



**T.C.  
KASTAMONU ÜNİVERSİTESİ  
MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME PROJESİ II**

**KONU**

Optik Koherans Kullanarak Göz Hastalıkları Tespitinde Derin Öğrenme Yaklaşımı

**HAZIRLAYAN**

174410027 – Sema Nur KAYA

174410029 – Onur GÜL

174410035 – Bahri ÇÖKMEZ

**DANIŞMAN**

Öğr. Gör. Atilla SUNCAK

Haziran- 2021  
**KASTAMONU**

# İÇİNDEKİLER

ÖZ

ABSTRACT

ŞEKİL TABLOSU

1. GİRİŞ ve TANITIM	4
2. LİTERATÜR TARAMASI.....	5-7
3. MATERYAL ve YÖNTEM	
3.1. MATERYAL	
3.1.1. Veriseti .....	8-9
3.1.2. Google Colabratory .....	10
3.1.3. Python.....	10
3.1.4. Keras.....	11
3.1.5. Tensorflow .....	11
3.1.6. ImageDatagenerator .....	12
3.2.YÖNTEM	
3.2.1. EfficientNet5 .....	12
3.2.2. CNN(Konvolüsyonel Sinir Ağları).....	13
3.2.3. InceptionV3 .....	14
4. UYGULAMA ve GELİŞTİRME .....	14-20
5. BULGULAR .....	20-23
6. SONUÇ ve ÖNERİLER.....	23-24
7. KAYNAKLAR .....	24-25

## ŞEKİL TABLOSU

### 1. LİTERATÜR TARAMASI

- 1.1. ARIAS yönteminde kullanılan mdel ..... 4
- 1.2. Retina görüntü analizinde kullanılan Lowa Detection Program ..... 5
- 1.3. Önerilen VGG-16 mimarisi ..... 6

### 2. MATERYAL ve YÖNTEM

#### 2.1. MATERYAL

- 2.1.1. Sınıflara göre kullanılan Optik Koherans Tomografisi görüntüleri ..... 8
- 2.1.2. Kullanılan Google Colabratory görseli ..... 9
- 2.1.3. Kullanılan Tensorflow görseli ..... 10
- 2.1.4. Kullanılmış olan Keras kütüphane görseli ..... 10

#### 2.2. YÖNTEM

- 2.2.1. EfficientNet5 model katman mimarisi ..... 11
- 2.2.2. InceptionV3 model katman mimarisi ..... 12
- 2.2.3. CNN model mimarisi ..... 13

### 3. UYGULAMA ve GELİŞTİRME

- 3.1. Şekil 4.2. ve Şekil 4.2.1. EfficientNet5 eğitim sonuçları ..... 14
- 3.2. Şekil 4.2.3. ve Şekil 4.3.1. InceptionV3 eğitim sonuçları ..... 15
- 3.2. Şekil 4.4., Şekil 4.4.1, Şekil 4.4.2, Şekil 4.4.3, Şekil 4.4.4, Şekil 4.4.5, CNN eğitim sonuçları ..... 16-18

### 4. BULGULAR

- 4.1. Şekil 5.1. EfficientNet5 model adam optimizer başarı-loss grafikleri ..... 19
- 4.2. Şekil 5.2. EfficientNet5 model adam optimizer confusion matrix grafiği ..... 20
- 4.3. Şekil 5.3. InceptionV3 model adam optimizer başarı-loss grafiği ..... 20
- 4.4. Şekil 5.4. InceptionV3 model adam optimizer confusion matrix grafiği ..... 21
- 4.5. Şekil 5.5. CNN model adam optimizer başarı-loss grafiği ..... 21
- 4.6. Şekil 5.4. CNN model adam optimizer confusion matrix grafiği ..... 22

## ÖZ

Tıbbi görüntüleme için klinik karar destek algoritmalarının uygulanması, güvenilirlik ve yorumlanması ile ilgili zorluklarla karşı karşıyadır. Burada, yaygın olarak tedavi edilebilir kör edici retina hastalıkları olan hastaların taranması için derin öğrenme çerçevesine dayalı bir teşhis aracı oluşturulmaktadır. Çerçevemiz, geleneksel yaklaşımların verilerinin bir kısmı ile bir sinir ağını eğiten transfer öğrenimini kullanmaktadır. Bu yaklaşımı optik koherens tomografi görüntülerinden oluşan bir veri kümesine uygulayarak, yaşa bağlı makula dejenerasyonu ve diyabetik makula ödemi sınıflandırmada insan uzmanların performansı ile karşılaştırılabilir bir performans oluşturulmaktadır. Ayrıca sinir ağı tarafından tanınan bölgeleri vurgulayarak daha şeffaf ve yorumlanabilir bir teşhis sağlanmaktadır. Kullanılmış olan derin öğrenme modelleri sonucunda tedavi edilebilir durumların teşhisini ve sevkini hızlandırmaya yardımcı olabilir, böylece daha erken tedaviyi kolaylaştırarak daha iyi klinik sonuçlarla sonuçlandırılmaktadır. Günümüzde popüler olan evrişimli sinir ağları tabanlı farklı tekniklerin aynı veri kümesi üzerinde deneysel çalışmaları yapılarak her bir tekniğin performans değerlendirilmesi yapılmış ve en iyi tahminleme yöntemi belirlenmiştir. Yapılan deneysel çalışmalarda, en yüksek doğruluk değeri %98.74 ile CNN modeli ile elde edilmiştir. Anahtar Kelimeler: Derin öğrenme, Evrişimli sinir ağları, Sınıflandırma, Optik Koherans Tomografisi(OCT), Makula.

## ABSTRACT

The application, reliability, and interpretation of clinical decision support algorithms for medical imaging face challenges. Here we build a diagnostic tool based on a deep learning framework for screening patients with commonly treatable blinding retinal diseases. Our framework uses transfer learning, which trains a neural network with some of the data from traditional approaches. By applying this approach to a dataset of optical coherence tomography images, performance comparable to that of human experts in classifying age-related macular degeneration and diabetic macular edema is established. It also highlights regions recognized by the neural network, providing a more transparent and interpretable diagnosis. The deep learning models that have been used can help speed the diagnosis and referral of treatable conditions as a result, thus facilitating earlier treatment resulting in better clinical outcomes. Experimental studies of different techniques based on convolutional neural networks, which are popular today, were performed on the same dataset, performance evaluation of each technique was made and the best estimation method was determined. In the experimental studies, the highest accuracy value was obtained with the CNN model with 98.74%. Keywords: Deep learning, Convolutional neural networks, Classification, Optical Coherence Tomography(OCT), Macula.

## 1. GİRİŞ

Vücudumuzda en önemli duyu organlarımızdan biri de gözdür. Göz hastalıkları, çevresel ya da genetik faktörlere bağlı olarak ortaya çıkar. Bu faktörler çeşitli görme problemlerine yol açabilen fiziksel rahatsızlıklardır. Göz merceğinde ya da gözü saran sinir dokularında meydana gelen her türlü hastalık göz hastalıkları olarak adlandırılmaktadır.

Göz hastalıklarının tedavisinde ve teşhisinde doktorlar büyük rol oynamaktadır. Teknolojinin gelişmesiyle tedavi ve teşhis için bir çok cihaz ve tomografi sistemi geliştirilmiştir. Göz hastalıklarının teşhisinde en yaygın kullanılan tomografi sistemi OCT (Optik Koherens Tomografi)'dir. İlk olarak 1991'de Massachusetts Teknoloji Enstitüsünde Prof. James Fujimoto laboratuvarında Huang ve arkadaşları tarafından geliştirilen OCT, mikrometre çözünürlüğü ve enine kesitsel görüntüleme yetenekleri ile belirgin bir biyomedikal doku görüntüleme tekniği haline gelmiştir (şahin, 2016).

