşılaştırmalar yapılmıştır. Literatürde sigmoid ve tanh gibi klasik aktivasyon fonksiyonlarının, daha sonra geliştirilen modern fonksiyonlara kıyasla zayıf performans sergilediği gözlemlendiğinden bu çalışmada değerlendirmeye alınmamıştır. Bu doğrultuda, eklenen üç yoğun (dense) katmanda performansı karşılaştırılmak üzere kullanılan aktivasyon fonksiyonları şunlardır: ReLU, Leaky ReLU, Swish, Parametric Swish (P-Swish) ve SaRa.

Modelde aşırı uyumun olup olmadığını değerlendirmek amacıyla iterasyon (epoch) sayısı 100 olarak belirlenmiştir. Hem aşırı uyumu engellemek hem de modelin genelleme yeteneğini artırmak için, ayrıca eğitimin çok yavaşlamaması adına batch size değeri 16 olarak seçilmiştir. Modelin kararlılığını sağlamak amacıyla özel bir öğrenme oranı zamanlama stratejisi uygulanmıştır. Bu strateji, üstel azalma (exponential decay) ile ısınma (warm-up) yöntemlerinin birleşiminden oluşmaktadır. Fonksiyonel olarak aşağıdaki şekilde tanımlanabilir:

$$x_t = \begin{cases} k \cdot \frac{t}{T}, & t < T \\ k \cdot e^{-a(t-T)}, & t \geq T \end{cases}$$

Burada:

- k: başlangıç öğrenme oranı,
- T: warm-up iterasyon sayısı,
- a: üstel azalma katsayısıdır.

Çalışmada k, T ve a sırasıyla 0.00001, 5 ve 0.1 olarak belirlenmiştir. Veri setinin küçük olması nedeniyle modelin kararlılığını artırmak için optimizer olarak Adam (Adaptive Moment Estimation) (öğrenme hızı: 0.01, ilk Moment: 0.9, ikinci moment: 0.999, ε = 1e-08) kullanılmıştır. Son olarak, çoklu sınıflandırma problemleri için uygun olduğundan, kayıp fonksiyonu olarak Kategorik Çapraz Entropi (Categorical Crossentropy Loss) tercih edilmiştir.

3.5. Değerlendirme Metrikleri

3.5.1. Model Performansının Değerlendirilmesi

Eğitim doğruluğu, modelin eğitim veri kümesi üzerindeki öğrenme başarısını ölçer. Yüksek eğitim doğruluğu, modelin eğitildiği verileri iyi öğrendiğini gösterir. Ancak bu değer, doğrulama veya test doğruluğundan belirgin şekilde yüksekse, bu durum aşırı uyum göstergesi olabilir.

Doğrulama doğruluğu, eğitim sırasında modele sunulmayan doğrulama veri kümesi üzerinde hesaplanır. Modelin genelleme yeteneğini ve hiperparametre ayarlarının başarımını değerlendirmede önemli bir ölçüttür. Eğitim ve doğrulama doğruluğu arasındaki farkın artması, modelin yalnızca eğitim verilerine uyum sağladığını ve yeni verilere genellemede yetersiz kaldığını gösterebilir. Doğrulama oranı şu şekilde hesaplanmaktadır:

$$\text{Doğrulama Oranı} = \frac{\text{Doğru Tahmin Sayısı}}{\text{Toplam Tahmin Sayısı}}$$

Test doğruluğu ise, eğitim sürecinde kullanılmayan tamamen bağımsız test veri kümesi üzerinde ölçülür. Modelin gerçek dünya verileri üzerindeki genelleme kabiliyetini

yansıtır. Doğrulama doğruluğuna yakın ve yüksek bir test doğruluğu, modelin başarılı bir şekilde genelleme yaptığını gösterir.

Eğitim kaybı, modelin tahminleri ile gerçek değerler arasındaki farkı (hata) ölçer. Bu değerin azalması, modelin öğrenme sürecinde doğru yönde ilerlediğine işaret eder. Ancak, çok düşük bir eğitim kaybına rağmen doğrulama performansı düşükse, modelin aşırı uyum gösterdiği düşünülebilir. Kayıp oranı şu şekilde hesaplanmaktadır:

$$\text{Kayıp Oranı} = \frac{\text{Yanlış Tahmin Sayısı}}{\text{Toplam Tahmin Sayısı}}$$

Doğrulama kaybı, doğrulama veri kümesi üzerindeki hata oranını ifade eder ve modelin genelleme başarımını değerlendirmede kullanılır. Eğitim kaybına yakın ve zamanla azalan doğrulama kaybı, modelin dengeli bir şekilde öğrendiğini gösterir. Aksine, doğrulama kaybının artması veya eğitim kaybıyla arasındaki farkın büyümesi, aşırı uyumu işaret edebilir.

Test kaybı, modelin son haliyle test veri kümesi üzerindeki hata oranını gösterir. Düşük test kaybı ve buna eşlik eden yüksek test doğruluğu, modelin hem öğrenme hem de genelleme açısından başarılı olduğunu gösterir.

3.5.2. Grafiksel Analiz

Doğruluk grafiği, eğitim süreci boyunca modelin eğitim ve doğrulama doğruluk değerlerindeki değişimi görsel olarak sunar. Bu grafik, modelin yakınsama davranışını, öğrenme eğilimini ve olası aşırı uyum belirtilerini (örneğin, eğitim doğruluğu artarken doğrulama doğruluğunun düşmesi) ortaya koyar.

Kayıp grafiği ise, eğitim ve doğrulama kayıplarının zaman içindeki değişimini gösterir. Başarılı bir modelde her iki kayıp da giderek azalır ve belirli bir noktada sabitlenir. Eğitim ve doğrulama kayıpları arasındaki sapmanın artması ise modelin aşırı uyum yaptığını veya yeterince genelleme sağlayamadığını gösterebilir.

3.5.3. Kullanılan Metrikler

Bu çalışmada, aktivasyon fonksiyonlarının performansını değerlendirmek amacıyla aşağıdaki metrikler kullanılmıştır:

- Eğitim doğruluk oranı
- Eğitim kayıp oranı
- Doğrulama doğruluk oranı
- Doğrulama kayıp oranı
- Test doğruluk oranı
- Test kayıp oranı

Ayrıca modelin öğrenme sürecindeki kararlılığını değerlendirebilmek için eğitim sürecine ait iterasyonlar boyunca elde edilen doğruluk ve kayıp değerleri grafiksel olarak görselleştirilmiş, bu grafikler üzerinden analizler gerçekleştirilmiştir.

4. BULGULAR

4.1. VGG16 Modelin Bulguları ve Tartışması

Standart VGG16 modeli (VGG16 modelin dense katmanları eklemeden) ile yapılan sınıflandırma işleminde değerlendirme metriklerine göre elde edilen sonuçlar aşağıda verilmiştir:

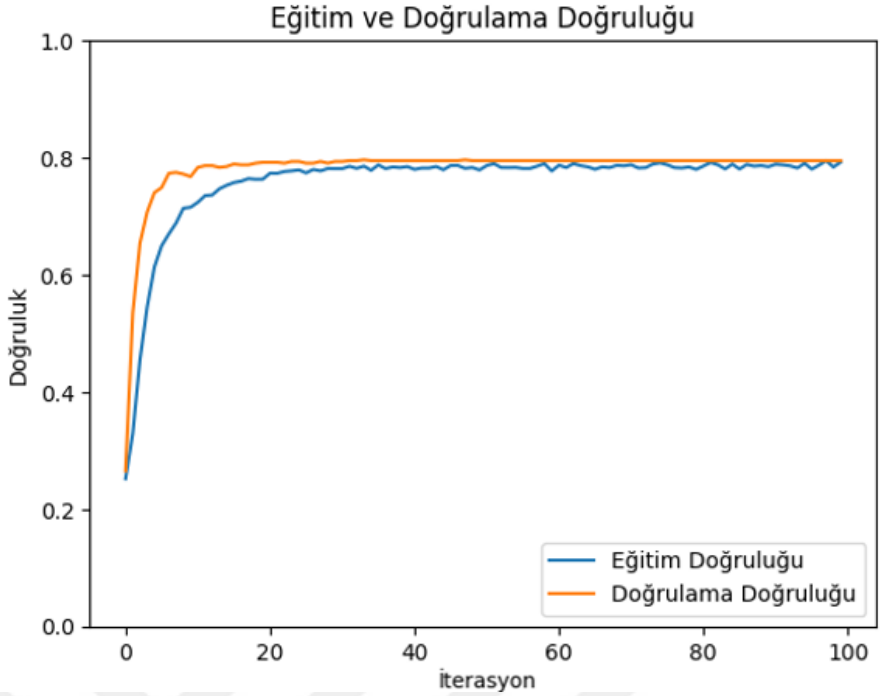
- Eğitim doğruluk oranı: 0.7946
- Eğitim kayıp oranı: 0.5854
- Doğrulama doğruluk oranı: 0.7948
- Doğrulama kayıp oranı: 0.5621
- Test doğruluk oranı: 0.8102
- Test kayıp oranı: 0.5545

Elde edilen sonuçlara göre, modelin doğruluk oranları oldukça düşük seviyelerde kalırken, kayıp oranları ise yüksek çıkmıştır. Ayrıca, dikkat çeken bir diğer durum, doğrulama ve test doğruluk oranlarının, eğitim doğruluk oranlarından daha yüksek olmasıdır. Bu durum, tüm iterasyonlar boyunca çizilen doğruluk grafiğinde (Şekil 4.1) açıkça gözlemlenmektedir.

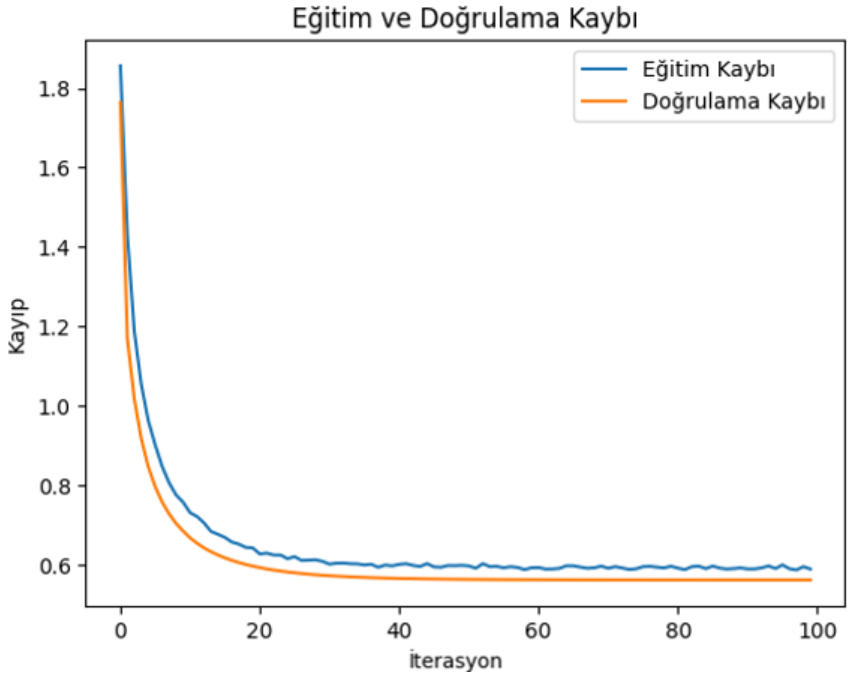
Benzer şekilde, doğrulama ve test kayıp oranlarının eğitim kayıp oranlarından daha düşük olduğu görülmektedir. Bu durum da iterasyonlar boyunca çizilen kayıp grafiğinde (Şekil 4.2) açıkça ortaya konmuştur.

Bu bulgular, modelin hem eğitim verileri üzerinde yeterince öğrenemediğini hem de genelleme yeteneğinin zayıf olduğunu göstermektedir. Eğitim doğruluğunun düşük kalması, modelin öğrenme sürecinde başarısız olduğunu; doğrulama ve test sonuçlarındaki tutarsızlıklar ise modelin kararsız bir yapı sergilediğini ortaya koymaktadır.

Tüm bu değerlendirmeler, modelin genel anlamda zayıf bir performans sergilediğini ve pratik uygulamalarda kullanılabilirlik açısından yetersiz olduğunu göstermektedir.



Şekil 4.1. VGG16 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.2. VGG16 modelin eğitim ve doğrulama kayıp grafiği

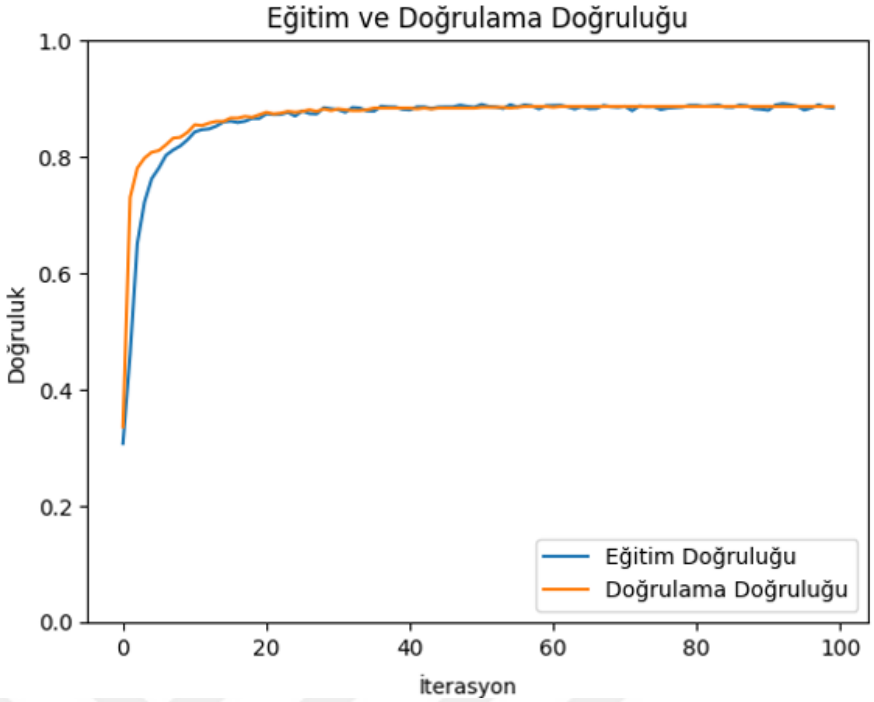
VGG16 modeline ReLU aktivasyon fonksiyonu ile üç dense katmanı ekledikten sonra, elde edilen değerlendirme metriklerinin oranları:

- Eğitim doğruluk oranı: 0.8845
- Eğitim kayıp oranı: 0.3293
- Doğrulama doğruluk oranı: 0.8867
- Doğrulama kayıp oranı: 0.2988
- Test doğruluk oranı: 0.8823
- Test kayıp oranı: 0.3022

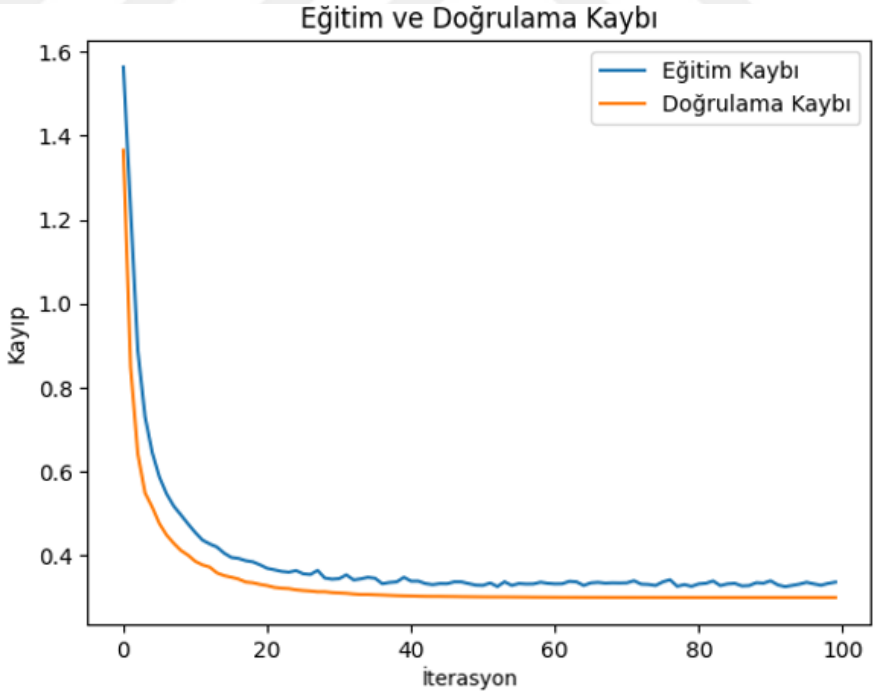
Elde edilen bulgulara göre, ReLU aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir. Ancak bu iyileşmelere rağmen, elde edilen doğruluk ve kayıp değerleri modelin başarılı veya kullanılabilir bir yapı olduğunu göstermek için yeterli seviyede değildir.

Modelin doğruluk grafiği (Şekil 4.3) incelendiğinde, belirli bir iyileşme sağlandığı tespit edilmiştir. Bununla birlikte, kayıp grafiği (Şekil 4.4), doğrulama ve test kayıp oranlarının eğitim kayıp oranlarına kıyasla daha düşük seviyelerde seyrettiğini göstermektedir. Bu durum, modelin eğitim verisine kıyasla doğrulama ve test verileri üzerinde daha iyi performans sergilediği izlenimini oluştursa da, genel başarı seviyeleri dikkate alındığında modelin istikrarlı ve güvenilir bir yapı sergilemediği sonucuna varılmıştır.

VGG16 mimarisi üzerine ReLU aktivasyon fonksiyonu ile yoğun katmanlar eklendiğinde doğruluk ve kayıp oranlarında sınırlı düzeyde iyileşme elde edilmiş ve modelin kararlılığı artmıştır. Ancak bu iyileşmeler yeterli düzeyde olmadığı gibi, özellikle kayıp grafiğindeki dalgalanmalar da dikkate alındığında, modelin yetersiz olduğu yorumu yapılabilir.



Şekil 4.3. ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.4. ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama kayıp grafiği

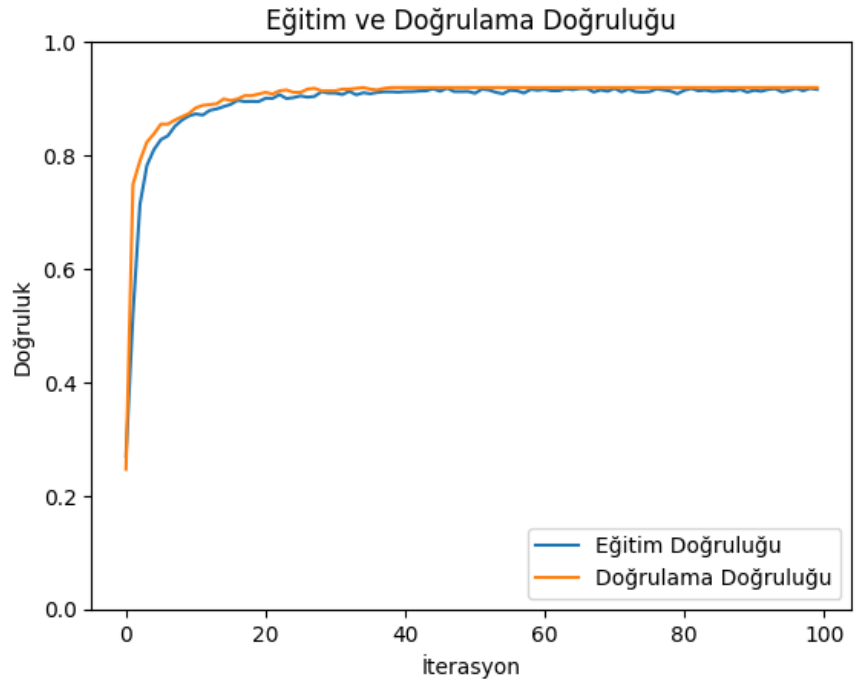
VGG16 modeline Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekledikten sonra, elde edilen sonuçlar;

- Eğitim doğruluk oranı: 0.9108
- Eğitim kayıp oranı: 0.2640
- Doğrulama doğruluk oranı: 0.9197
- Doğrulama kayıp oranı: 0.2523
- Test doğruluk oranı: 0.9071
- Test kayıp oranı: 0.2638

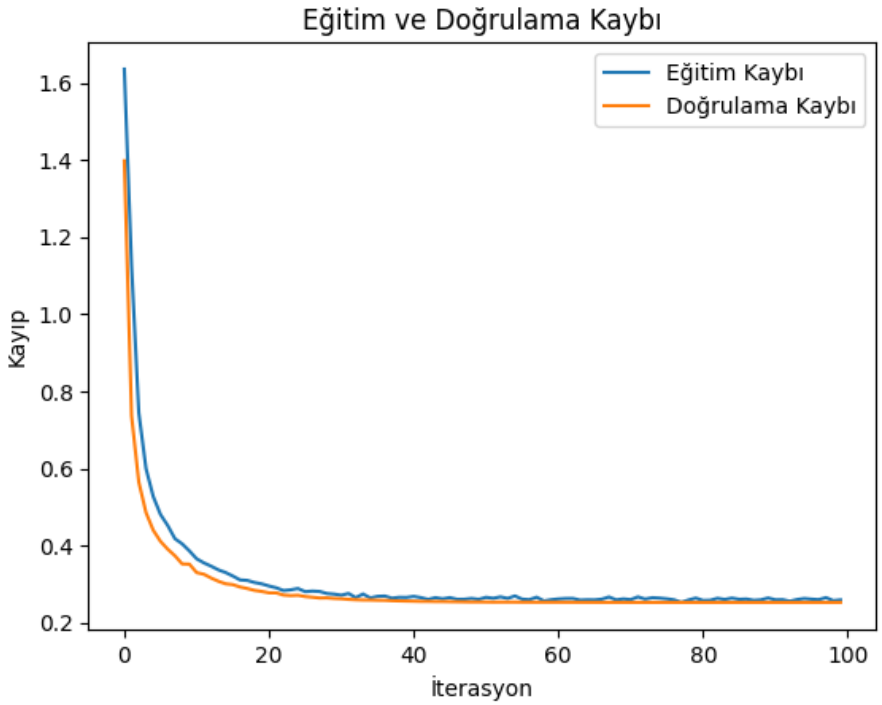
Elde edilen bulgulara göre, Leaky-ReLU aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir. Ancak bu iyileşmelere rağmen, elde edilen doğruluk ve kayıp değerleri, modelin başarılı veya kullanılabilir bir yapı olduğunu göstermek için yeterli düzeyde değildir.

Modelin doğruluk grafiği (Şekil 4.5) ve kayıp grafiği (Şekil 4.6) incelendiğinde, belirli bir iyileşme sağlandığı tespit edilmiştir. Bu durum, modelin kararlılığının arttığını göstermektedir. Ancak detaylı analiz yapıldığında, doğrulama doğruluk oranlarının eğitim doğruluk oranından daha yüksek olduğu ve doğrulama kayıp oranlarının eğitim kayıp oranlarından daha düşük olduğu gözlemlenmiştir.

VGG16 mimarisi üzerine Leaky-ReLU aktivasyon fonksiyonu ile yoğun katmanlar eklendiğinde, doğruluk ve kayıp grafiklerinde iyileşme elde edilmiş ve modelin kararlılığı artırılmıştır. Ancak genel doğruluk ve kayıp oranları dikkate alındığında geliştirilebilir bir model olduğuna analmına gelmektedir.



Şekil 4.5. Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.6. Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama kayıp grafiği

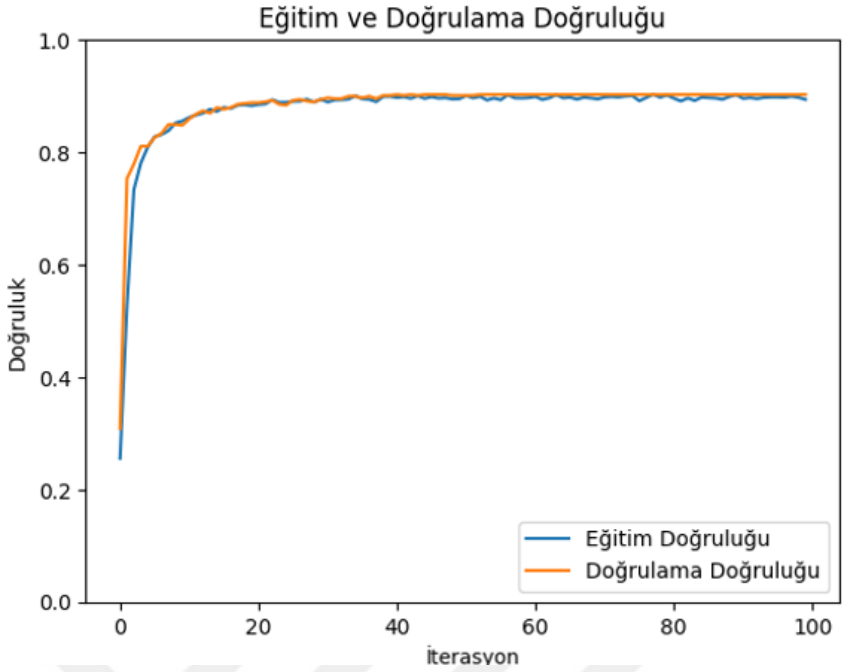
VGG16 modeline Swish aktivasyon fonksiyonu ile üç dense katmanı eklendikten sonra, elde edilen sonuçları;

- Eğitim doğruluk oranı: 0.8871
- Eğitim kayıp oranı: 0.3134
- Doğrulama doğruluk oranı: 0.9024
- Doğrulama kayıp oranı: 0.2861
- Test doğruluk oranı: 0.8940
- Test kayıp oranı: 0.2898

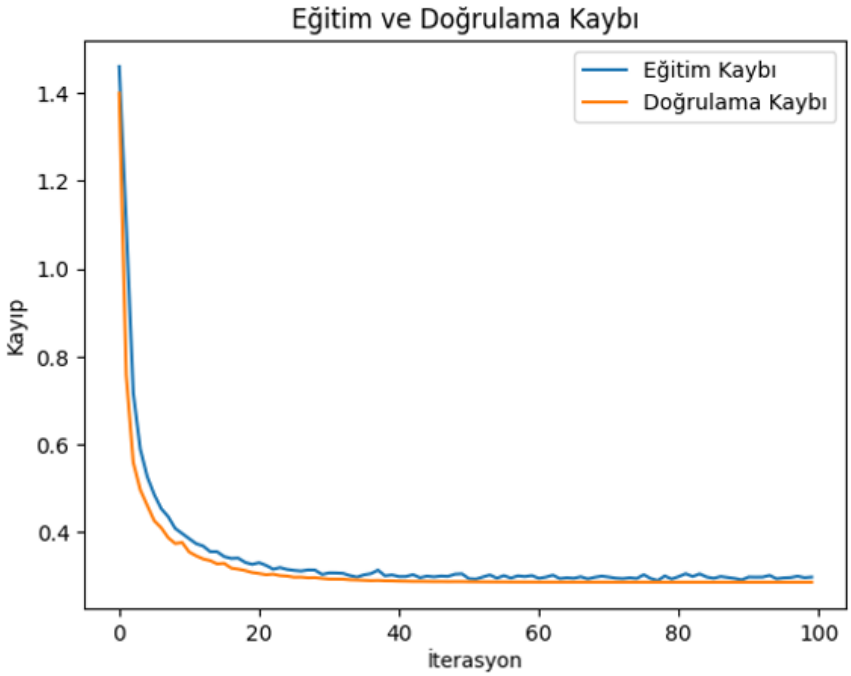
Elde edilen bulgulara göre, Swish aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir.

Modelin doğruluk grafiği (Şekil 4.7) ve kayıp grafiği (Şekil 4.8) incelendiğinde, belirli bir iyileşme sağlandığı tespit edilmiştir. Bu durum, modelin kararlılığının arttığını göstermektedir. Ancak detaylı analiz yapıldığında, doğrulama ve test doğruluk oranlarının eğitim doğruluk oranından daha yüksek olduğu ve doğrulama ile test kayıp oranlarının eğitim kayıp oranlarından daha düşük olduğu gözlemlenmiştir.

VGG16 mimarisi üzerine Swish aktivasyon fonksiyonu ile yoğun katmanlar eklendiğinde, doğruluk ve kayıp grafiklerinde iyileşme elde edilmiş ve modelin kararlılığı artırılmıştır. Ancak genel doğruluk ve kayıp oranları dikkate alındığında, bu yapılandırma beklentileri karşılamamaktadır.



Şekil 4.7. Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.8. Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama kayıp grafiği

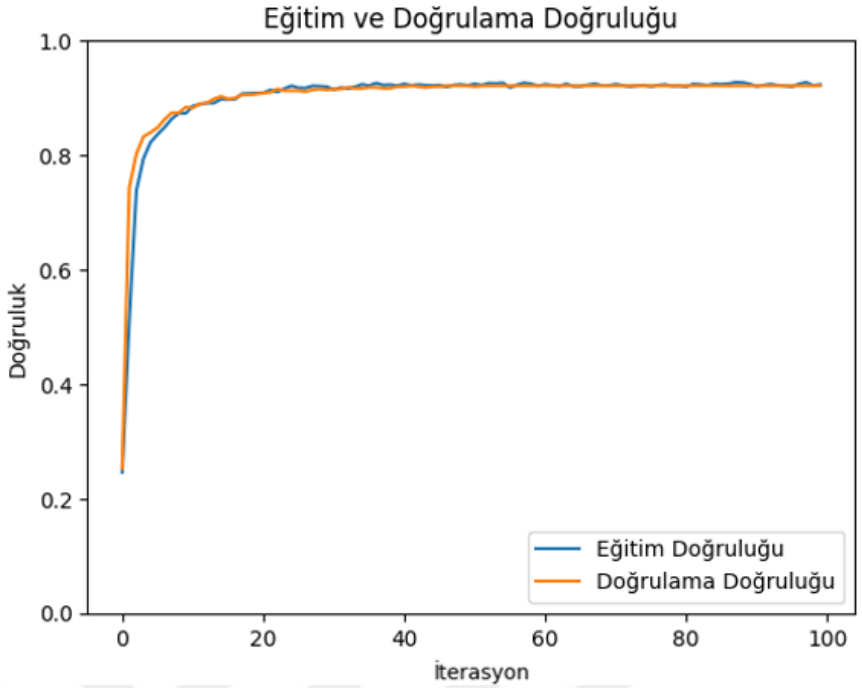
VGG16 modeline P-Swish aktivasyon fonksiyonu ile üç dense katmanı eklendikten sonra, elde edilen sonuçları;

- Eğitim doğruluk oranı: 0.9287
- Eğitim kayıp oranı: 0.2320
- Doğrulama doğruluk oranı: 0.9211
- Doğrulama kayıp oranı: 0.2420
- Test doğruluk oranı: 0.9127
- Test kayıp oranı: 0.2472

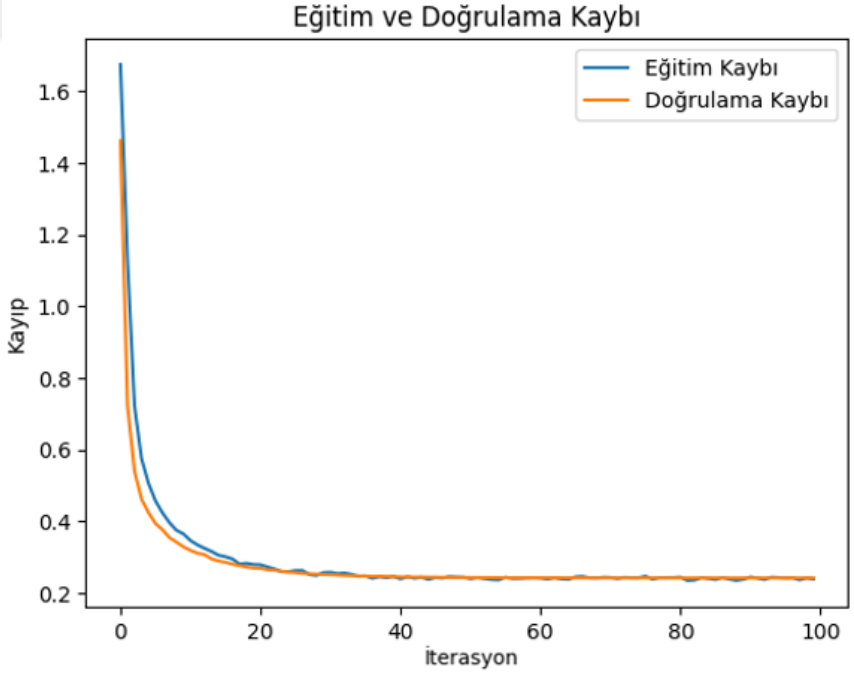
Elde edilen bulgulara göre, P-Swish aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir.