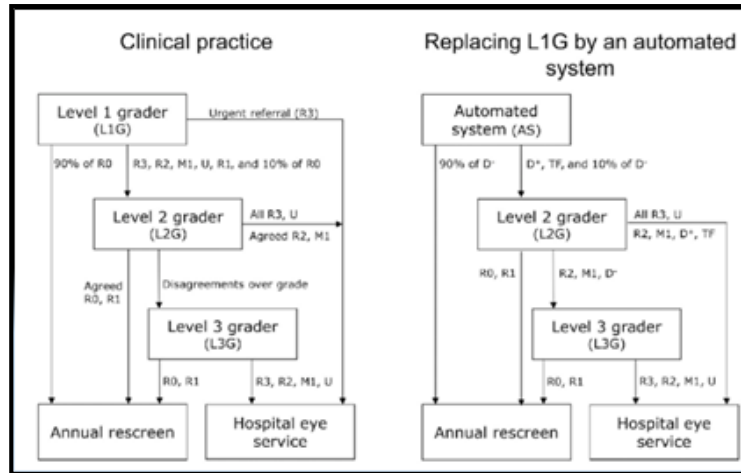
Yapay zeka ve derin öğrenme algoritmaları son yıllarda tıp alanında çeşitli çalışmalarla karşımıza gelmektedir. Optik Koherens Tomografi görüntülemeleri ile diyabetik retinopati, yaşa bağlı maküla dejenerasyonu, glokom, maküler ödem, optik nöropati, santral seröz koryoretinopati ve retina segmentasyonu gibi konularda derin öğrenme algoritmaları üzerine çalışılmaktadır (Yaya,2015). Erken tanı ve doğru zamanda yapılacak hasta yönlendirmesi ile insanların yaşam kalitesinin artırılması ve önlenabilir körlüğün önüne geçilmesi hedeflenmektedir . Bu çalışmada yapay zeka, makine öğrenmesi ve derin öğrenme sistemleri uygulama modelleri üzerine gelişmeler özetlenecektir.

Bu Projede OCT tomografileri tanıma işlemindeki başarıları incelenmiştir. Bu projede; CNN,Inceptionv3 ve EffNet5 yaklaşımları kullanılmaktadır. Öncelikle farklı göz tomografilerini belirlenen resim kümesinden öğrenme işlemini gerçekleştirmesi beklenmektedir. Makinanın öğrenmesini sağlayacak olan resim veriseti kümesi test ve train olarak iki gruba ayrılmıştır. Eğitim işlemi gerçekleştirildikten sonra program "yeniveriseti" adlı klasörden aldığı görüntüyü değerlendirerek daha önceden eğitilmiş olan verilerden seçmiş olduğumuz veriyi kendince eğitime sokma işlemi yaparak tahmin etme işlemini gerçekleştirmektedir.

## 2. LİTERATÜR TARAMASI

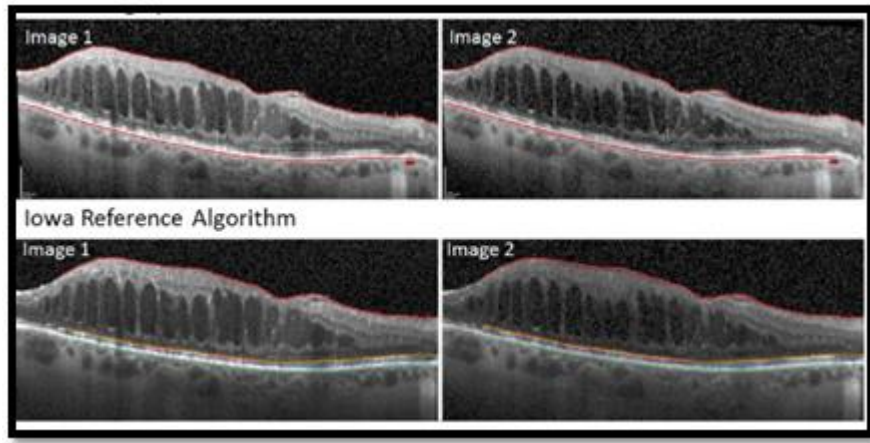
Retina fundus fotoğraflarında mikroanevrizma, hemoraji, eksuda, pamuk yün nokta bulguları otomatize sistemlere eğitilerek derin öğrenme konusunda çalışmalar yapılmıştır. DR tespiti için Gulshan ve arkadaşları retina fotoğrafları ile derin öğrenme algoritmaları üzerine çalışan ilk isimlerdir. İki tabanlı sistemdeki 9963 ve 1748 görüntüyle, orta şiddet veya kötü durum DR ve ön görülebilir diyabetik maküler ödem (DMÖ) üzerine çalışmışlardır. Daha sonradan Ting ve ark. ise DR, YBMD ve glokom üzerine ilişkili göz hastalıkları üzerine derin öğrenme sistemleri kurmuşlardır. Son yapılan çalışmalar OCT görüntüleme üzerine olmaktadır. Kermany ve ark. DMÖ ve YBMD için tarama programı hedefleyerek derin öğrenme modeli geliştirmişlerdir (Karaküçük,2014).

Tufail ve arkadaşları otomatize olan DR görüntüleme değerlendirme sistemleri (ARIAS) ile insanların taramasının yerine geçmesini hedefleyen çalışma yapmışlardır. DR, ön görülen DR, ve proliferatif DR sensitiviteğini EyeArt için sırayla %94,7, %93,8 ve %99,6 olarak saptamış olup, Retmarker için %73, %85, ve %97,9 tespit etmişlerdir. Kabul edilebilir düzeylerde olan bu sensitivite değerleri ile ARIAS modelinin DR'yi erken saptamada insan pratiğinin yerine geçebileceği bildirilmiştir. Bu DR tarama modelinin uzak sağlık hizmetleri bünyesinde kullanılabileceğini öngörülmektedir (Tufail ve arkadaşları,2014).



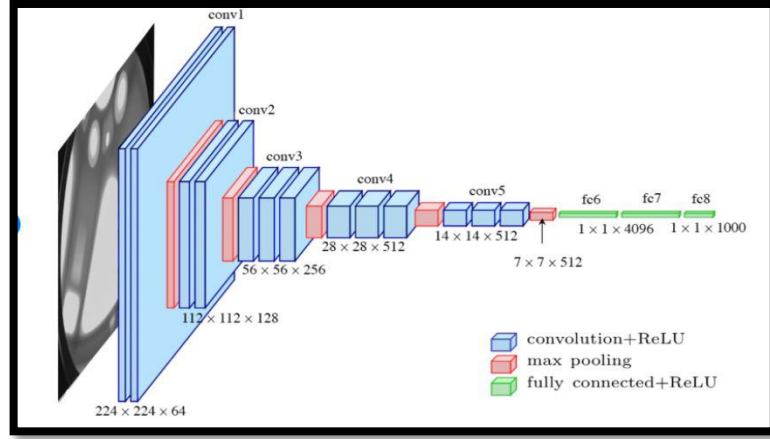
Şekil 2.1: ARIAS yönteminde kullanılan model.

Abramoff ve arkadaşları yaptığı çalışmada ön görülebilir DR'yi saptamak için retina görüntülerinin analizine dayanarak 'Iowa Detection Program' üzerine çalışmışlardır. Bu sistemin ön görülebilen DR' yi saptamada yüksek sensitivite ve spesifisite değerlerine sahip olduğu ve bu modelin DR erken tanısında kullanılabileceği ve görme keskinliğinde azalmanın önüne geçilebileceği vurgulanmıştır. Yaptıkları çalışma, Nisan 2018'de yapay zeka sistemi olarak IDx-DR, FDA tarafından onaylanan ve pazarlamasına başlanılan ilk sistem olarak karşımıza çıkmaktadır (Abramoff ve arkadaşları,2014).



Şekil 2.2: Retina görüntü analizinde kullanılan Iowa DetectionProgram.

Li ve arkadaşları OCT ile alınmış 207.130 görüntüyü kullanarak visual geometry group 16 (VGG-16) sinir ağı modelinde, YBMD ve DMÖ sınıflandırmasında %98,6 sensitivite veren başarılı sonuçlar elde etmişlerdir. Bu sistemin, DR taramalarında kullanılabilecek yüksek duyarlılığa sahip güvenilir bir araç olduğunu bildirmişlerdir (Li ve arkadaşlar,2014).



Şekil 2.3: Önerilen VGG-16 mimarisi.

Mehmet Hakan ÖZDEMİR ve Fevzi ŞENTÜRK'ün birlikte hazırlamış olduğu çalışma Okülokutanöz albinizmde fovea hipoplazisinin spektral domain optik koherens tomografi ile gösterilmesini gerçekleştirmişlerdir. Bu çalışmada Okülokutanöz albinizmi olan 15 yaşındaki kız çocuğunun spektral domain optik koherens tomografi (SD-OKT) ile fovea bölgesinde çok sayıda retina kesitleri alınmıştır. SD-OKT kesitlerinde fovea çukurluğunun saptanamadığı görülmüştür. Foveayı içine alan tüm retina kesitlerinde retina kalınlığının foveada hiç değişmeden devam ettiği ve etrafındaki maküla ile aynı olduğu gözlemlendi. Bu gözdeki fovea kalınlığı  $330 \mu$  idi. Okülokutanöz albinizm olanlarda fovea hipoplazisi SD-OKT ile görüntülenebilmektedir (Özdemir ve Şentürk, 2011).