Modelin doğruluk grafiği (Şekil 4.9) ve kayıp grafiği (Şekil 4.10) incelendiğinde, belirli bir iyileşmenin sağlandığı tespit edilmiştir. Bu durum, modelin eğitim sürecinde daha kararlı bir yapı sergilediğini ortaya koymaktadır.

VGG16 mimarisi üzerine P-Swish aktivasyon fonksiyonu ile yoğun katmanlar eklendiğinde, doğruluk ve kayıp grafiklerinde iyileşme elde edilmiş; modelin kararlılığı artırılmış ve performans metrikleri iyileştirilmiştir. Elde edilen doğruluk ve kayıp oranlarının dengeli olması ve modelin istikrarlı bir performans sergilemesi sonucunda, bu yapılandırmanın başarılı olduğu ve tercih edilebilir bir model haline geldiği sonucuna varılmıştır.



Şekil 4.9. P-Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.10. P-Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama kayıp grafiği

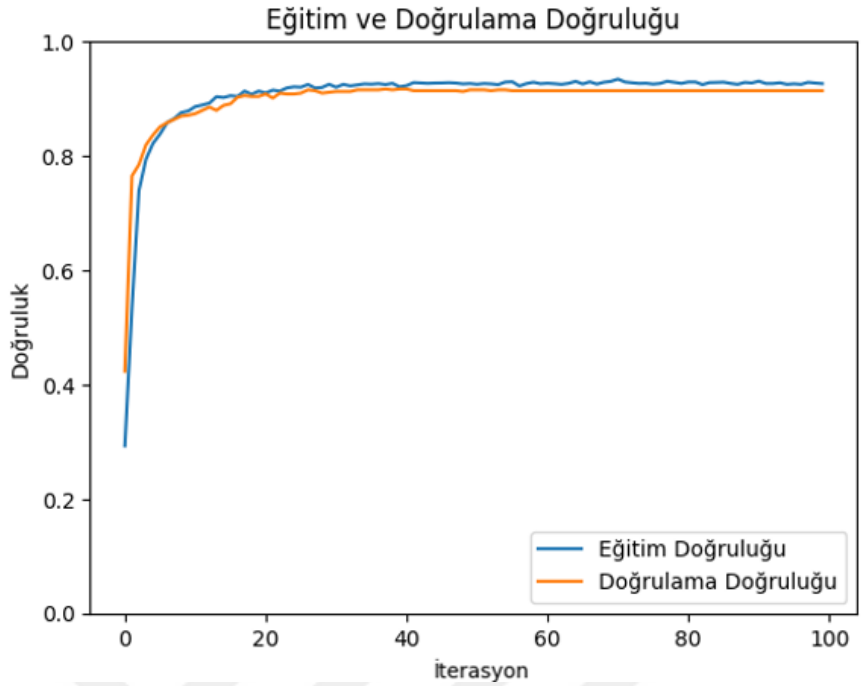
VGG16 modeline SaRa aktivasyon fonksiyonu ile üç dense katmanı ekledikten sonra, elde edilen sonuçları;

- Eğitim doğruluk oranı: 0.9293
- Eğitim kayıp oranı: 0.2255
- Doğrulama doğruluk oranı: 0.9139
- Doğrulama kayıp oranı: 0.2428
- Test doğruluk oranı: 0.9153
- Test kayıp oranı: 0.2426

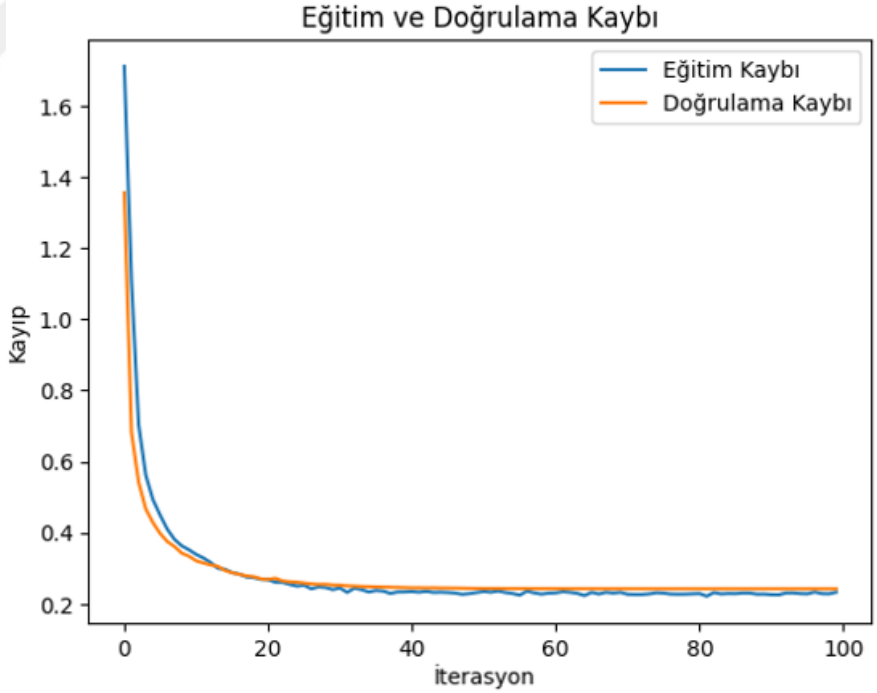
Elde edilen bulgulara göre, SaRa aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir.

Modelin doğruluk grafiği (Şekil 4.11) ve kayıp grafiği (Şekil 4.12) incelendiğinde, belirli bir iyileşmenin sağlandığı tespit edilmiştir. Bu durum, modelin eğitim sürecinde daha kararlı bir yapı sergilediğini ortaya koymaktadır.

VGG16 mimarisi üzerine SaRa aktivasyon fonksiyonu ve yoğun katmanlar eklendiğinde, doğruluk ve kayıp grafiklerinde iyileşme elde edilmiş; modelin kararlılığı artırılmış ve performans metrikleri iyileştirilmiştir. Elde edilen doğruluk ve kayıp oranlarının dengeli olması ve modelin istikrarlı bir performans sergilemesi sonucunda, bu yapılandırmanın başarılı olduğu ve tercih edilebilir bir model haline geldiği sonucuna varılmıştır.



Şekil 4.11. SaRa aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.12. SaRa aktivasyon fonksiyonu ile üç dense katmanı ekleyerek VGG16 modelin eğitim ve doğrulama kayıp grafiği

4.2. DenseNet121 Modelin Bulguları ve Tartışması

Diğer modellerde kullanılan veri seti ile DenseNet121 modelinin dense katmanları eklemeden elde edilen sonuçları;

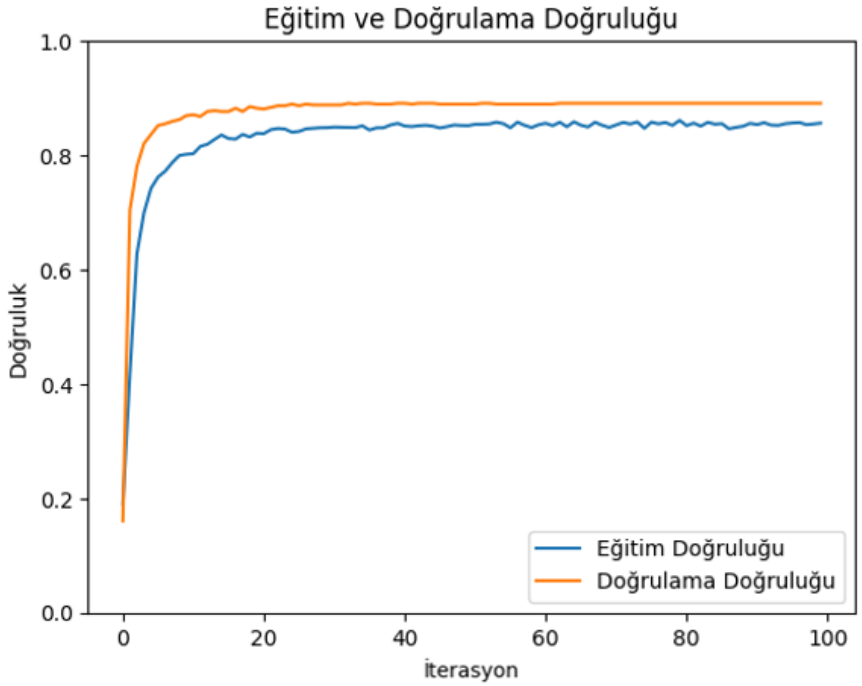
- Eğitim doğruluk oranı: 0.8575
- Eğitim kayıp oranı: 0.4024
- Doğrulama doğruluk oranı: 0.8910
- Doğrulama kayıp oranı: 0.2995
- Test doğruluk oranı: 0.8975
- Test kayıp oranı: 0.2909

Elde edilen sonuçlara göre, modelin doğruluk oranları oldukça düşük seviyelerde kalırken, kayıp oranları ise yüksek çıkmıştır. Ayrıca, dikkat çeken bir diğer durum, doğrulama ve test doğruluk oranlarının, eğitim doğruluk oranlarından daha yüksek olmasıdır. Bu durum, tüm iterasyonlar boyunca çizilen doğruluk grafiğinde (Şekil 4.13) açıkça gözlemlenmektedir.

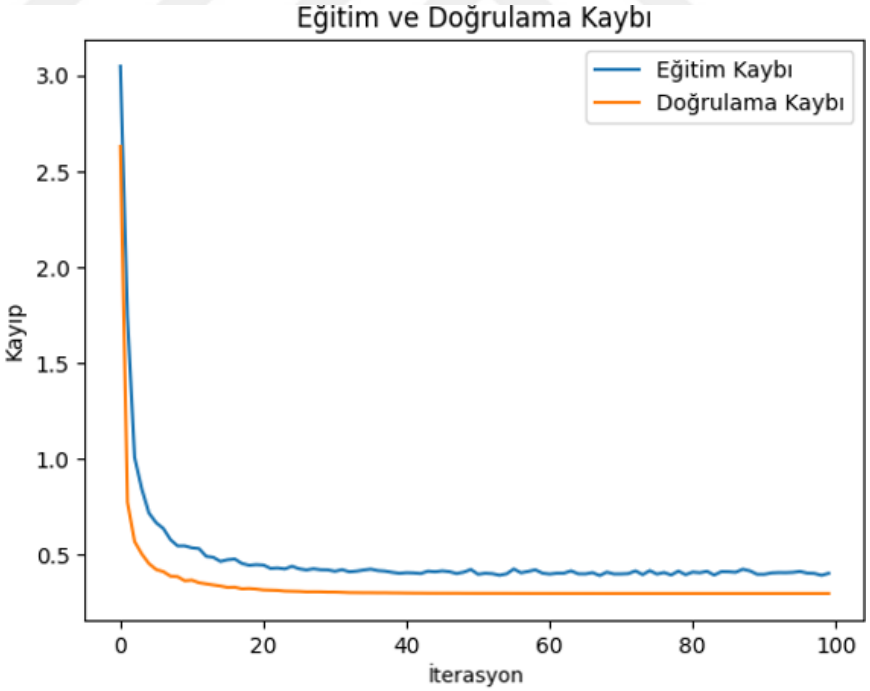
Benzer şekilde, doğrulama ve test kayıp oranlarının eğitim kayıp oranlarından daha düşük olduğu görülmektedir. Bu durum da iterasyonlar boyunca çizilen kayıp grafiğinde (Şekil 4.14) açıkça ortaya konmuştur.

Bu bulgular, modelin hem eğitim verileri üzerinde yeterince öğrenemediğini hem de genelleme yeteneğinin zayıf olduğunu göstermektedir. Eğitim doğruluğunun düşük kalması, modelin öğrenme sürecinde başarısız olduğunu; doğrulama ve test sonuçlarındaki tutarsızlıklar ise modelin kararsız bir yapı sergilediğini ortaya koymaktadır.

Tüm bu değerlendirmeler, modelin genel anlamda zayıf bir performans sergilediğini ve pratik uygulamalarda kullanılabilirlik açısından yetersiz olduğunu göstermektedir.



Şekil 4.13. DenseNet121 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.14. DenseNet121 modelin eğitim ve doğrulama kayıp grafiği

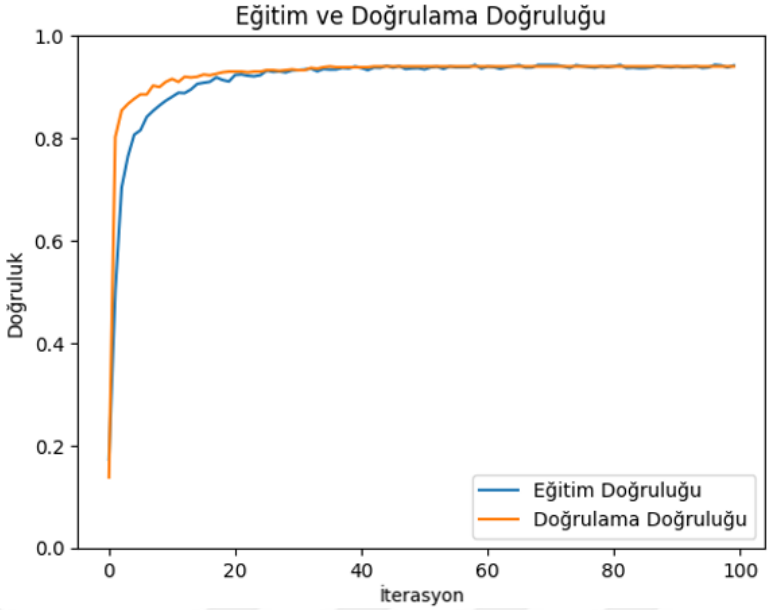
DenseNet121 modeline ReLU aktivasyon fonksiyonu ile üç dense katmanı ekledikten sonra, elde edilen sonuçları;

- Eğitim doğruluk oranı: 0.9376
- Eğitim kayıp oranı: 0.1983
- Doğrulama doğruluk oranı: 0.9397
- Doğrulama kayıp oranı: 0.1866
- Test doğruluk oranı: 0.9396
- Test kayıp oranı: 0.1845

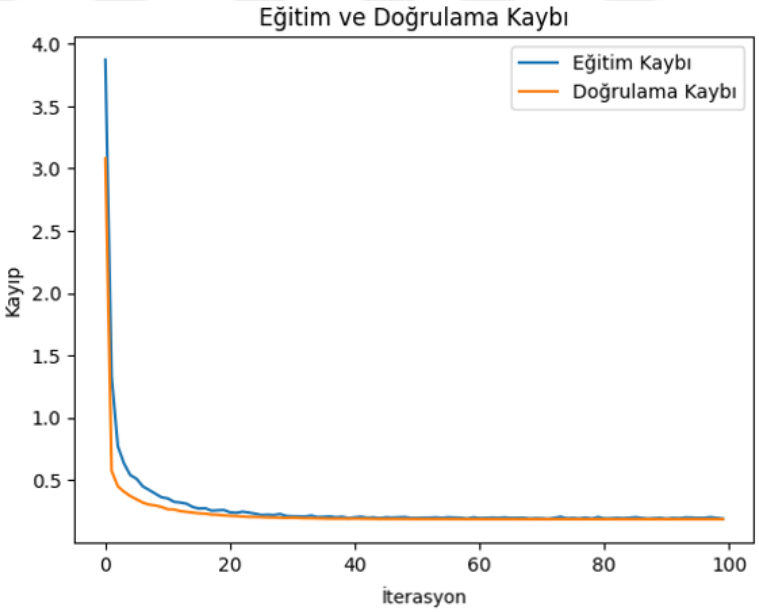
Elde edilen bulgulara göre, ReLU aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir.

Modelin doğruluk grafiği (Şekil 4.15) ve kayıp grafiği (Şekil 4.16) incelendiğinde, belirli bir iyileşme sağlandığı tespit edilmiştir. Bu durum, modelin kararlılığının arttığını göstermektedir. Ancak detaylı analiz yapıldığında, doğrulama doğruluk oranlarının eğitim doğruluk oranından daha yüksek olduğu ve doğrulama kayıp oranlarının eğitim kayıp oranlarından daha düşük olduğu gözlemlenmiştir.

DenseNet121 mimarisi üzerine ReLU aktivasyon fonksiyonu ile yoğun katmanlar eklendiğinde, doğruluk ve kayıp grafiklerinde iyileşme elde edilmiş ve modelin kararlılığı artırılmıştır. Ancak genel doğruluk ve kayıp oranları dikkate alındığında, modelin yetersiz olduğu ve geliştirilebileceği sonucuna varılabilir.



Şekil 4.15. ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.16. ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama kayıp grafiği

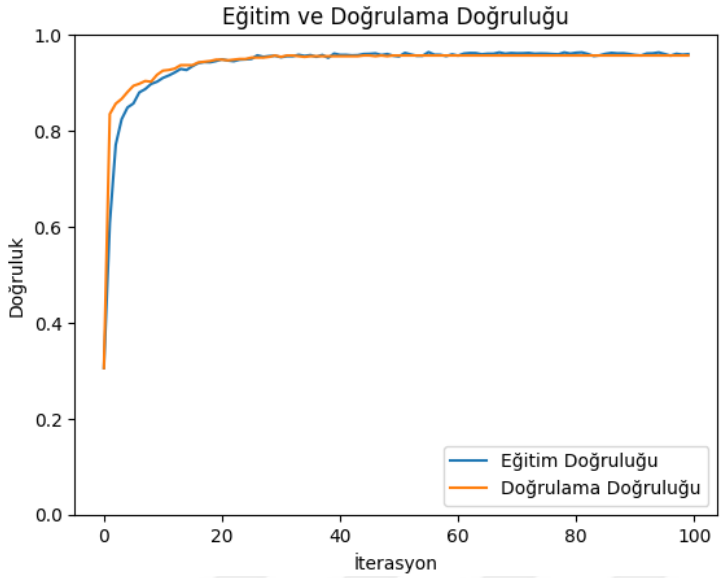
DenseNet121 modeline Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekledikten sonra, elde edilen sonuçları;

- Eğitim doğruluk oranı: 0.9628
- Eğitim kayıp oranı: 0.1262
- Doğrulama doğruluk oranı: 0.9570
- Doğrulama kayıp oranı: 0.1506
- Test doğruluk oranı: 0.9539
- Test kayıp oranı: 0.1547

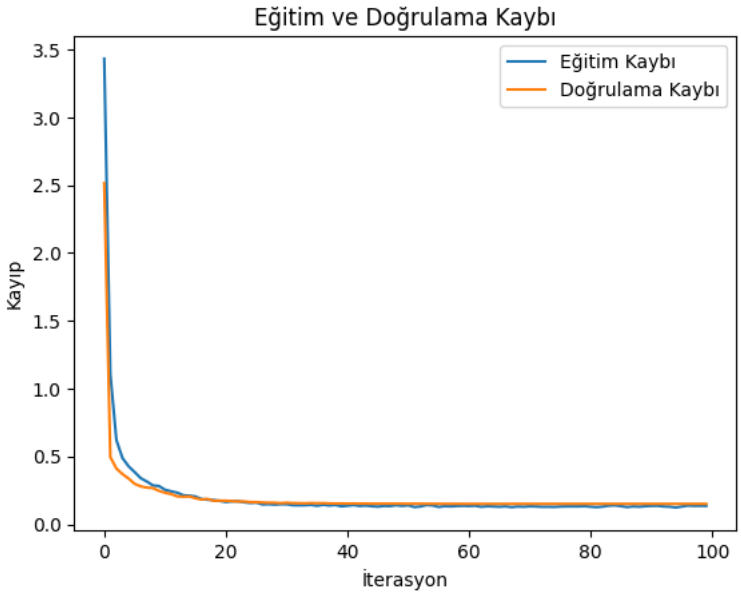
Elde edilen bulgulara göre, Leaky-ReLU aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir.

Modelin doğruluk grafiği (Şekil 4.17) ve kayıp grafiği (Şekil 4.18) incelendiğinde, belirli bir iyileşmenin sağlandığı tespit edilmiştir. Bu durum, modelin eğitim sürecinde daha kararlı bir yapı sergilediğini ortaya koymaktadır.

DenseNet121 mimarisi üzerine Leaky-ReLU aktivasyon fonksiyonu ile yoğun katmanlar eklendiğinde, doğruluk ve kayıp grafiklerinde iyileşme elde edilmiş; modelin kararlılığı artırılmış ve performans metrikleri iyileştirilmiştir. Elde edilen doğruluk ve kayıp oranlarının dengeli olması ve modelin istikrarlı bir performans sergilemesi sonucunda, bu yapılandırmanın başarılı olduğu ve tercih edilebilir bir model haline geldiği sonucuna varılmıştır.



Şekil 4.17. Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.18. Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelim eğitim ve doğrulama kayıp grafiği

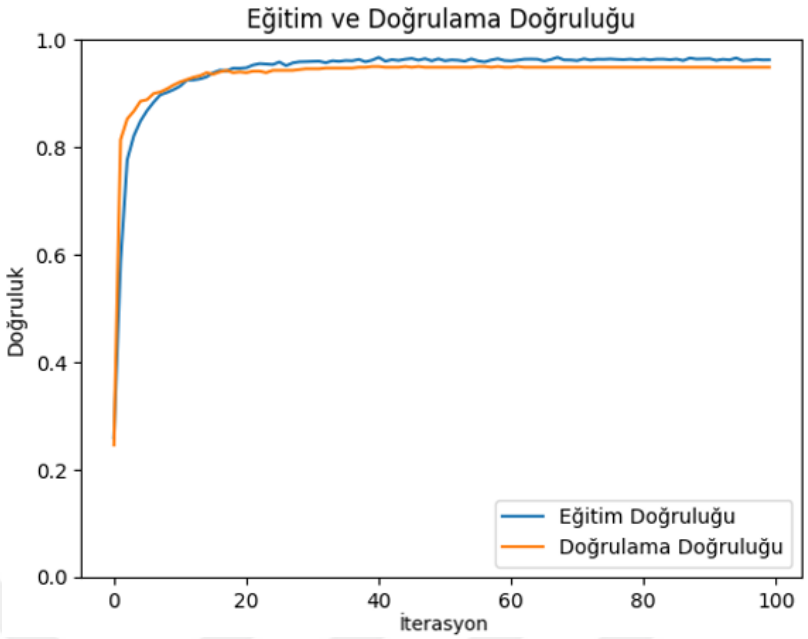
DenseNet121 modeline Swish aktivasyon fonksiyonu ile üç dense katmanı ekledikten sonra, elde edilen sonuçları;

- Eğitim doğruluk oranı: 0.9630
- Eğitim kayıp oranı: 0.1315
- Doğrulama doğruluk oranı: 0.9484
- Doğrulama kayıp oranı: 0.1508
- Test doğruluk oranı: 0.9577
- Test kayıp oranı: 0.1516

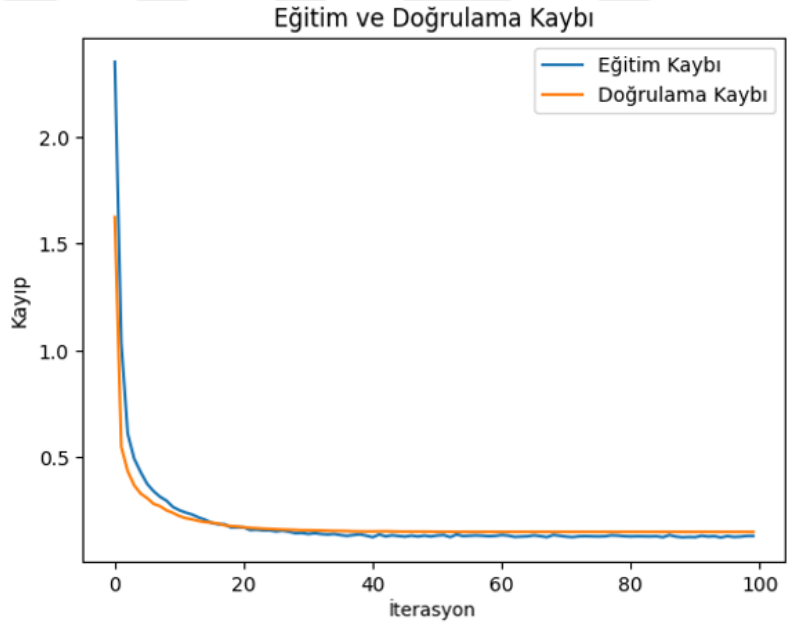
Elde edilen bulgulara göre, Swish aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir.

Modelin doğruluk grafiği (Şekil 4.19) ve kayıp grafiği (Şekil 4.20) incelendiğinde, belirli bir iyileşmenin sağlandığı tespit edilmiştir.

DenseNet121 mimarisi üzerine Swish aktivasyon fonksiyonu ile yoğun katmanlar eklendiğinde, doğruluk ve kayıp grafiklerinde iyileşme elde edilmiş; modelin kararlılığı artırılmış ve performans metrikleri iyileştirilmiştir. Elde edilen doğruluk ve kayıp oranlarının dengeli olması ve modelin istikrarlı bir performans sergilemesi sonucunda, bu yapılandırmanın başarılı olduğu ve tercih edilebilir bir model haline geldiği sonucuna varılmıştır.



Şekil 4.19. Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.20. Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama kayıp grafiği

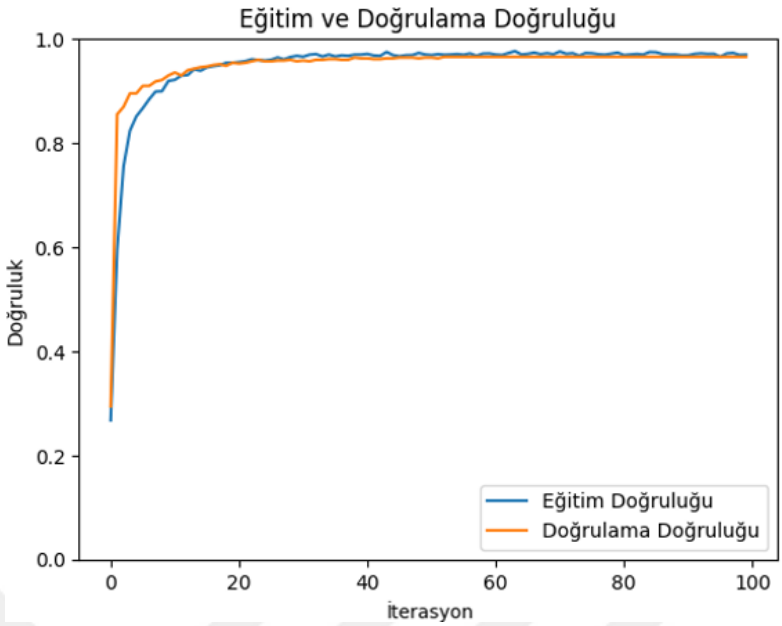
DenseNet121 modeline P-Swish aktivasyon fonksiyonu ile üç dense katmanı ekledikten sonra, elde edilen sonuçları;

- Eğitim doğruluk oranı: 0.9712
- Eğitim kayıp oranı: 0.1155
- Doğrulama doğruluk oranı: 0.9656
- Doğrulama kayıp oranı: 0.1357
- Test doğruluk oranı: 0.9568
- Test kayıp oranı: 0.1418

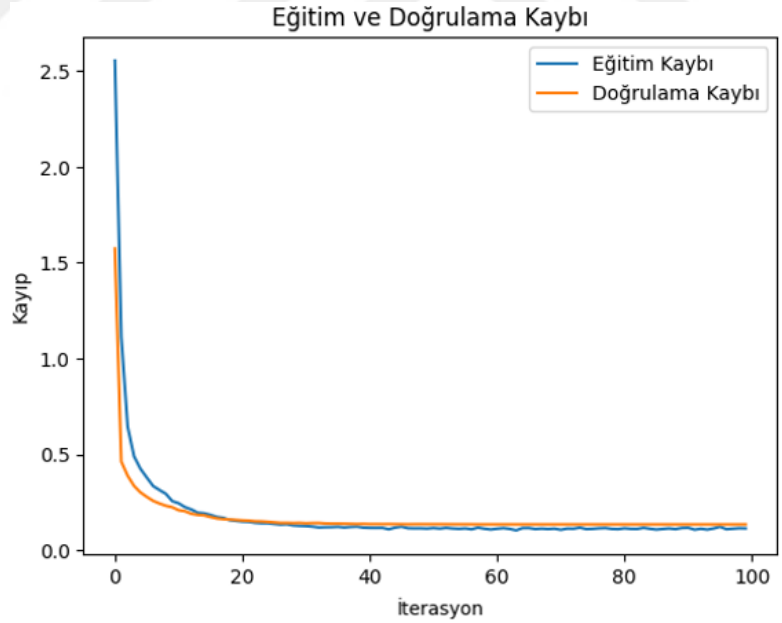
Elde edilen bulgulara göre, P-Swish aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir.

Modelin doğruluk grafiği (Şekil 4.21) ve kayıp grafiği (Şekil 4.22) incelendiğinde, belirli bir iyileşmenin sağlandığı tespit edilmiştir.

DenseNet121 mimarisi üzerine P-Swish aktivasyon fonksiyonu ile yoğun katmanlar eklendiğinde, doğruluk ve kayıp grafiklerinde iyileşme elde edilmiş; modelin kararlılığı artırılmış ve performans metrikleri iyileştirilmiştir. Elde edilen doğruluk ve kayıp oranlarının dengeli olması ve modelin istikrarlı bir performans sergilemesi sonucunda, bu yapılandırmanın başarılı olduğu ve tercih edilebilir bir model haline geldiği sonucuna varılmıştır.