### 3. MATERYAL ve YÖNTEM

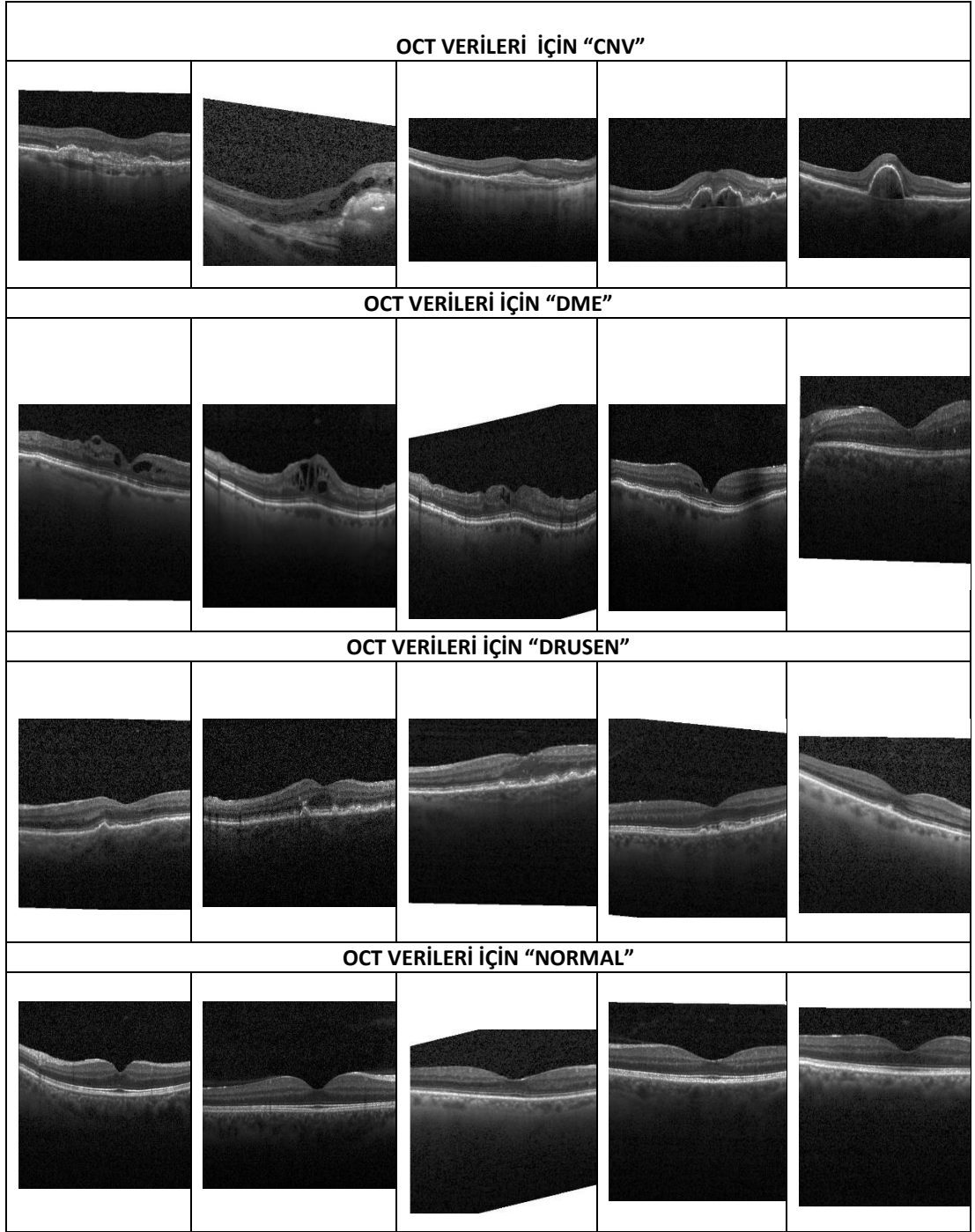
#### 3.1. Materyal

##### 3.1.1. Veri Seti

Bu çalışmada kullanılmış olan veri seti en güncel ve verimli olacak şekilde seçilmiştir. Kullanılan veri setindeki görüntülerin sınıflandırılmasında 3 ayrı göz uzmanı tarafından yararlanıldığı için verilerin sınıflandırılmasındaki hata oranı çok azdır, bundan dolayı da ünlü bir veri setidir. Kullanılan veri setinin adı; Sınıflandırma İçin Etiketli Optik Koherens Tomografi(OCT) olarak adlandırılmaktadır (Mendeleyev,2021). Geliştirilecek olan çalışma için gerekli incelemelerin yapılmış olup; Google Colaboratory ortamı kullanılmıştır. Veri seti test ve train ayırma işlemi yapıp klasör haline getirildikten sonra başarıları çalışmaya dahil edilmiştir. Veri setindeki görüntüler derin öğrenme modellerini kullanmış olup sonrasında da modellerin başarılarını karşılaştırma işlemi yapılmıştır.

Tablo 1: Sınıflar ve veri sayıları

Sınıf Adı	Eğitim (%87.5)	Test (%12.5)	Toplam
CNV	2000	250	2250
DME	2000	250	2250
DRUSEN	2000	250	2250
NORMAL	2000	250	2250



Şekil 3.1: Sınıflara göre kullanılan optik koherans tomografisi görüntüleri

### 3.1.2. Colaboratory

Colaboratory uygulaması google'ın tarayıcımızda python kodlarını bir not defteri aracılığıyla yazıp çalıştırabildiğimiz bir uygulamadır. GPU'ya bağlanma imkanı sağlar, ücretsizdir. Veri setini kullanmak için kodlarımızı yazdığımız not defterine drive bağlamamız sonucunda uygulamamızı gerçekleştirebileceğimiz bir ortamdır. Herhangi bir yapılandırmaya ihtiyaç duymaz, kolay paylaşım imkanı sunar ve GPU'lara ücretsiz erişim imkanı sağlamaktadır. GPU'lar ve TPU'lar gibi güçlü donanım seçenekleri üzerinde makine öğrenimi modellerinin prototipini oluşturmaya yönelik bir araştırma projesidir (Bisong,2019). Etkileşimli geliştirme için sunucusuz bir Jupyter not defteri ortamı sağlamaktadır.



Şekil 3.1.2. : Kullanılan Colaboratory görseli

### 3.1.3. Python

Python, ilk olarak 1990'lı yılların başında Guido Van Rossum tarafından geliştirilmeye başlanmıştır. Python'un ortaya çıkışında ve geliştirilme sürecinde, bir başka genel amaçlı yazılım dili olan ABC programlama dilinin etkisi olmuştur (Dierbach, 2012). Van Rossum, Centrum Wiskunde & Informatica (CWI) isimli araştırma merkezinde, üzerinde çalıştığı ABC programlama dilinin sınırlı olduğunu, geliştirmeye açık olmadığını fark ederek ve ABC dilinin olumlu yanlarını alarak Python'u ortaya çıkarmıştır (Ayvaz,2012).

Python, üst düzey basit sözdizimine sahip, öğrenmesi oldukça kolay, okunabilirliği destekleyen platformdan bağımsız nesne yönelimli yorumlanabilir bir dildir. Bu dil açık kaynak kodlu bir programlama dilidir, ücretsizdir, yüksek dinamik veri türleri ve sınıfları ile birlikte çalışmaktadır.

### 3.1.4. Kütüphaneler

#### 3.1.4.1. Tensorflow

TensorFlow Google şirketinin açık kaynak projesidir. 2015 yılının sonlarında halka duyurulmuş olmasına rağmen en popüler derin öğrenme kütüphanesi haline gelmiştir. TensorFlow, C++’da geliştirildi ve C++ ve Python API’leri mevcuttur. TensorFlow’da kullanılan veri akış grafiklerindeki düğümler matematiksel işlemleri temsil ederken, grafik kenarları aralarında akan çok boyutlu veri dizilerini (tensörler) temsil etmektedir. Bu esnek mimari, hesaplama kodunu yeniden yazmadan masaüstünde, sunucuda veya mobil cihazda bir veya daha fazla CPU’ya veya GPU’ya dağıtmanıza olanak tanımaktadır (Şişman,2019).



Şekil 3.1.4.1. : Kullanılan TensorFlow görseli

#### 3.1.4.2. Keras

Python'da yazılmış açık kaynaklı bir sinir ağı kütüphanesidir. Tensorflow ile birlikte kullanılmaktadır. Çalışmalarda derin öğrenme işlevleri eklemek isteyen geliştiriciler için Keras, uygulamaları daha hızlı bir şekilde geliştirmenize izin verdiği için iyi bir seçenektir ( Şişman,2019). Derin öğrenme algoritmalarını oluşturmak için uygun olmasının yanı sıra, CPU veya GPU’da çalıştırmak için aynı Python kodunu kullanması da ayrı bir avantajıdır.