Şekil 4.21. P-Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.22. P-Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama kayıp grafiği

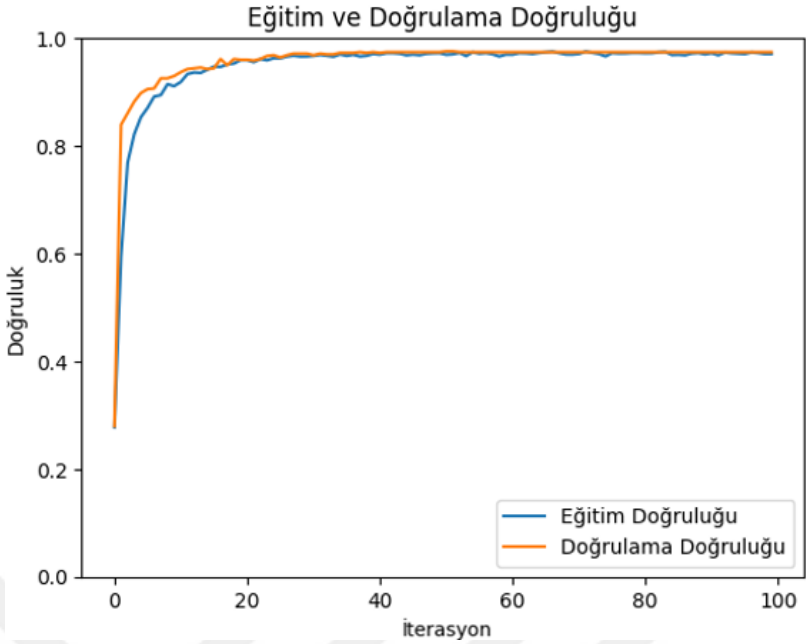
DenseNet121 modeline SaRa aktivasyon fonksiyonu ile üç dense katmanı ekledikten sonra, elde edilen sonuçları;

- Eğitim doğruluk oranı: 0.9704
- Eğitim kayıp oranı: 0.1077
- Doğrulama doğruluk oranı: 0.9742
- Doğrulama kayıp oranı: 0.1268
- Test doğruluk oranı: 0.9628
- Test kayıp oranı: 0.1460

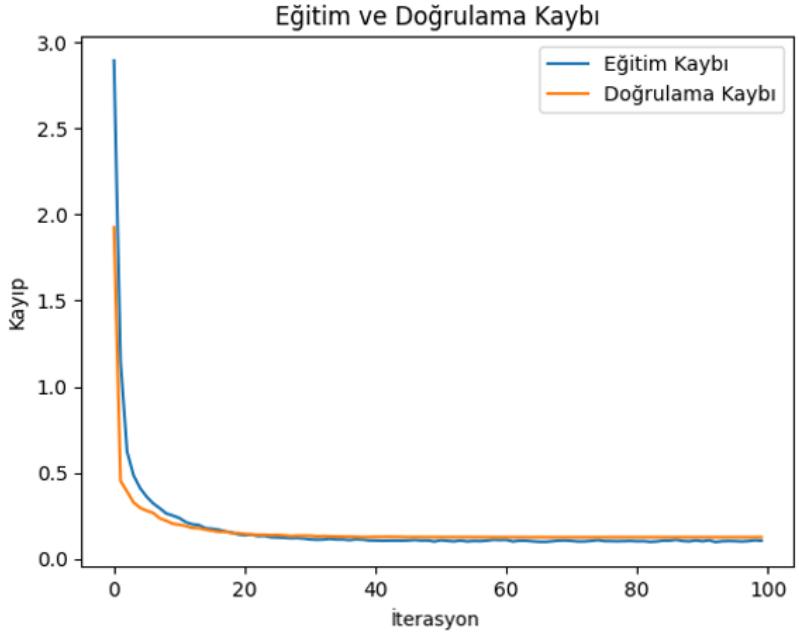
Elde edilen bulgulara göre, SaRa aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir.

Modelin doğruluk grafiği (Şekil 4.23) ve kayıp grafiği (Şekil 4.24) incelendiğinde, belirli bir iyileşmenin sağlandığı tespit edilmiştir. Bu durum, modelin eğitim sürecinde daha kararlı bir yapı sergilediğini ortaya koymaktadır.

DenseNet121 mimarisi üzerine SaRa aktivasyon fonksiyonu ile yoğun katmanlar eklendiğinde, doğruluk ve kayıp grafiklerinde iyileşme elde edilmiş; modelin kararlılığı artırılmış ve performans metrikleri iyileştirilmiştir. Elde edilen doğruluk ve kayıp oranlarının dengeli olması ve modelin istikrarlı bir performans sergilemesi sonucunda, bu yapılandırmanın başarılı olduğu ve tercih edilebilir bir model haline geldiği sonucuna varılmıştır.



Şekil 4.23. SaRa aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.24. SaRa aktivasyon fonksiyonu ile üç dense katmanı ekleyerek DenseNet121 modelin eğitim ve doğrulama kayıp grafiği

4.3. Inception-V3 Modelin Bulguları ve Tartışması

Standart bir Inception-V3 modelinin dense katmanları eklemeden tüm mimariler için ortak kullandığımız veri seti ile elde edilen sonuçlar;

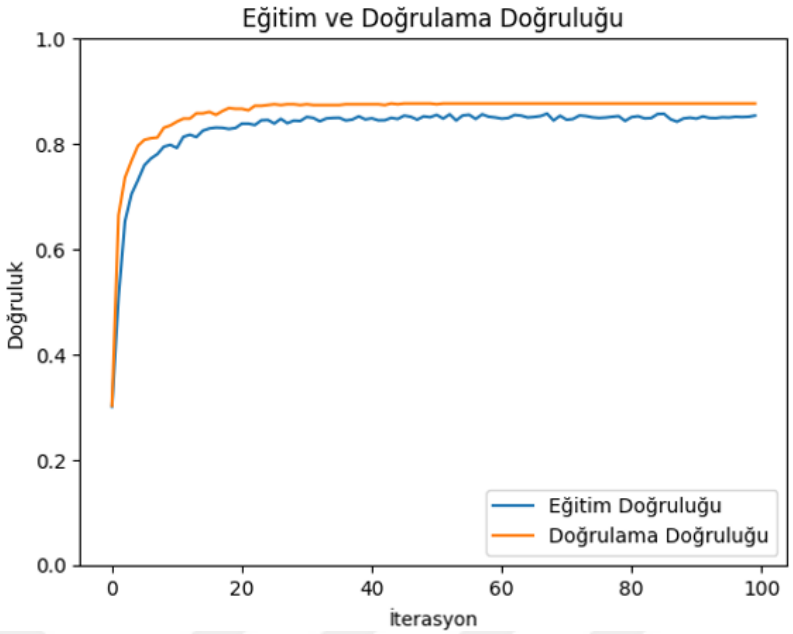
- Eğitim doğruluk oranı: 0.8513
- Eğitim kayıp oranı: 0.4137
- Doğrulama doğruluk oranı: 0.8766
- Doğrulama kayıp oranı: 0.3188
- Test doğruluk oranı: 0.8814
- Test kayıp oranı: 0.3680

Elde edilen sonuçlara göre, modelin doğruluk oranları oldukça düşük seviyelerde kalırken, kayıp oranları ise yüksek çıkmıştır. Ayrıca, dikkat çeken bir diğer durum, doğrulama ve test doğruluk oranlarının, eğitim doğruluk oranlarından daha yüksek olmasıdır. Bu durum, tüm iterasyonlar boyunca çizilen doğruluk grafiğinde (Şekil 4.25) açıkça gözlemlenmektedir.

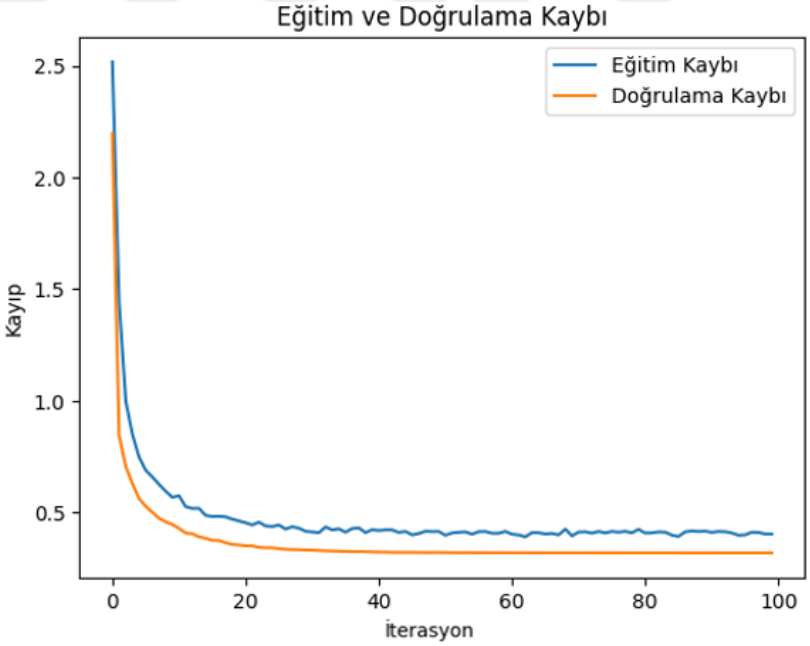
Benzer şekilde, doğrulama ve test kayıp oranlarının eğitim kayıp oranlarından daha düşük olduğu görülmektedir. Bu durum da iterasyonlar boyunca çizilen kayıp grafiğinde (Şekil 4.26) açıkça ortaya konmuştur.

Bu bulgular, modelin hem eğitim verileri üzerinde yeterince öğrenemediğini hem de genelleme yeteneğinin zayıf olduğunu göstermektedir. Eğitim doğruluğunun düşük kalması, modelin öğrenme sürecinde başarısız olduğunu; doğrulama ve test sonuçlarındaki tutarsızlıklar ise modelin kararsız bir yapı sergilediğini ortaya koymaktadır.

Tüm bu değerlendirmeler, modelin genel anlamda zayıf bir performans sergilediğini ve pratik uygulamalarda kullanılabilirlik açısından yetersiz olduğunu göstermektedir.



Şekil 4.25. Inception-V3 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.26. Inception-V3 modelin eğitim ve doğrulama kayıp grafiği

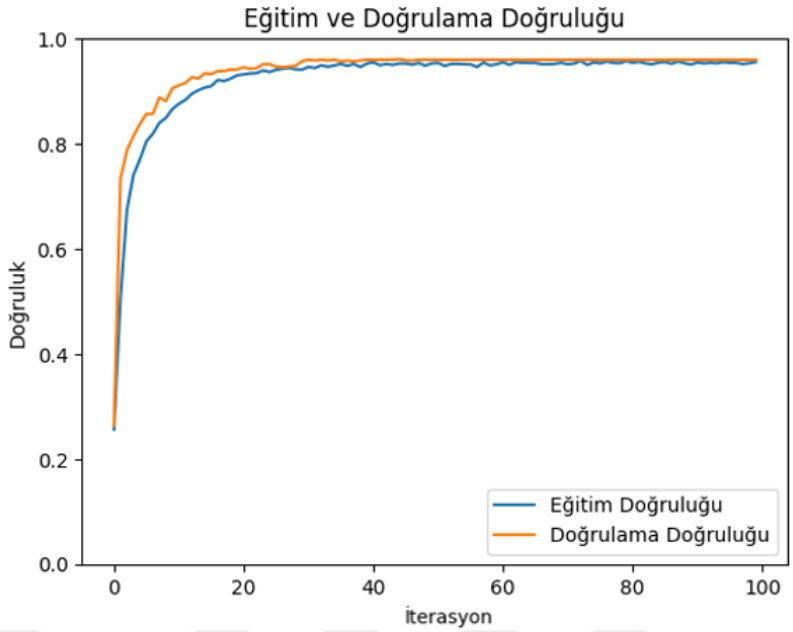
Inception-V3 modeline ReLU aktivasyon fonksiyonu ile üç dense katmanı ekledikten sonra, elde edilen sonuçları;

- Eğitim doğruluk oranı: 0.9523
- Eğitim kayıp oranı: 0.1657
- Doğrulama doğruluk oranı: 0.9598
- Doğrulama kayıp oranı: 0.1463
- Test doğruluk oranı: 0.9298
- Test kayıp oranı: 0.2074

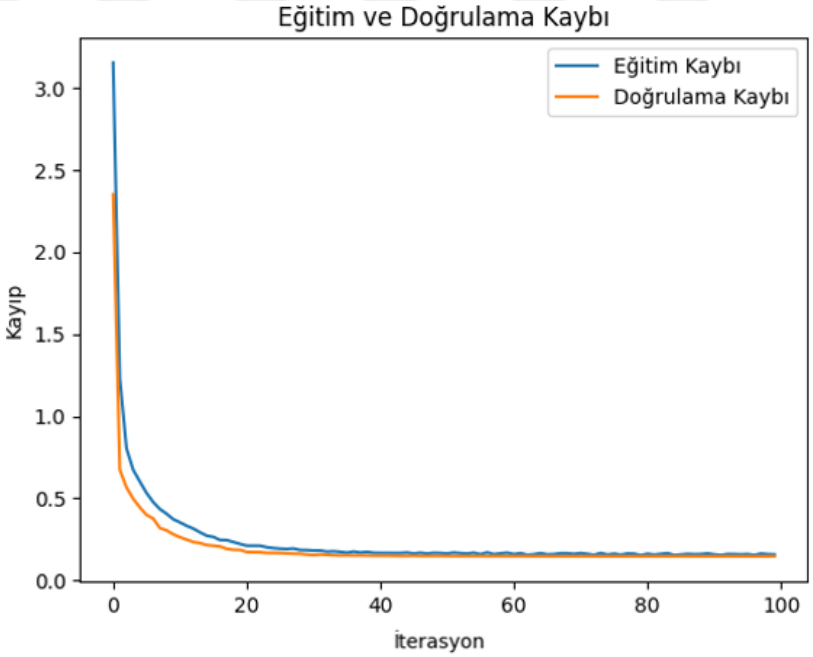
Elde edilen bulgulara göre, ReLU aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir.

Modelin doğruluk grafiği (Şekil 4.27) incelendiğinde, belirli bir iyileşme sağlandığı tespit edilmiştir. Bununla birlikte, kayıp grafiği (Şekil 4.28), doğrulama ve test kayıp oranlarının eğitim kayıp oranlarına kıyasla daha düşük seviyelerde seyrettiğini göstermektedir. Bu durum, modelin eğitim verisine kıyasla doğrulama ve test verileri üzerinde daha iyi performans sergilediği izlenimini oluştursa da, genel başarı seviyeleri dikkate alındığında modelin istikrarlı ve güvenilir bir yapı sergilemediği sonucuna varılmıştır.

Inception-V3 mimarisi üzerine ReLU aktivasyon fonksiyonu ile yoğun katmanlar eklendiğinde doğruluk ve kayıp oranlarında sınırlı düzeyde iyileşme elde edilmiş ve modelin kararlılığı artmıştır. Ancak bu iyileşmeler yeterli düzeyde olmadığı gibi, özellikle kayıp grafiğindeki dalgalanmalar da dikkate alındığında, yapılandırılan bu model tercih edilmeyen bir model olduğuna analmına gelmektedir.



Şekil 4.27. ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.28. ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama kayıp grafiği

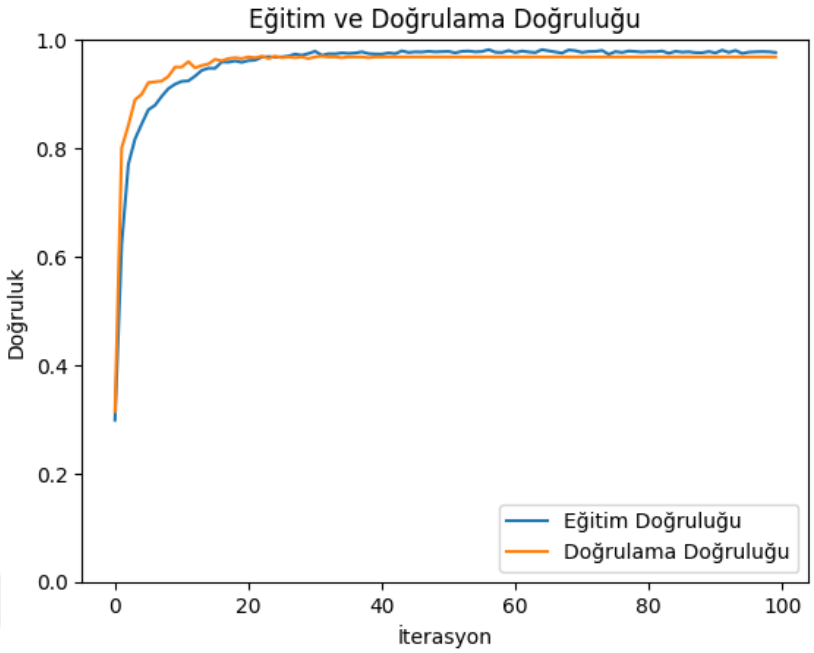
Inception-V3 modeline Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekledikten sonra, elde edilen sonuçları;

- Eğitim doğruluk oranı: 0.9775
- Eğitim kayıp oranı: 0.0885
- Doğrulama doğruluk oranı: 0.9684
- Doğrulama kayıp oranı: 0.1189
- Test doğruluk oranı: 0.9466
- Test kayıp oranı: 0.1678

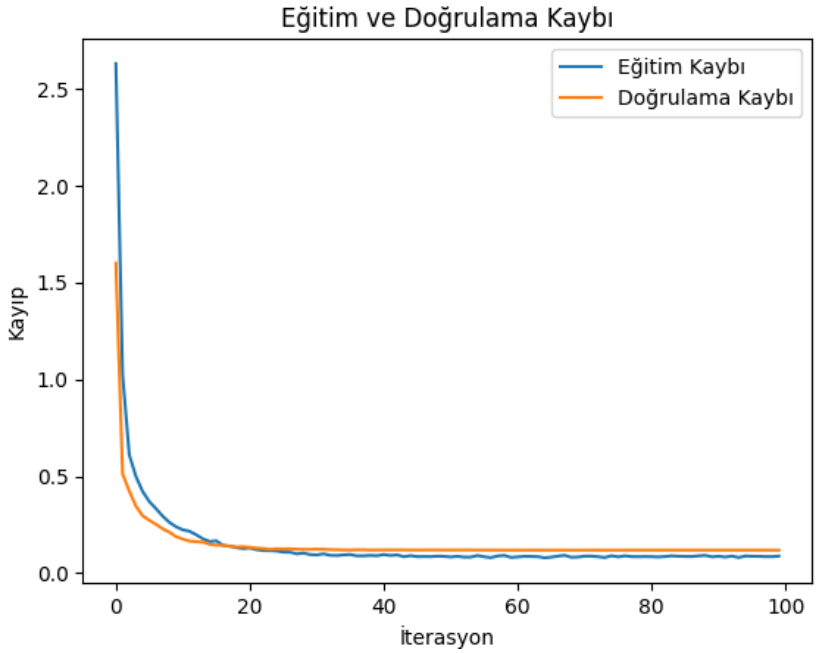
Elde edilen bulgulara göre, Leaky-ReLU aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir. Bu da modelin performanını iyileşmesine anlamına gelmektedir. Bu durum, modelin genel performansında iyileşme sağlandığını göstermektedir.

Modelin doğruluk grafiği (Şekil 4.29) ve kayıp grafiği (Şekil 4.30) incelendiğinde, doğruluk oranında artış ve kayıp oranında azalma gözlemlenmiş; bu da modelin belirli bir iyileşmenin sağlandığı tespit edilmiştir. Bu durum, modelin eğitim sürecinde daha kararlı bir yapı sergilediğini ortaya koymaktadır.

Inception-V3 mimarisi üzerine Leaky-ReLU aktivasyon fonksiyonu ile yoğun katmanlar eklendiğinde, doğruluk ve kayıp grafiklerinde iyileşme elde edilmiş; modelin kararlılığı artırılmış ve performans metrikleri iyileştirilmiştir. Grafiklerde görüldüğü üzere, eğitim doğruluğu doğrulama doğruluğundan yüksek olmakla birlikte, doğruluk ve kayıp oranlarının dengeli seyretmesi, ve modelin istikrarlı bir performans sergilemesi sonucunda, bu yapılandırmanın başarılı olduğu ve tercih edilebilir bir model haline geldiği sonucuna varılmıştır.



Şekil 4.29. Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.30. Leaky-ReLU aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama kayıp grafiği

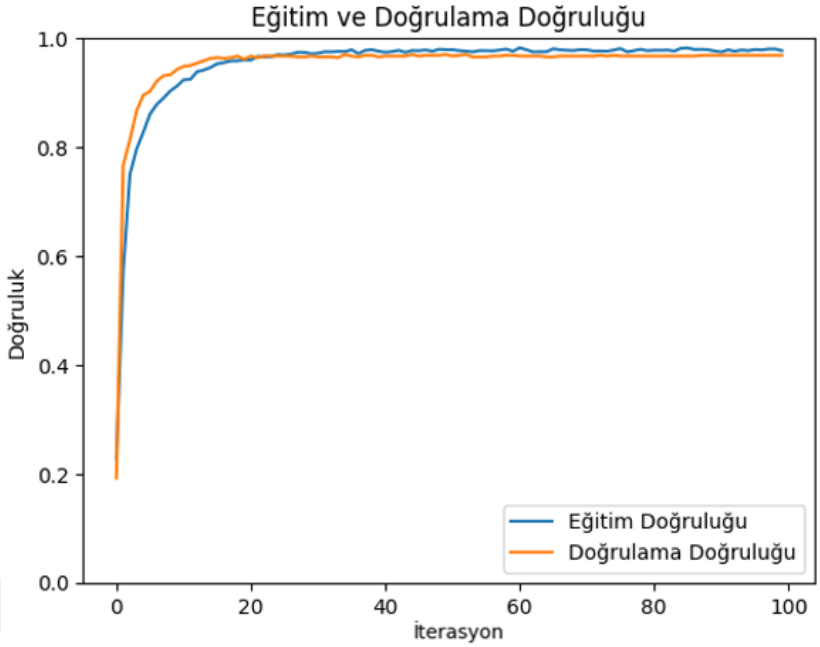
Inception-V3 modeline Swish aktivasyon fonksiyonu ile üç dense katmanı ekledikten sonra, elde edilen sonuçları;

- Eğitim doğruluk oranı: 0.9767
- Eğitim kayıp oranı: 0.0993
- Doğrulama doğruluk oranı: 0.9684
- Doğrulama kayıp oranı: 0.1068
- Test doğruluk oranı: 0.9501
- Test kayıp oranı: 0.1617

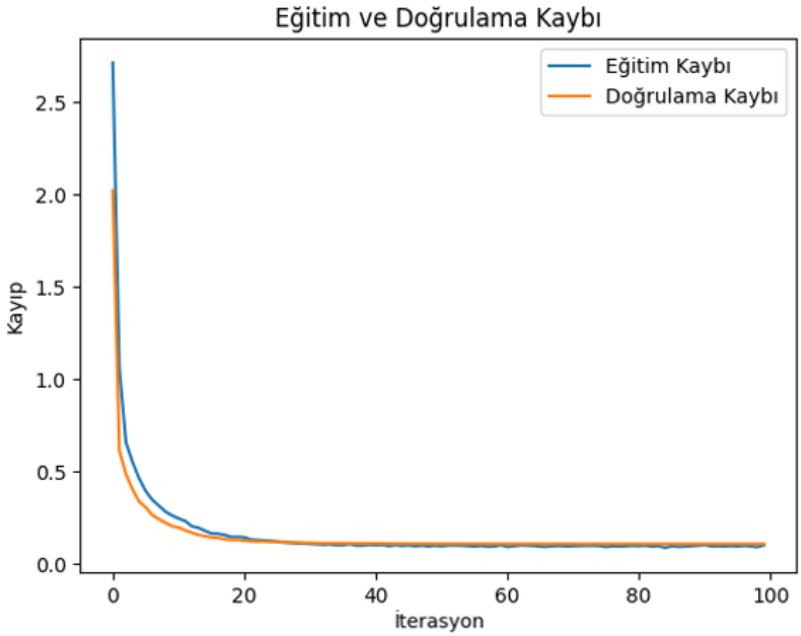
Elde edilen bulgulara göre, Swish aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir. Bu da modelin performanını iyileşmesine anlamına gelmektedir. Bu durum, modelin genel performansında iyileşme sağlandığını göstermektedir.

Modelin doğruluk grafiği (Şekil 4.31) ve kayıp grafiği (Şekil 4.32) incelendiğinde, doğruluk oranında artış ve kayıp oranında azalma gözlemlenmiş; bu da modelin belirli bir iyileşmenin sağlandığı tespit edilmiştir. Bu durum, modelin eğitim sürecinde daha kararlı bir yapı sergilediğini ortaya koymaktadır.

Inception-V3 mimarisi üzerine Swish aktivasyon fonksiyonu ile yoğun katmanlar eklendiğinde, doğruluk ve kayıp grafiklerinde iyileşme elde edilmiş; modelin kararlılığı artırılmış ve performans metrikleri iyileştirilmiştir. Grafiklerde görüldüğü üzere, eğitim doğruluğu doğrulama doğruluğundan yüksek olmakla birlikte, doğruluk ve kayıp oranlarının dengeli seyretmesi, ve modelin istikrarlı bir performans sergilemesi sonucunda, bu yapılandırmanın başarılı olduğu ve tercih edilebilir bir model haline geldiği sonucuna varılmıştır.



Şekil 4.31. Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.32. Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama kayıp grafiği

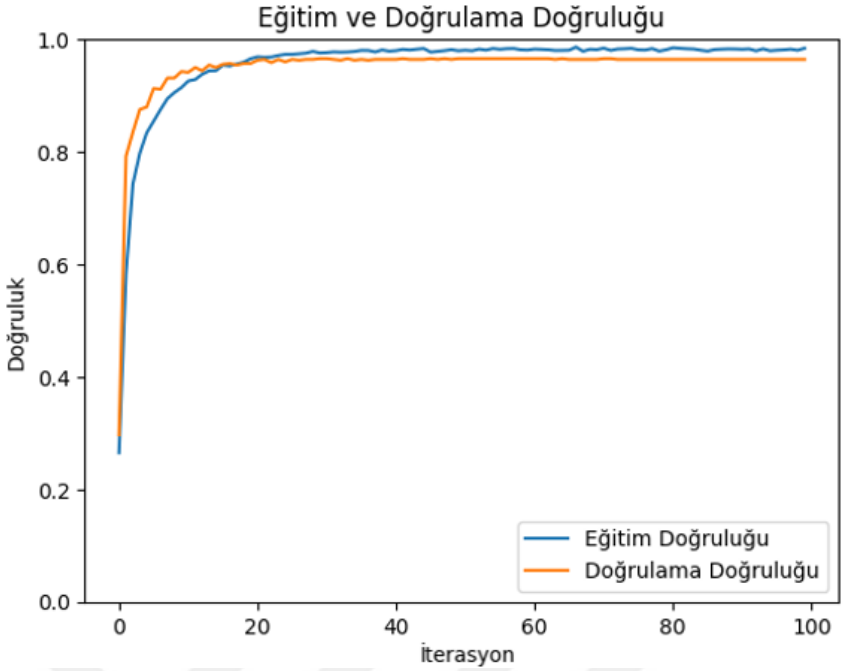
Inception-V3 modeline P-Swish aktivasyon fonksiyonu ile üç dense katmanı ekledikten sonra, elde edilen sonuçları;

- Eğitim doğruluk oranı: 0.9846
- Eğitim kayıp oranı: 0.0853
- Doğrulama doğruluk oranı: 0.9641
- Doğrulama kayıp oranı: 0.1192
- Test doğruluk oranı: 0.9482
- Test kayıp oranı: 0.1693

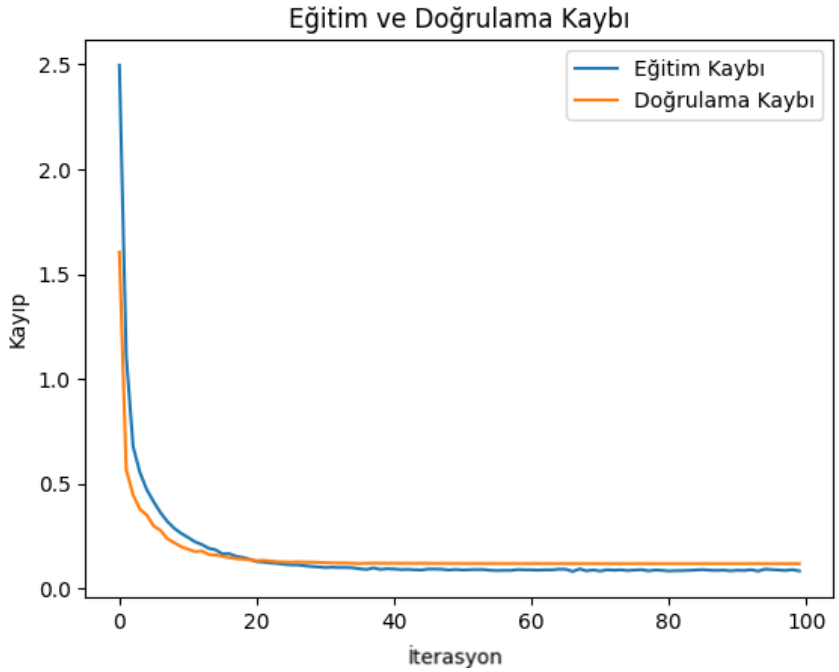
Elde edilen bulgulara göre, P-Swish aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir.

Modelin doğruluk grafiği (Şekil 4.33) ve kayıp grafiği (Şekil 4.34) incelendiğinde, belirli bir iyileşmenin sağlandığı tespit edilmiştir. Bu durum, modelin eğitim sürecinde daha kararlı bir yapı sergilediğini ortaya koymaktadır.

Inception-V3 mimarisi üzerine P-Swish aktivasyon fonksiyonu ile yoğun katmanlar eklendiğinde, doğruluk ve kayıp grafiklerinde iyileşme elde edilmiş; modelin kararlılığı artırılmış ve performans metrikleri iyileştirilmiştir. Elde edilen doğruluk ve kayıp oranlarının dengeli olması ve modelin istikrarlı bir performans sergilemesi sonucunda, bu yapılandırmanın başarılı olduğu ve tercih edilebilir bir model haline geldiği sonucuna varılmıştır.