Şekil 3.1.4.2: Kullanılmış olan Keras kütüphane görseli.

### 3.1.4.3. ImageDatagenerator

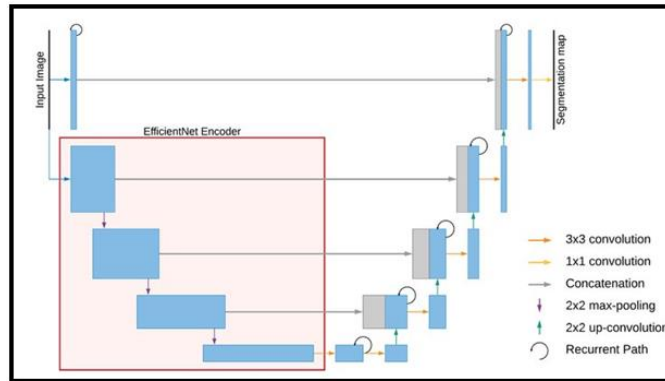
ImageDataGenerator, Keras'ın derin öğrenme için görüntü verilerinin ardışık düzenlenmesi için başvurduğu sınıftır. Yerel dosya sisteminize kolay erişim ve farklı yapılardan veri yüklemek için birden fazla farklı yöntem sağlar. Ayrıca oldukça güçlü veri ön işleme ve artırma yeteneklerine sahiptir.

## 3.2. Yöntem

Bu çalışmada Transfer Learning modellerinden CNN, EfficientNet5 ve IncetionV3 modelleri kullanılmıştır.

### 3.2.1. EfficientNet5

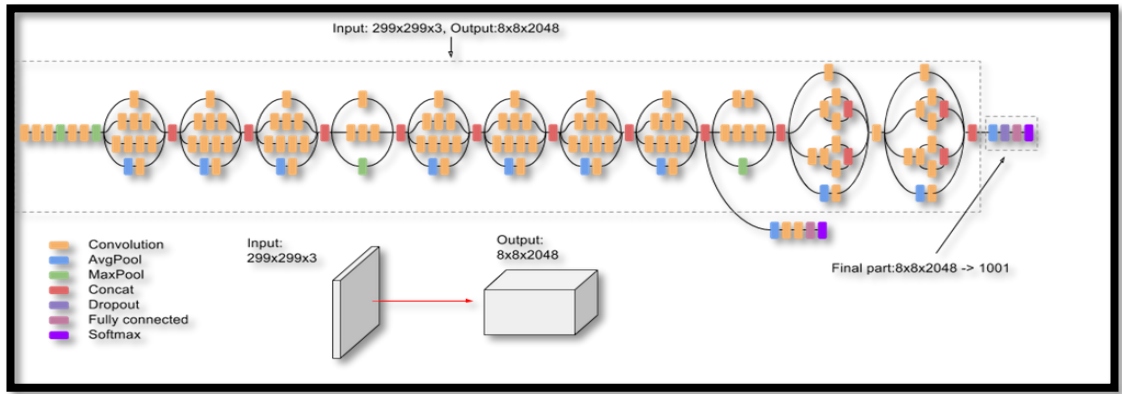
Google 2019'da yeni bir yapısal yaklaşıma sahip CNN ( Evrişimsel Sinir Ağları) hakkında ilgi çekici bir çalışma yayınlamıştır . Efficientnet; doğruluk , işlem verimliliği bileşik katsayı kullanarak tüm derinlik, genişlik, çözünürlük boyutlarını tek tip olarak ölçekleyen bir evrişimli sinir ağı mimarisi ve ölçekleme yöntemidir .EfficientNet, sadece doğruluğu değil, aynı zamanda modellerin verimliliğini de geliştirmeye odaklanmıştır. Çoğu modelden 5–10 kat daha verimli iken % 6'ya varan doğruluk artışı da sergilemektedir (Ömer Özgür,2021).



Şekil 3.2.1: EfficientNet5 model katman mimarisi

### 3.2.2. InceptionV3

InceptionV3, temel olarak önceki Inception mimarilerini değiştirerek daha az hesaplama gücünü ortaya çıkarmaya odaklanmıştır. InceptionV3 (Tang, May 2018), görüntü analizine ve nesne algılamaya yardımcı olmak için evrimsel bir sinir ağıdır ve GoogLeNet için bir modül olarak başlamıştır. İlk olarak ImageNet Tanıma Yarışması sırasında tanıtılan, Google'ın Inception Convolutional Neural Network'ün üçüncü baskısıdır. Tıpkı ImageNet sınıflandırılmış görsel nesneler veri tabanı olarak düşünülebilir, Inception nesnelerin sınıflandırılması yardımcı olmaktadır (Milton-Barker,2019).



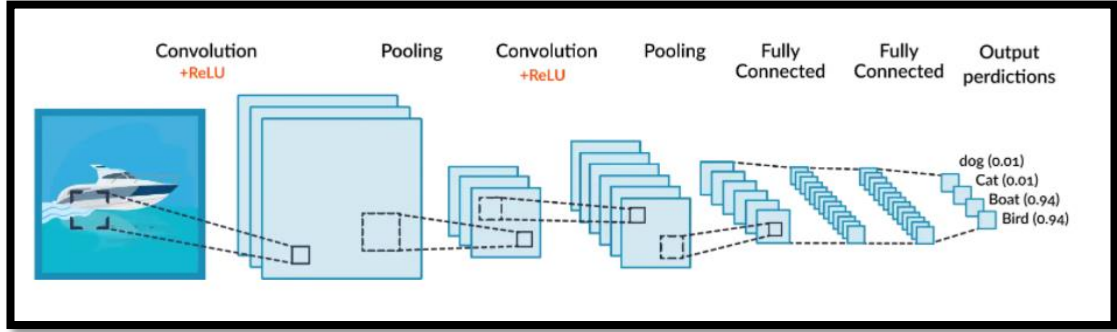
Şekil 3.2.2: InceptionV3 model katman mimarisi

### 3.2.3. CNN

CNN genellikle görüntü işlemede kullanılan ve girdi olarak görselleri alan bir derin öğrenme algoritmasıdır. Farklı operasyonlarla görsellerdeki özellikleri (featureları) yakalayan ve onları sınıflandıran bu algoritma farklı katmanlardan oluşmaktadır. Convolutional “Layer”, “Pooling” ve “Fully Connected” olan bu katmanlardan geçen görsel, farklı işlemlere tabii tutularak derin öğrenme modeline girmektedir. CNN modelleri oluştururken, düzensiz (unstructural) veri ile uğraşıldığından klasik makine öğrenmesi algoritmalarına kıyasla veri ön işleme kısmında çok uğraşmamaktadır (Özgür Doğan,2021).

Convolutional evrim katmanı(Layer), CNN algoritmalarında görüntüyü ele alan ilk katmandır. Bilindiği üzere görselle aslında içlerinde belirli değerler taşıyan

piksellerden oluşan matrislerdir. Evrişim katmanında da orijinal görsel boyutlarından daha küçük bir filtre görsellerden belirli özellikleri yakalamaya çalışmaktadır.



Şekil 3.2.3: CNN model mimarisi

#### 4. UYGULAMA VE GELİŞTİRME

Bu projede veri toplama işlemi yapılarak başlanmıştır. Kullanılan ver setindeki verilerde veri arttırımı (data augmentation) kullanılmıştır fakat; aşırı öğrenmeye(overfit) sebep açtığı için sonrasında kullanılmamıştır. Kullanılmış olan görüntüler yeterli ve düzgün olduğu için; yeniden boyutlandırma (resize) işleminin dışında herhangi bir işlem yapılmamıştır. Veri boyutlarını “160x160” olarak yeniden ölçeklendirme işlemleri gerçekleştirilmiştir.

Sınıf Adı	Epoch	Batch Size	Optimizer	Target Size	Loss
EfficientNet5	20	16	adam	160,160	Categorical_crossentropy
CNN	100	16	adam	160,160	Categorical_crossentropy
InceptionV3	20	16	adam	160,160	Categorical_crossentropy

Tablo 4.1: Kullanılan modeller ve veri etiketleme değerleri

Optik Koherens Tomografi (OCT) üç farklı Transfer Learning modeli kullanılarak gerçekleştirilmiştir. InceptionV3 modelinde yapılan bu çalışmada farklı optimizeler deneme yapılarak başarı anlamında hangi optimizelerin daha yüksek bir başarı elde edeceğini test edilmektedir. Adam optimizer kullanıldığında %97.83 gibi bir başarı elde edilmiştir fakat kullanılmış olan bu modelde aşırı öğrenme (overfitting)

yaşanmaktadır. Uygulanan dropout ve benzer yöntemler sayesinde aşırı öğrenme(overfitting) genel olarak önlenebilmektedir fakat bu çalışmada aşırı öğrenme (overfitting) önleme konusunda olumlu bir sonuç alınamamıştır.