Şekil 4.33. P-Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.34. P-Swish aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama kayıp grafiği

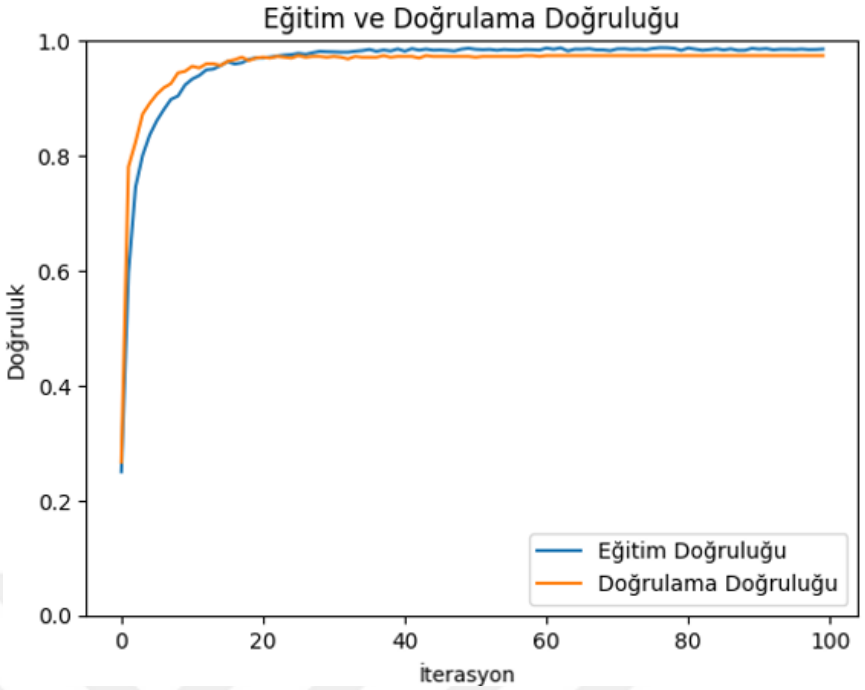
Inception-V3 modeline SaRa aktivasyon fonksiyonu ile üç dense katmanı ekledikten sonra, elde edilen sonuçları;

- Eğitim doğruluk oranı: 0.9849
- Eğitim kayıp oranı: 0.0769
- Doğrulama doğruluk oranı: 0.9742
- Doğrulama kayıp oranı: 0.1085
- Test doğruluk oranı: 0.9546
- Test kayıp oranı: 0.1653

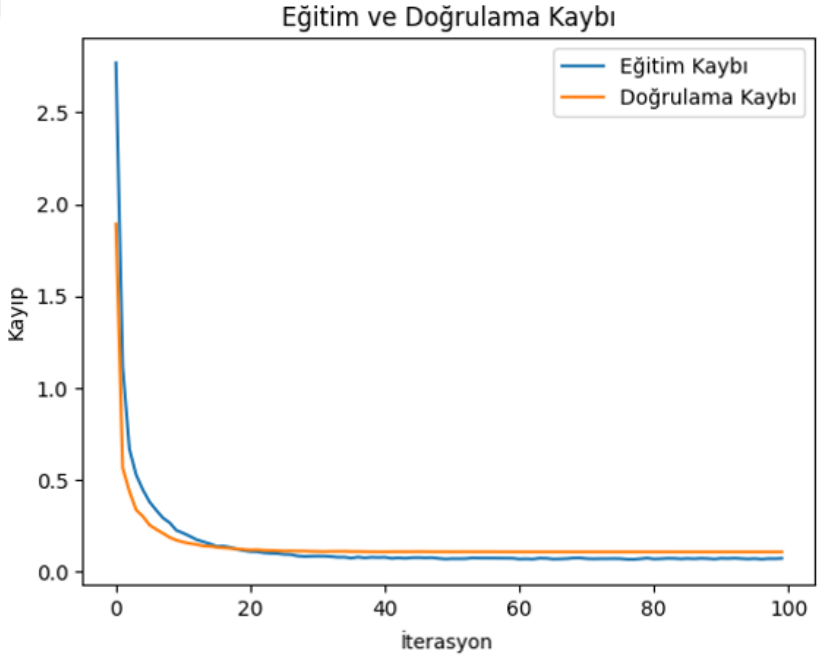
Elde edilen bulgulara göre, SaRa aktivasyon fonksiyonu ile birlikte yoğun (dense) katmanların modele eklenmesi sonucunda doğruluk oranlarında artış, kayıp oranlarında ise azalma gözlemlenmiştir.

Modelin doğruluk grafiği (Şekil 4.35) ve kayıp grafiği (Şekil 4.36) incelendiğinde, belirli bir iyileşmenin sağlandığı tespit edilmiştir.

Inception-V3 mimarisi üzerine SaRa aktivasyon fonksiyonu ile yoğun katmanlar eklendiğinde, doğruluk ve kayıp grafiklerinde iyileşme elde edilmiş; modelin kararlılığı artırılmış ve performans metrikleri iyileştirilmiştir. Elde edilen doğruluk ve kayıp oranlarının dengeli olması ve modelin istikrarlı bir performans sergilemesi sonucunda, bu yapılandırmanın başarılı olduğu ve tercih edilebilir bir model haline geldiği sonucuna varılmıştır.



Şekil 4.35. SaRa aktivasyon fonksiyonu ile üç dense katmanı ekleyerek Inception-V3 modelin eğitim ve doğrulama doğruluk grafiği



Şekil 4.36. SaRa aktivasyon fonksiyonu ile üç dense ekleyerek Inception-V3 modelin eğitim ve doğrulama kayıp grafiği

5. SONUÇ

5.1. VGG16 Modelin Sonuçları

Modelin kararlılığını değerlendirmek amacıyla öncelikle doğruluk ve kayıp grafiklerinin incelenmesi gerekmektedir. Ek yoğun (dense) katmanların bulunmadığı temel modelde, hem eğitim hem de doğrulama doğrulukları düşüktür. Doğrulama doğruluğunun, eğitim doğruluğundan biraz daha yüksek olması ise zayıf öğrenme ve sınırlı genelleme yeteneğine işaret etmektedir. Bu durum, kayıp eğrileri ile de desteklenmekte ve modelin genel olarak düşük performans sergilediğini ortaya koymaktadır.

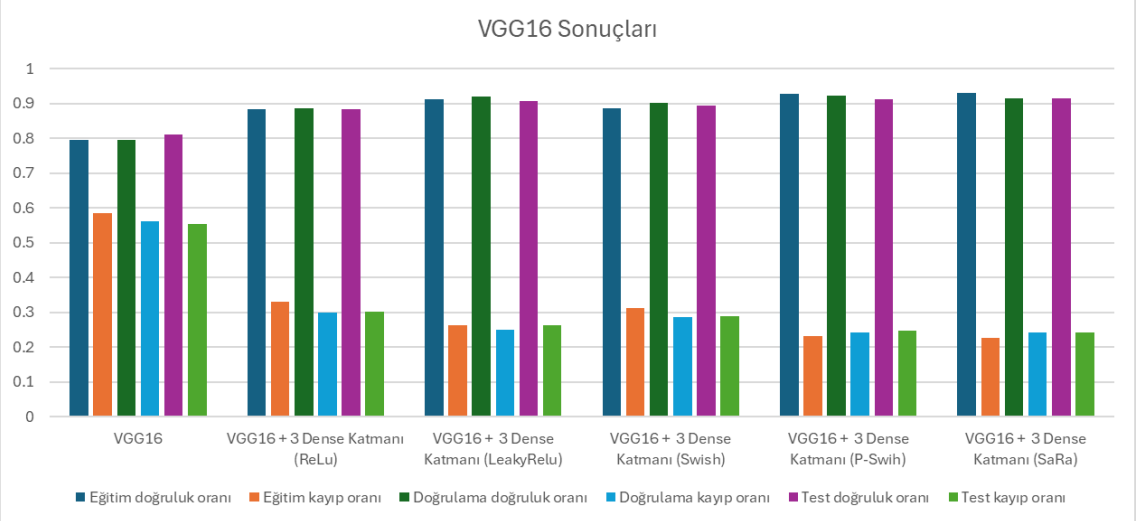
Yoğun katmanların modele eklenmesiyle birlikte performansta belirli düzeyde iyileşmeler gözlemlenmiştir. ReLU aktivasyon fonksiyonu kullanıldığında doğruluk ve kayıp değerlerinde bir miktar iyileşme sağlansa da modelin kararsız bir yapı sergilediği görülmektedir. Buna karşılık, LeakyReLU ve Swish aktivasyon fonksiyonları, daha yüksek doğruluk oranları ile birlikte daha istikrarlı bir eğitim süreci sunmuştur; ancak bu yapıların da optimum seviyeye ulaşamadığı tespit edilmiştir.

Özellikle P-Swish ve SaRa aktivasyon fonksiyonlarıyla, model hem daha yüksek doğruluk oranlarına (Çizelge 5.1) ulaşmış hem de kararlılık açısından daha tatmin edici sonuçlar vermiştir. Yapılan tüm konfigürasyonlar arasında, üç yoğun katman ve SaRa aktivasyon fonksiyonu kullanılan modelin en yüksek test doğruluğuna ulaştığı ve genel anlamda en iyi performansı gösterdiği belirlenmiştir.

Sonuç olarak, test doğruluğu açısından değerlendirildiğinde, Swish tabanlı aktivasyon fonksiyonlarının (Swish, P-Swish, SaRa), klasik ReLU ve LeakyReLU fonksiyonlarına göre daha üstün performans sergilediği görülmektedir (Şekil 5.1).

Çizelge 5. 1. VGG16 modelin sonuçları

	VGG16	VGG16 + 3 Dense Katmanı (ReLu)	VGG16 + 3 Dense Katmanı (LeakyRelu)	VGG16 + 3 Dense Katmanı (Swish)	VGG16 + 3 Dense Katmanı (P-Swih)	VGG16 + 3 Dense Katmanı (SaRa)
Eğitim doğruluk oranı	0.7946	0.8845	0.9108	0.8871	0.9287	0.9293
Eğitim kayıp oranı	0.5854	0.3293	0.264	0.3134	0.232	0.2255
Doğrulama doğruluk oranı	0.7948	0.8867	0.9197	0.9024	0.9211	0.9139
Doğrulama kayıp oranı	0.5621	0.2988	0.2504	0.2861	0.242	0.2428
Test doğruluk oranı	0.8102	0.8823	0.9071	0.894	0.9127	**0.9153**
Test kayıp oranı	0.5545	0.3022	0.2638	0.2898	0.2472	0.2426



Şekil 5.1. VGG16 mimarisnin sonuçları

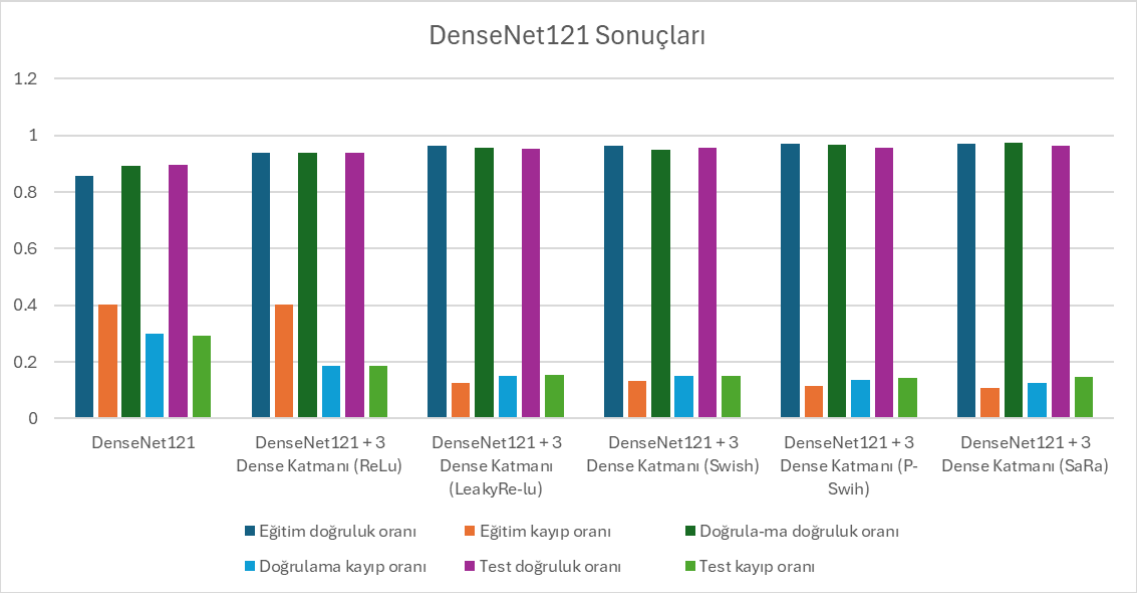
5.2. DenseNet121 Modelin Sonuçları

DenseNet121 mimarisine ait, ek yoğun (dense) katmanlar içermeyen temel model, düşük eğitim ve doğrulama doğrulukları sergilemiştir. Bu doğruluklar arasında, doğrulama doğruluğunun eğitim doğruluğundan bir miktar yüksek olması, modelin zayıf öğrenme gerçekleştirdiğini ve sınırlı genelleme yeteneğine sahip olduğunu göstermektedir. Bu durum, modelin düşük performansını ortaya koyan kayıp eğrileriyle de doğrulanmıştır.
Modele üç yoğun katman eklendikten sonra, genel performansta belirgin bir iyileşme gözlemlenmiştir. ReLU aktivasyon fonksiyonu ile doğruluk ve kayıp değerlerinde (Çizelge 5.2) belirli bir artış sağlansa da, modelin kararlılığının yetersiz kaldığı tespit edilmiştir. LeakyReLU aktivasyon fonksiyonu ise ReLU'ya kıyasla daha yüksek doğruluk ve daha istikrarlı bir eğitim süreci sunmuştur. Swish, P-Swish ve SaRa gibi daha gelişmiş aktivasyon fonksiyonlarının kullanılması durumunda, hem doğruluk oranlarında artış hem de modelin eğitim sürecinde daha yüksek kararlılık sağlanmıştır. Yapılan tüm denemeler arasında, üç yoğun katman içeren ve SaRa aktivasyon fonksiyonu kullanılan model, en yüksek test doğruluğuna ulaşmış ve en tutarlı performansı sergilemiştir.

Sonuç olarak, test doğruluğu açısından değerlendirildiğinde, Swish tabanlı aktivasyon fonksiyonlarının (Swish, P-Swish, SaRa), geleneksel ReLU ve LeakyReLU fonksiyonlarına kıyasla daha başarılı olduğu tespit edilmiştir (Şekil 5.2).

Çizelge 5. 2. DenseNet121modelin sonuçları

	DenseNet 121	DenseNet 121 + 3 Dense Katmanı (ReLu)	DenseNet 121 + 3 Dense Katmanı (LeakyRe-lu)	DenseNet 121 + 3 Dense Katmanı (Swish)	DenseNet 121 + 3 Dense Katmanı (P-Swih)	DenseNet 121 + 3 Dense Katmanı (SaRa)
Eğitim doğruluk oranı	0.8575	0.9376	0.9628	0.963	0.9712	0.9704
Eğitim kayıp oranı	0.4024	0.4024	0.1262	0.1315	0.1155	0.1077
Doğrula-ma doğruluk oranı	0.891	0.9397	0.957	0.9484	0.9656	0.9742
Doğrulama kayıp oranı	0.2995	0.1866	0.1506	0.1508	0.1357	0.1268
Test doğruluk oranı	0.8975	0.9396	0.9539	0.9577	0.9568	**0.9628**
Test kayıp oranı	0.2909	0.1845	0.1547	0.1516	0.1418	0.146



Şekil 5.2. DenseNet121 mimarisinin sonuçları

5.3. Inception-V3 Modelin Sonuçları

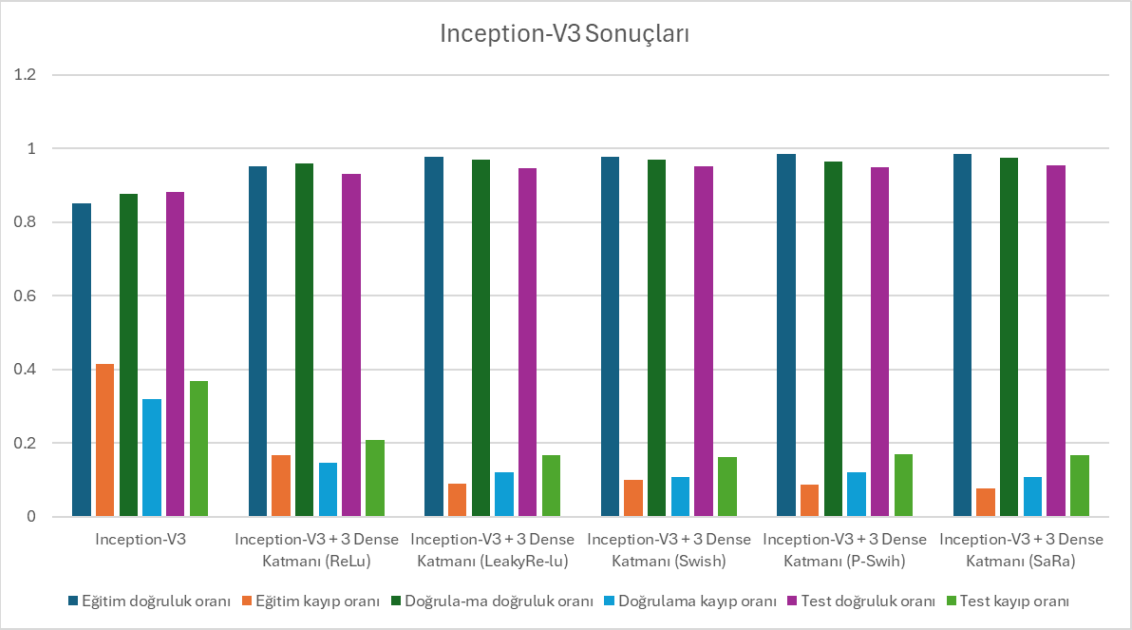
Inception-V3 mimarisine ait, ek yoğun (dense) katmanlar içermeyen temel model, düşük eğitim ve doğrulama doğrulukları göstermiştir. Doğrulama doğruluğunun eğitim doğruluğundan bir miktar yüksek olması, modelin zayıf öğrenme gerçekleştirdiğini ve genelleme kapasitesinin sınırlı olduğunu ortaya koymaktadır. Bu durum, modelin yetersiz performansını yansıtan kayıp eğrileri ile de desteklenmiştir.