Epoch 1/20	
201/201 [=====]	- 25s 90ms/step - loss: 1.2132 - accuracy: 0.5951 - val_loss: 0.9781 - val_accuracy: 0.7500
Epoch 2/20	
201/201 [=====]	- 17s 84ms/step - loss: 0.3766 - accuracy: 0.8846 - val_loss: 1.1779 - val_accuracy: 0.6467
Epoch 3/20	
201/201 [=====]	- 17s 84ms/step - loss: 0.3240 - accuracy: 0.9055 - val_loss: 0.1877 - val_accuracy: 0.9620
Epoch 4/20	
201/201 [=====]	- 17s 85ms/step - loss: 0.2042 - accuracy: 0.9439 - val_loss: 0.1021 - val_accuracy: 0.9565
Epoch 5/20	
201/201 [=====]	- 17s 84ms/step - loss: 0.1876 - accuracy: 0.9498 - val_loss: 0.1664 - val_accuracy: 0.9457
Epoch 6/20	
201/201 [=====]	- 17s 85ms/step - loss: 0.3054 - accuracy: 0.9050 - val_loss: 0.1445 - val_accuracy: 0.9728
Epoch 7/20	
201/201 [=====]	- 17s 85ms/step - loss: 0.1881 - accuracy: 0.9491 - val_loss: 0.0520 - val_accuracy: 0.9891
Epoch 8/20	
201/201 [=====]	- 17s 85ms/step - loss: 0.1142 - accuracy: 0.9688 - val_loss: 0.6538 - val_accuracy: 0.8859
Epoch 9/20	
201/201 [=====]	- 17s 85ms/step - loss: 0.0900 - accuracy: 0.9729 - val_loss: 0.1611 - val_accuracy: 0.9565
Epoch 10/20	
201/201 [=====]	- 17s 86ms/step - loss: 0.0907 - accuracy: 0.9699 - val_loss: 0.0385 - val_accuracy: 0.9946
Epoch 11/20	
201/201 [=====]	- 17s 85ms/step - loss: 0.0648 - accuracy: 0.9841 - val_loss: 0.2744 - val_accuracy: 0.9239
Epoch 12/20	
201/201 [=====]	- 17s 85ms/step - loss: 0.1076 - accuracy: 0.9678 - val_loss: 0.2352 - val_accuracy: 0.9185
Epoch 13/20	
201/201 [=====]	- 17s 86ms/step - loss: 0.1020 - accuracy: 0.9695 - val_loss: 0.2447 - val_accuracy: 0.9185
Epoch 14/20	
201/201 [=====]	- 17s 86ms/step - loss: 0.0749 - accuracy: 0.9760 - val_loss: 0.0876 - val_accuracy: 0.9674
Epoch 15/20	
201/201 [=====]	- 17s 85ms/step - loss: 0.0648 - accuracy: 0.9791 - val_loss: 0.1468 - val_accuracy: 0.9457
Epoch 16/20	
201/201 [=====]	- 17s 85ms/step - loss: 0.0623 - accuracy: 0.9842 - val_loss: 0.1760 - val_accuracy: 0.9348

Şekil 4.2.: EfficiNet5 eğitim sonuçları

Epoch 16/20	
201/201 [=====]	- 17s 85ms/step - loss: 0.0623 - accuracy: 0.9842 - val_loss: 0.1760 - val_accuracy: 0.9348
Epoch 17/20	
201/201 [=====]	- 17s 85ms/step - loss: 0.0739 - accuracy: 0.9793 - val_loss: 0.0602 - val_accuracy: 0.9891
Epoch 18/20	
201/201 [=====]	- 17s 84ms/step - loss: 0.0494 - accuracy: 0.9856 - val_loss: 0.4651 - val_accuracy: 0.8152
Epoch 19/20	
201/201 [=====]	- 17s 86ms/step - loss: 0.1219 - accuracy: 0.9647 - val_loss: 0.0484 - val_accuracy: 0.9728
Epoch 20/20	
201/201 [=====]	- 18s 87ms/step - loss: 0.0315 - accuracy: 0.9910 - val_loss: 0.0649 - val_accuracy: 0.9783

Şekil 4.2.1: EfficiNet5 eğitim sonuçları

InceptionV3 madeline adam optimizieri kullanıldığın zaman elde edilen sonuç %98.81 başarı edilmiştir. Fakat kullanılmış olan bu modelde aşırı öğrenme (overfitting) yaşanmaktadır. Uygulanan dropout ve benzeri yöntemler sayesinde aşırı öğrenme(overfitting) genel olarak önlenebilmektedir fakat bu çalışmada aşırı öğrenme (overfitting) önleme konusunda olumlu bir sonuç alınamamıştır.



```

Epoch 1/20
250/250 [=====] - 87s 347ms/step - loss: 0.5297 - accuracy: 0.8218 - val_loss: 0.2026 - val_accuracy: 0.9494
Epoch 2/20
250/250 [=====] - 84s 336ms/step - loss: 0.2867 - accuracy: 0.9158 - val_loss: 0.2142 - val_accuracy: 0.9246
Epoch 3/20
250/250 [=====] - 84s 336ms/step - loss: 0.2295 - accuracy: 0.9273 - val_loss: 0.1148 - val_accuracy: 0.9591
Epoch 4/20
250/250 [=====] - 84s 337ms/step - loss: 0.2132 - accuracy: 0.9335 - val_loss: 0.1151 - val_accuracy: 0.9601
Epoch 5/20
250/250 [=====] - 84s 337ms/step - loss: 0.2056 - accuracy: 0.9400 - val_loss: 0.0844 - val_accuracy: 0.9688
Epoch 6/20
250/250 [=====] - 84s 336ms/step - loss: 0.1679 - accuracy: 0.9468 - val_loss: 0.2910 - val_accuracy: 0.9289
Epoch 7/20
250/250 [=====] - 84s 335ms/step - loss: 0.1561 - accuracy: 0.9510 - val_loss: 0.0700 - val_accuracy: 0.9828
Epoch 8/20
250/250 [=====] - 84s 335ms/step - loss: 0.1573 - accuracy: 0.9473 - val_loss: 0.1878 - val_accuracy: 0.9494
Epoch 9/20
250/250 [=====] - 84s 337ms/step - loss: 0.1625 - accuracy: 0.9470 - val_loss: 0.2225 - val_accuracy: 0.9429
Epoch 10/20
250/250 [=====] - 84s 335ms/step - loss: 0.1371 - accuracy: 0.9605 - val_loss: 0.0546 - val_accuracy: 0.9817
Epoch 11/20
250/250 [=====] - 84s 336ms/step - loss: 0.1311 - accuracy: 0.9605 - val_loss: 0.0616 - val_accuracy: 0.9795
Epoch 12/20
250/250 [=====] - 84s 337ms/step - loss: 0.1173 - accuracy: 0.9665 - val_loss: 0.1018 - val_accuracy: 0.9655
Epoch 13/20
250/250 [=====] - 84s 337ms/step - loss: 0.1476 - accuracy: 0.9555 - val_loss: 0.1314 - val_accuracy: 0.9731
Epoch 14/20
250/250 [=====] - 84s 336ms/step - loss: 0.0998 - accuracy: 0.9673 - val_loss: 0.0748 - val_accuracy: 0.9795
Epoch 15/20
250/250 [=====] - 84s 336ms/step - loss: 0.1164 - accuracy: 0.9633 - val_loss: 0.2551 - val_accuracy: 0.9450
Epoch 16/20

```

Şekil 4.3: InceptionV3 eğitim sonuçları

```

Epoch 17/20
250/250 [=====] - 84s 336ms/step - loss: 0.0998 - accuracy: 0.9673 - val_loss: 0.0748 - val_accuracy: 0.9795
Epoch 18/20
250/250 [=====] - 84s 336ms/step - loss: 0.1164 - accuracy: 0.9633 - val_loss: 0.2551 - val_accuracy: 0.9450
Epoch 19/20
250/250 [=====] - 84s 336ms/step - loss: 0.1088 - accuracy: 0.9678 - val_loss: 0.1568 - val_accuracy: 0.9612
Epoch 20/20
250/250 [=====] - 84s 335ms/step - loss: 0.1269 - accuracy: 0.9603 - val_loss: 0.2849 - val_accuracy: 0.9235
Epoch 21/20
250/250 [=====] - 84s 336ms/step - loss: 0.0942 - accuracy: 0.9698 - val_loss: 0.0458 - val_accuracy: 0.9881
Epoch 22/20
250/250 [=====] - 84s 336ms/step - loss: 0.1009 - accuracy: 0.9715 - val_loss: 0.1245 - val_accuracy: 0.9720
Epoch 23/20
250/250 [=====] - 84s 337ms/step - loss: 0.0816 - accuracy: 0.9738 - val_loss: 0.1267 - val_accuracy: 0.9591

```

Şekil 4.3.1: InceptionV3 eğitim sonuçları

Cnn modelinde yapılan çalışmada en düzenli sonuçları elde edilen çalışmadır. Başarı (accuracy) ve kayıp (loss) değerleri düzenli bir şekilde artış ve azalış göstermektedir. Bu çalışmada Adam optimizieri ile elde edilen en yüksek başarı sonucu %99.15 olarak bir başarı sonucu elde edilmiştir.

```

Epoch 1/100
434/434 [=====] - 2596s 6s/step - loss: 1.1078 - accuracy: 0.7948 - val_loss: 0.5874 - val_accuracy: 0.7995
Epoch 2/100
434/434 [=====] - 66s 151ms/step - loss: 0.1834 - accuracy: 0.9387 - val_loss: 1.9240 - val_accuracy: 0.2628
Epoch 3/100
434/434 [=====] - 66s 151ms/step - loss: 0.1139 - accuracy: 0.9587 - val_loss: 0.3569 - val_accuracy: 0.8278
Epoch 4/100
434/434 [=====] - 66s 152ms/step - loss: 0.1012 - accuracy: 0.9639 - val_loss: 0.0558 - val_accuracy: 0.9881
Epoch 5/100
434/434 [=====] - 66s 152ms/step - loss: 0.0834 - accuracy: 0.9730 - val_loss: 0.0508 - val_accuracy: 0.9835
Epoch 6/100
434/434 [=====] - 66s 152ms/step - loss: 0.0784 - accuracy: 0.9745 - val_loss: 0.1626 - val_accuracy: 0.9478
Epoch 7/100
434/434 [=====] - 66s 152ms/step - loss: 0.0941 - accuracy: 0.9709 - val_loss: 0.0259 - val_accuracy: 0.9908
Epoch 8/100
434/434 [=====] - 66s 151ms/step - loss: 0.0560 - accuracy: 0.9832 - val_loss: 0.0113 - val_accuracy: 0.9954
Epoch 9/100
434/434 [=====] - 66s 151ms/step - loss: 0.0882 - accuracy: 0.9723 - val_loss: 0.2372 - val_accuracy: 0.9148
Epoch 10/100
434/434 [=====] - 66s 151ms/step - loss: 0.0645 - accuracy: 0.9794 - val_loss: 0.0637 - val_accuracy: 0.9716
Epoch 11/100
434/434 [=====] - 66s 152ms/step - loss: 0.0652 - accuracy: 0.9816 - val_loss: 0.0233 - val_accuracy: 0.9927
Epoch 12/100
434/434 [=====] - 66s 152ms/step - loss: 0.0589 - accuracy: 0.9805 - val_loss: 0.1221 - val_accuracy: 0.9570
Epoch 13/100
434/434 [=====] - 66s 151ms/step - loss: 0.0480 - accuracy: 0.9830 - val_loss: 0.0044 - val_accuracy: 1.0000
Epoch 14/100
434/434 [=====] - 66s 152ms/step - loss: 0.0458 - accuracy: 0.9844 - val_loss: 0.2066 - val_accuracy: 0.9176
Epoch 15/100
434/434 [=====] - 66s 152ms/step - loss: 0.0658 - accuracy: 0.9814 - val_loss: 2.8728 - val_accuracy: 0.3681
Epoch 16/100
434/434 [=====] - 66s 152ms/step - loss: 0.0660 - accuracy: 0.9774 - val_loss: 0.0931 - val_accuracy: 0.9679
Epoch 17/100
434/434 [=====] - 66s 151ms/step - loss: 0.0600 - accuracy: 0.9790 - val_loss: 0.0333 - val_accuracy: 0.9826
Epoch 18/100

```

Şekil 4.4: CNN eğitim sonuçları

```

Epoch 19/100
434/434 [=====] - 65s 149ms/step - loss: 0.0317 - accuracy: 0.9909 - val_loss: 0.0075 - val_accuracy: 0.9973
Epoch 20/100
434/434 [=====] - 66s 152ms/step - loss: 0.0586 - accuracy: 0.9839 - val_loss: 0.0065 - val_accuracy: 0.9991
Epoch 21/100
434/434 [=====] - 66s 152ms/step - loss: 0.0487 - accuracy: 0.9847 - val_loss: 6.8979e-04 - val_accuracy: 1.0000
Epoch 22/100
434/434 [=====] - 66s 152ms/step - loss: 0.0295 - accuracy: 0.9892 - val_loss: 0.0023 - val_accuracy: 1.0000
Epoch 23/100
434/434 [=====] - 66s 153ms/step - loss: 0.0472 - accuracy: 0.9851 - val_loss: 0.0257 - val_accuracy: 0.9899
Epoch 24/100
434/434 [=====] - 66s 152ms/step - loss: 0.0547 - accuracy: 0.9875 - val_loss: 0.0059 - val_accuracy: 0.9991
Epoch 25/100
434/434 [=====] - 66s 153ms/step - loss: 0.0422 - accuracy: 0.9889 - val_loss: 0.0011 - val_accuracy: 1.0000
Epoch 26/100
434/434 [=====] - 67s 154ms/step - loss: 0.0349 - accuracy: 0.9880 - val_loss: 0.4637 - val_accuracy: 0.8416
Epoch 27/100
434/434 [=====] - 66s 153ms/step - loss: 0.0758 - accuracy: 0.9820 - val_loss: 0.0076 - val_accuracy: 0.9954
Epoch 28/100
434/434 [=====] - 66s 151ms/step - loss: 0.0307 - accuracy: 0.9901 - val_loss: 0.0117 - val_accuracy: 0.9945
Epoch 29/100
434/434 [=====] - 66s 152ms/step - loss: 0.0289 - accuracy: 0.9901 - val_loss: 0.0372 - val_accuracy: 0.9844
Epoch 30/100
434/434 [=====] - 66s 153ms/step - loss: 0.0398 - accuracy: 0.9884 - val_loss: 0.0053 - val_accuracy: 0.9973
Epoch 31/100
434/434 [=====] - 66s 151ms/step - loss: 0.0308 - accuracy: 0.9919 - val_loss: 0.0081 - val_accuracy: 0.9963
Epoch 32/100
434/434 [=====] - 66s 151ms/step - loss: 0.0527 - accuracy: 0.9857 - val_loss: 0.0078 - val_accuracy: 0.9973
Epoch 33/100
434/434 [=====] - 65s 150ms/step - loss: 0.0226 - accuracy: 0.9934 - val_loss: 0.0034 - val_accuracy: 0.9991
Epoch 34/100
434/434 [=====] - 66s 151ms/step - loss: 0.0298 - accuracy: 0.9891 - val_loss: 0.0136 - val_accuracy: 0.9936
Epoch 35/100
434/434 [=====] - 66s 151ms/step - loss: 0.0437 - accuracy: 0.9870 - val_loss: 7.7789e-04 - val_accuracy: 0.9991
Epoch 36/100
434/434 [=====] - 65s 151ms/step - loss: 0.0363 - accuracy: 0.9890 - val_loss: 0.0306 - val_accuracy: 0.9927