Yoğun katmanların modele eklenmesiyle birlikte performansta gözle görülür bir iyileşme sağlanmıştır. ReLU aktivasyon fonksiyonu kullanıldığında doğruluk değerleri (Çizelge 5.3) artmış ve eğitim süreci daha kararlı hâle gelmiştir; ancak bu yapı yine de optimum performansı sağlayamamıştır. LeakyReLU, Swish, P-Swish ve SaRa gibi gelişmiş aktivasyon fonksiyonlarının kullanıldığı konfigürasyonlarda ise hem daha yüksek doğruluk değerleri elde edilmiş hem de modelin eğitim sürecinde kararlılığın arttığı gözlemlenmiştir.

Uygulanan tüm konfigürasyonlar arasında, üç yoğun katman ve SaRa aktivasyon fonksiyonu içeren model en yüksek test doğruluğuna ulaşmış ve genel olarak en iyi performansı sergilemiştir. Ayrıca, test doğruluğu açısından değerlendirildiğinde, Swish tabanlı aktivasyon fonksiyonlarının (Swish, P-Swish, SaRa), ReLU ve LeakyReLU'ya kıyasla daha üstün performans gösterdiği tespit edilmiştir (Şekil 5.3).

Çizelge 5. 3. Inception-V3 modelin sonuçları

	Inception -V3	Inception -V3 + 3 Dense Katmanı (ReLu)	Inception-V3 + 3 Dense Katmanı (LeakyRe -lu)	Inception -V3 + 3 Dense Katmanı (Swish)	Inception -V3 + 3 Dense Katmanı (P-Swih)	Inception -V3 + 3 Dense Katmanı (SaRa)
Eğitim doğruluk oranı	0.8513	0.9523	0.9775	0.9767	0.9846	0.9849
Eğitim kayıp oranı	0.4137	0.1657	0.0885	0.0993	0.0853	0.0769
Doğrula-ma doğruluk oranı	0.8766	0.9598	0.9684	0.9684	0.9641	0.9742
Doğrulam a kayıp oranı	0.3188	0.1463	0.1189	0.1068	0.1192	0.1085
Test doğruluk oranı	0.8814	0.9298	0.9466	0.9501	0.9482	0.9546
Test kayıp oranı	0.368	0.2074	0.1678	0.1617	0.1693	0.1653



Şekil 5.3. Inception-V3 mimarisinin sonuçları

5.4. Sonuç

VGG16, DenseNet121 ve Inception-V3 mimarilerinde çıkış katmanından önce üç yoğun (dense) katman eklenmesi, kullanılan tüm aktivasyon fonksiyonlarıyla birlikte modellerin performansını genel olarak iyileştirmiştir. Ancak Swish tabanlı aktivasyon fonksiyonları tanımlamış olduğumuz beyin tümörü sınıflandırma probleminde ReLU ve LeakyReLU gibi geleneksel fonksiyonlara kıyasla daha fazla performans artışı sağlamıştır.

VGG16 modeli özelinde, üç yoğun katman eklenmesi modelin kararlılığını artırmış; ancak ReLU, LeakyReLU ve Swish aktivasyon fonksiyonları ile elde edilen test doğruluk oranları sırasıyla %88.23, %90.71 ve %89.40 olarak gerçekleşmiştir. Bu oranlar, temel modelin %81.02 olan test doğruluk oranına kıyasla önemli bir artış sağlasa da, modelin kararlılığı açısından yeterli düzeyde iyileştirme sunduğu söylenemez.
Öte yandan, P-Swish ve SaRa aktivasyon fonksiyonları ile birlikte üç yoğun katman eklendiğinde model hem daha kararlı hâle gelmiş hem de test doğruluk oranlarında daha yüksek sonuçlar elde edilmiştir. Bu yapılandırmalarla test doğruluk oranları sırasıyla %91.27 ve %91.53 olarak gerçekleşmiştir.

Özetle, VGG16 modeli için çıkış katmanından önce SaRa aktivasyon fonksiyonu ile üç yoğun katman eklenmesi, modeli hem kararlı hem de kullanılabilir hâle getirmiştir. Bu yapı, temel modele kıyasla yaklaşık %10'luk bir doğruluk artışı sağlamış; LeakyReLU + 3 yoğun katman konfigürasyonuna kıyasla ise yaklaşık %3'lük bir fark yaratmıştır.

DenseNet121 modeli özelinde, üç yoğun katman eklenmesi modelin kararlılığını artırmış; ancak ReLU aktivasyon fonksiyonu ile elde edilen test doğruluk oranı sırasıyla %93.96 olarak gerçekleşmiştir. Bu oran, temel modelin %89.75 olan test doğruluk oranına kıyasla önemli bir artış sağlasa da, modelin kararlılığı açısından yeterli düzeyde iyileştirme sunduğu söylenemez.

Öte yandan, Leaky-ReLu, Swish, P-Swish ve SaRa aktivasyon fonksiyonları ile birlikte üç yoğun katman eklendiğinde model hem daha kararlı hâle gelmiş hem de test doğruluk oranlarında daha yüksek sonuçlar elde edilmiştir. Bu yapılandırmalarla test doğruluk oranları sırasıyla %95.39 , %95.77, %95.68 ve %96.28 olarak gerçekleşmiştir.

Özetle, DenseNet121 modeli için çıkış katmanından önce SaRa aktivasyon fonksiyonu ile üç yoğun katman eklenmesi, modeli hem kararlı hem de kullanılabilir hâle getirmiştir. Bu yapı, temel modele kıyasla yaklaşık %6'lık bir doğruluk artışı sağlamış; LeakyReLU + 3 yoğun katman konfigürasyonuna kıyasla ise yaklaşık %1'lik bir fark yaratmıştır.

Inception-V3 modeli özelinde, üç yoğun katman eklenmesi modelin kararlılığını artırmış; ancak ReLU aktivasyon fonksiyonu ile elde edilen test doğruluk oranı %92.98 olarak gerçekleşmiştir. Bu oran, temel modelin %88.14 olan test doğruluk oranına kıyasla önemli bir artış sağlasa da, modelin kararlılığı açısından yeterli düzeyde iyileştirme sunduğu söylenemez.
Öte yandan, Leaky-ReLu, Swish, P-Swish ve SaRa aktivasyon fonksiyonları ile birlikte üç yoğun katman eklendiğinde model hem daha kararlı hâle gelmiş hem de test doğruluk oranlarında daha yüksek sonuçlar elde edilmiştir. Bu yapılandırmalarla test doğruluk oranları sırasıyla %94.66, %95.01, %94.82 ve %95.46 olarak gerçekleşmiştir.

Özetle, Inception-V3 modeli için çıkış katmanından önce SaRa aktivasyon fonksiyonu ile üç yoğun katman eklenmesi, modeli hem kararlı hem de kullanılabilir hâle getirmiştir. Bu yapı, temel modele kıyasla yaklaşık %7'lik bir doğruluk artışı sağlamış; LeakyReLU + 3 yoğun katman konfigürasyonuna kıyasla ise yaklaşık %1'lık bir fark yaratmıştır.

Her üç modelde de (VGG16, DenseNet121 ve Inception-V3) çıkış katmanından önce eklenen yoğun (dense) katmanlar, model doğruluğunu ve kararlılığını önemli ölçüde artırmıştır. ReLU ve LeakyReLU aktivasyon fonksiyonları orta düzeyde performans artışı sağlarken, Swish tabanlı fonksiyonlar (Swish, P-Swish ve SaRa) genellikle daha yüksek test doğruluğu ve daha istikrarlı eğitim süreçleri sunmuştur. Bu fonksiyonlar arasında SaRa, en yüksek test doğruluğunu elde ederek en güçlü genel performansı göstermiştir. Sonuç olarak, Swish tabanlı aktivasyon fonksiyonlarının ek yoğun katmanlarla birlikte kullanılması, farklı mimarilerde modelin genelleme yeteneğini ve güvenilirliğini artırmada oldukça etkili olmuştur.

KAYNAKLAR

Goodfellow, I., Bengio, Y. ve Courville, A. (2016). Deep Learning. *Deep Learning by Ian Goodfellow | Goodreads*.

Kukačka, J., Golkov, V. Ve Cremers, D. (2017). Reg-ularization for deep learning: A taxonomy. ArXiv

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. ve Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, vol. 15, no. 56, pp. 1929-1958.

Prechelt, L. (2012). Early Stopping But When?. Springer Berlin Heidelberg.

Yu, T. ve Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. ArXiv.

Shorten, C. ve Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, vol. 6, no. 1, pp. 60.

Zhang, H., Cisse, M., Dauphin, Y. N. ve Lopez-Paz, D. (2018). Mixup: Beyond empirical risk minimization. International Conference on Learning Representations.

Li, Z. ve Arora, S. (2019). An Exponential Learning Rate Schadule For Deep Learning. ArXiv. *1910.07454*

Kalra, D. S. ve Barkeshli, M. (2024). Why Warmup the Learning Rate? Underlying Mechanisms and Improvements. ArXiv. *2406.09405*

Ramachandran, P., Zoph, B. ve Le, Q. V. (2017). Swish: a Self-Gated Activation Function. ResearchGate, 12. DOI*: 10.48550/arXiv.1710.05941*,

Mercioni, M. A. ve Holban, S. (2020). P-Swish: Activation Function with learnable Parameters Based on Swish Activation Function in Deep Learning. *2020 International Symposium on Electronics and Telecommunications (ISETC)*, DOI: *10.1109/ISETC50328.2020.9301059.*

Qureshi, M. N. ve Mohammad Sarosh Umar, M. S. (2022). SaRa: A Novel Activation Function with Application to Melanoma Image Classification. *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS).* DOI: *10.1109/ICACRS55517.2022.10029161.*

Huang, G., Liu, Z., Maaten, L. ve Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition. DOI: *10.1109/CVPR.2017.243*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. ve Rabinovich, A. (2015). Going Deeper with Convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* DOI: *10.1109/CVPR.2015.7298594*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. ve Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: *10.1109/CVPR.2016.308.*

Ioffe, S. ve Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. 32nd International Conference on Machine Learning.

Russell, S. Ve Norvig, P. (2003). Artificial Intelligence A Modern Approach.

Zurada, J. M. (1992). Introduction to Artificial Neural Systems.

Simonyan, K. ve Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. ResearchGate. *https://doi.org/10.48550/arXiv.1409.1556.*

Gangadia, D. (2021). Activation Functions: Experimentation and Comparison. *6th International Conference for Convergence in Technology (I2CT)*. DOI: *10.1109/I2CT51068.2021.9417890.*

Khadim, E. U., Shah, S. A. ve Wagan, R. A. (2021). Evaluation of Activation Functions in CNN Model for Detection of Malaria Parasite using Blood Smear Images. 2021 International Conference on Innovative Computing (ICIC). DOI: *10.1109/ICIC53490.2021.9693056.*

Shatravin, S., Shashev, D. ve Shidlovskiy, S. (2023). Implementation of the SoftMax Activation for Reconfigurable Neural Network Hardware Accelerators. applied science. *https://www.mdpi.com/2076-3417/13/23/12784*.

Bai, Y. (2022). RELU-Function and Derived Function Review. ResearchGate. DOI:*10.1051/shsconf/202214402006*.

Sunkari, S., Sangam, A., Sreeram, V. P., Suchetha, M., Raman, R., Rajalakshmi, R. ve Tamilselvi, S. (2023). A refined ResNet18 architecture with Swish activation function for Diabetic Retinopathy classification. ScienceDirect. *https://doi.org/10.1016/j.bspc.2023.105630.*

Anumol, C. S. (2023). Advancements in CNN Architectures for Computer Vision: A Comprehensive Review. *2023 Annual International Conference on Emerging Research Areas: International Conference on Intelligent Systems (AICERA/ICIS)*, IEEE. DOI: *10.1109/AICERA/ICIS59538.2023.10420413.*

Tsai, C. F., Jinsakul, N. ve Wu, P. ile Tsai, C. E. (2019). Enhancement of Deep Learning in Image Classification Performance Using Xception with the Swish Activation Function for Colorectal Polyp Preliminary Screening. Mathematics, 7(12), 1170. *https://doi.org/10.3390/math7121170.*

Nilsson, N. J. (1998). Introduction To Machine Learning. Robotics Laboratory, Department of Computer Science, Stanford University Stanford, CA 94305.

Javid, A., Das, S., Skoglund, M. ve Chatterjee, S. (2021). A ReLU Dense Layer to Improve the Performance of Neural Networks. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE. DOI*: 10.1109/ICASSP39728.2021.9414269.*

Sun, Y. (2021). The role of activation function in image classification. *2021 International Conference on Communications, Information System and Computer Engineering (CISCE),* IEEE. DOI*: 10.1109/CISCE52179.2021.9445868*.

Saeedi1, S., Rezayi1, S., Keshavarz, H. ve Kalhori, S.R.N. (2023). MRI-based Brain Tumor Detection Using Convolutional Deep Learning Methods and Chosen Machine Learning Techniques. BMC Medical Informatics and Decision Making. *https://doi.org/10.1186/s12911-023-02114-6.*

Gupta, B. B., Gaurav, A. ve Arya, V. (2024). Deep CNN Based Brain Tumor Detection in Intelligent Systems. International Journal of Intelligent Networks 5, 30–37, ScienceDirect. *https://doi.org/10.1016/j.ijin.2023.12.001.*

Zingaretti, L., Gezan, S. A., Ferrao, L. F. V. ve Osorio, L. F. (2020). Exploring Deep Learning for Complex Trait Genomic Prediction in Polyploid Outcrossing Species. ResearchGate. DOI: *10.3389/fpls.2020.00025*

Prasanth, M. Demystifying VGG16: A Powerful Tool in Your Deep Learning Arsenal. LinkedIn. Nisan 5, 2024. *Demystifying VGG16: A Powerful Tool in Your Deep Learning Arsenal.*