```

Şekil 4.4.1: CNN eğitim sonuçları

Epoch 37/100	
434/434 [=====]	- 65s 151ms/step - loss: 0.0444 - accuracy: 0.9898 - val_loss: 0.0036 - val_accuracy: 0.9982
Epoch 38/100	
434/434 [=====]	- 65s 150ms/step - loss: 0.0399 - accuracy: 0.9874 - val_loss: 6.1449e-04 - val_accuracy: 1.0000
Epoch 39/100	
434/434 [=====]	- 66s 151ms/step - loss: 0.0322 - accuracy: 0.9899 - val_loss: 0.0200 - val_accuracy: 0.9899
Epoch 40/100	
434/434 [=====]	- 65s 151ms/step - loss: 0.0252 - accuracy: 0.9913 - val_loss: 0.0018 - val_accuracy: 0.9991
Epoch 41/100	
434/434 [=====]	- 66s 151ms/step - loss: 0.0442 - accuracy: 0.9868 - val_loss: 0.0070 - val_accuracy: 0.9982
Epoch 42/100	
434/434 [=====]	- 66s 152ms/step - loss: 0.0260 - accuracy: 0.9916 - val_loss: 6.4316e-05 - val_accuracy: 1.0000
Epoch 43/100	
434/434 [=====]	- 66s 151ms/step - loss: 0.0356 - accuracy: 0.9892 - val_loss: 0.0377 - val_accuracy: 0.9872
Epoch 44/100	
434/434 [=====]	- 65s 150ms/step - loss: 0.0241 - accuracy: 0.9919 - val_loss: 0.0046 - val_accuracy: 0.9991
Epoch 45/100	
434/434 [=====]	- 65s 151ms/step - loss: 0.0267 - accuracy: 0.9892 - val_loss: 6.7474e-04 - val_accuracy: 1.0000
Epoch 46/100	
434/434 [=====]	- 66s 151ms/step - loss: 0.0211 - accuracy: 0.9927 - val_loss: 0.0011 - val_accuracy: 1.0000
Epoch 47/100	
434/434 [=====]	- 65s 150ms/step - loss: 0.0183 - accuracy: 0.9927 - val_loss: 0.0194 - val_accuracy: 0.9945
Epoch 48/100	
434/434 [=====]	- 65s 150ms/step - loss: 0.0544 - accuracy: 0.9863 - val_loss: 6.4261e-04 - val_accuracy: 1.0000
Epoch 49/100	
434/434 [=====]	- 66s 152ms/step - loss: 0.0300 - accuracy: 0.9899 - val_loss: 0.0015 - val_accuracy: 0.9991
Epoch 50/100	
434/434 [=====]	- 66s 151ms/step - loss: 0.0204 - accuracy: 0.9933 - val_loss: 0.0244 - val_accuracy: 0.9908
Epoch 51/100	
434/434 [=====]	- 66s 151ms/step - loss: 0.0124 - accuracy: 0.9951 - val_loss: 0.0057 - val_accuracy: 0.9973
Epoch 52/100	
434/434 [=====]	- 66s 151ms/step - loss: 0.0216 - accuracy: 0.9940 - val_loss: 3.5948e-04 - val_accuracy: 1.0000
Epoch 53/100	
434/434 [=====]	- 66s 151ms/step - loss: 0.0325 - accuracy: 0.9918 - val_loss: 0.0106 - val_accuracy: 0.9954
Epoch 54/100	
434/434 [=====]	- 66s 151ms/step - loss: 0.0350 - accuracy: 0.9892 - val_loss: 0.0355 - val_accuracy: 0.9881

Şekil 4.4.2: CNN eğitim sonuçları

Epoch 54/100	
434/434 [=====]	- 66s 151ms/step - loss: 0.0350 - accuracy: 0.9892 - val_loss: 0.0355 - val_accuracy: 0.9881
Epoch 55/100	
434/434 [=====]	- 66s 152ms/step - loss: 0.0421 - accuracy: 0.9903 - val_loss: 0.0063 - val_accuracy: 0.9982
Epoch 56/100	
434/434 [=====]	- 66s 152ms/step - loss: 0.0175 - accuracy: 0.9946 - val_loss: 5.6404e-04 - val_accuracy: 1.0000
Epoch 57/100	
434/434 [=====]	- 66s 153ms/step - loss: 0.0207 - accuracy: 0.9930 - val_loss: 0.0018 - val_accuracy: 0.9991
Epoch 58/100	
434/434 [=====]	- 66s 152ms/step - loss: 0.0135 - accuracy: 0.9964 - val_loss: 0.0033 - val_accuracy: 0.9991
Epoch 59/100	
434/434 [=====]	- 66s 152ms/step - loss: 0.0284 - accuracy: 0.9932 - val_loss: 8.9111e-04 - val_accuracy: 0.9991
Epoch 60/100	
434/434 [=====]	- 66s 152ms/step - loss: 0.0185 - accuracy: 0.9946 - val_loss: 0.0376 - val_accuracy: 0.9899
Epoch 61/100	
434/434 [=====]	- 66s 152ms/step - loss: 0.0252 - accuracy: 0.9935 - val_loss: 3.4929e-04 - val_accuracy: 1.0000
Epoch 62/100	
434/434 [=====]	- 66s 152ms/step - loss: 0.0183 - accuracy: 0.9942 - val_loss: 0.0000 - val_accuracy: 0.9954
Epoch 63/100	
434/434 [=====]	- 66s 152ms/step - loss: 0.0111 - accuracy: 0.9957 - val_loss: 0.0034 - val_accuracy: 0.9991
Epoch 64/100	
434/434 [=====]	- 66s 151ms/step - loss: 0.0148 - accuracy: 0.9966 - val_loss: 0.2406 - val_accuracy: 0.9414
Epoch 65/100	
434/434 [=====]	- 66s 151ms/step - loss: 0.0179 - accuracy: 0.9951 - val_loss: 0.0151 - val_accuracy: 0.9954
Epoch 66/100	
434/434 [=====]	- 66s 152ms/step - loss: 0.0238 - accuracy: 0.9934 - val_loss: 0.0018 - val_accuracy: 0.9991
Epoch 67/100	
434/434 [=====]	- 66s 152ms/step - loss: 0.0148 - accuracy: 0.9965 - val_loss: 0.5159 - val_accuracy: 0.8013
Epoch 68/100	
434/434 [=====]	- 66s 152ms/step - loss: 0.0346 - accuracy: 0.9928 - val_loss: 0.0093 - val_accuracy: 0.9963
Epoch 69/100	
434/434 [=====]	- 66s 152ms/step - loss: 0.0219 - accuracy: 0.9930 - val_loss: 0.0011 - val_accuracy: 0.9991
Epoch 70/100	
434/434 [=====]	- 66s 153ms/step - loss: 0.0260 - accuracy: 0.9935 - val_loss: 0.0603 - val_accuracy: 0.9853
Epoch 71/100	
434/434 [=====]	- 66s 153ms/step - loss: 0.0350 - accuracy: 0.9894 - val_loss: 0.0104 - val_accuracy: 0.9973

Şekil 4.4.3 CNN eğitim sonuçları

```

Epoch 72/100
434/434 [=====] - 66s 152ms/step - loss: 0.0111 - accuracy: 0.9972 - val_loss: 0.0212 - val_accuracy: 0.9918
Epoch 73/100
434/434 [=====] - 66s 152ms/step - loss: 0.0193 - accuracy: 0.9949 - val_loss: 1.0472e-04 - val_accuracy: 1.0000
Epoch 74/100
434/434 [=====] - 66s 152ms/step - loss: 0.0129 - accuracy: 0.9956 - val_loss: 0.0020 - val_accuracy: 0.9991
Epoch 75/100
434/434 [=====] - 66s 152ms/step - loss: 0.0150 - accuracy: 0.9956 - val_loss: 0.0045 - val_accuracy: 0.9982
Epoch 76/100
434/434 [=====] - 66s 151ms/step - loss: 0.0185 - accuracy: 0.9939 - val_loss: 0.0027 - val_accuracy: 0.9991
Epoch 77/100
434/434 [=====] - 66s 152ms/step - loss: 0.0215 - accuracy: 0.9947 - val_loss: 0.0037 - val_accuracy: 0.9991
Epoch 78/100
434/434 [=====] - 66s 152ms/step - loss: 0.0232 - accuracy: 0.9943 - val_loss: 5.7544e-04 - val_accuracy: 1.0000
Epoch 79/100
434/434 [=====] - 66s 152ms/step - loss: 0.0206 - accuracy: 0.9947 - val_loss: 4.0503e-04 - val_accuracy: 1.0000
Epoch 80/100
434/434 [=====] - 66s 152ms/step - loss: 0.0307 - accuracy: 0.9945 - val_loss: 7.4629e-04 - val_accuracy: 1.0000
Epoch 81/100
434/434 [=====] - 66s 151ms/step - loss: 0.0215 - accuracy: 0.9932 - val_loss: 6.4557e-05 - val_accuracy: 1.0000
Epoch 82/100
434/434 [=====] - 66s 152ms/step - loss: 0.0180 - accuracy: 0.9922 - val_loss: 0.0043 - val_accuracy: 0.9982
Epoch 83/100
434/434 [=====] - 66s 151ms/step - loss: 0.0118 - accuracy: 0.9968 - val_loss: 0.0079 - val_accuracy: 0.9973
Epoch 84/100
434/434 [=====] - 66s 151ms/step - loss: 0.0145 - accuracy: 0.9962 - val_loss: 0.0066 - val_accuracy: 0.9982
Epoch 85/100
434/434 [=====] - 65s 151ms/step - loss: 0.0072 - accuracy: 0.9982 - val_loss: 1.1512e-04 - val_accuracy: 1.0000
Epoch 86/100
434/434 [=====] - 66s 151ms/step - loss: 0.0188 - accuracy: 0.9936 - val_loss: 0.0073 - val_accuracy: 0.9982

```

Şekil 4.4.4 CNN eğitim sonuçları

```

Epoch 86/100
434/434 [=====] - 66s 151ms/step - loss: 0.0188 - accuracy: 0.9936 - val_loss: 0.0073 - val_accuracy: 0.9982
Epoch 87/100
434/434 [=====] - 66s 151ms/step - loss: 0.0352 - accuracy: 0.9910 - val_loss: 2.6647e-04 - val_accuracy: 1.0000
Epoch 88/100
434/434 [=====] - 65s 151ms/step - loss: 0.0120 - accuracy: 0.9963 - val_loss: 8.5856e-04 - val_accuracy: 1.0000
Epoch 89/100
434/434 [=====] - 65s 151ms/step - loss: 0.0105 - accuracy: 0.9963 - val_loss: 2.9304e-04 - val_accuracy: 1.0000
Epoch 90/100
434/434 [=====] - 66s 151ms/step - loss: 0.0147 - accuracy: 0.9950 - val_loss: 9.8680e-05 - val_accuracy: 1.0000
Epoch 91/100
434/434 [=====] - 66s 151ms/step - loss: 0.0114 - accuracy: 0.9961 - val_loss: 0.0034 - val_accuracy: 0.9991
Epoch 92/100
434/434 [=====] - 65s 151ms/step - loss: 0.0192 - accuracy: 0.9934 - val_loss: 1.9192e-05 - val_accuracy: 1.0000
Epoch 93/100
434/434 [=====] - 66s 151ms/step - loss: 0.0270 - accuracy: 0.9922 - val_loss: 9.6891e-04 - val_accuracy: 0.9991
Epoch 94/100
434/434 [=====] - 66s 151ms/step - loss: 0.0378 - accuracy: 0.9937 - val_loss: 0.0031 - val_accuracy: 0.9982
Epoch 95/100
434/434 [=====] - 65s 151ms/step - loss: 0.0218 - accuracy: 0.9940 - val_loss: 5.1221e-04 - val_accuracy: 1.0000
Epoch 96/100
434/434 [=====] - 66s 152ms/step - loss: 0.0182 - accuracy: 0.9958 - val_loss: 7.1838e-04 - val_accuracy: 1.0000
Epoch 97/100
434/434 [=====] - 66s 152ms/step - loss: 0.0111 - accuracy: 0.9969 - val_loss: 1.3455e-04 - val_accuracy: 1.0000
Epoch 98/100
434/434 [=====] - 66s 151ms/step - loss: 0.0118 - accuracy: 0.9976 - val_loss: 2.2253e-05 - val_accuracy: 1.0000
Epoch 99/100
434/434 [=====] - 66s 151ms/step - loss: 0.0181 - accuracy: 0.9947 - val_loss: 1.1210e-04 - val_accuracy: 1.0000
Epoch 100/100
434/434 [=====] - 66s 151ms/step - loss: 0.0187 - accuracy: 0.9946 - val_loss: 0.0015 - val_accuracy: 0.9991

```

Şekil 4.4.5 CNN eğitim sonuçları

Kullanılmış olan bütün modellerde iyi sonuçlar alabilmek için dropout, farklı optimizeler üzerinde iyi bir başarı elde etme ve Adaptive Learning Rate denenmiştir. Fakat Adaptive Learning Rate sonuçlarında herhangi bir iyi sonuç alınmadığı için bu projede optimizeler ve dropout iyileşme anlamında etkili olmuştur.



Sınıf Adı	Epoch	Batch Size	Optimizer	Target Size	Loss	Başarı oranı
EfficientNet5	20	16	adam	160,160	Categorical_crossentropy	%95.91
CNN	100	16	adam	160,160	Categorical_crossentropy	%99.15
InceptionV3	20	16	adam	160,160	Categorical_crossentropy	%81.52

Tablo 4.2: Kullanılmış olan değerler ve başarı oranları

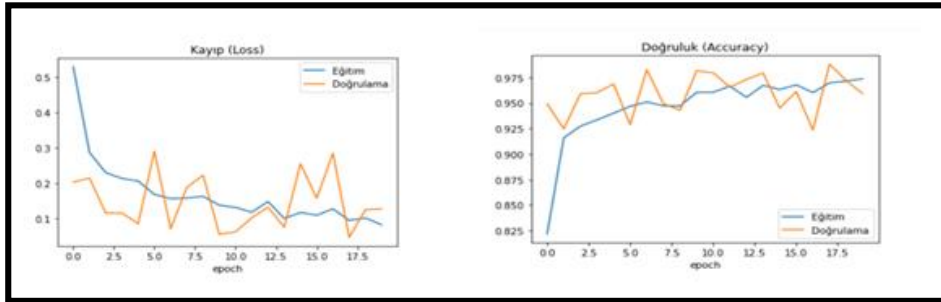
## 5. BULGULAR

Bu çalışmada elde edilen sonuçlar incelendiğinde performans anlamında EfficientNet5 modeli ile en yüksek performans sağlanmıştır. Elde edilen en iyi başarı “adam” optimizasyonu ile sağlanmıştır. En iyi başarı sonucu veren model CNN modelinde görülmektedir. En ideal epoch sayısı her model için farklılık göstermiştir. EfficientNet5 ve InceptionV3 modellerinde aşırı öğrenme (overfitting) engellenemediği için düzgün sonuçlar elde edilmemiştir.

Sınıf Adı	Acc	Loss	Val_acc	Val_loss
EfficientNet5	0.9738	0.0816	0.9591	0.1267
CNN	0.9874	0.0442	0.9915	0.0209
InceptionV3	0.9856	0.0494	0.8152	0.4651

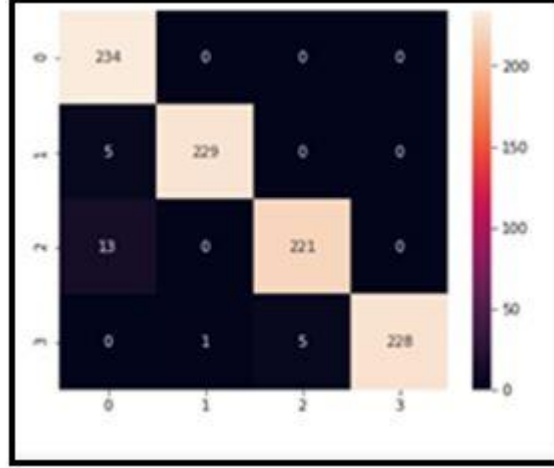
Tablo 5.1: Eğitim sonucundaki değerleri

Şekil 5.1’de EfficientNet5 modelde “adam” optimizasyonu için kayıp ve başarı grafiği aşağıdaki gibi belirtilmiştir. Grafiğe göre epoch sayısı 20, batch size değeri 16 alınarak, eğitim başarı oranı %95,91, test başarı oranı ise %97,38 olarak elde edilmiştir.



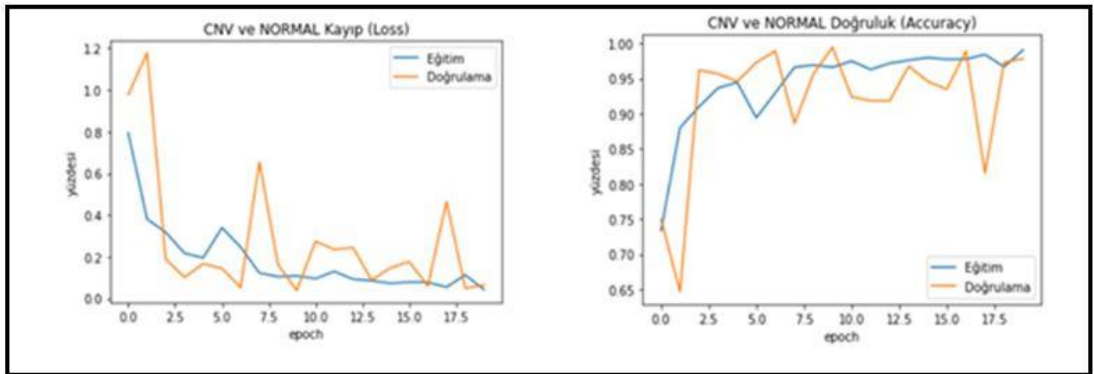
Şekil 5.1: EfficientNet5 model Adam optimizasyon başarı-loss grafikleri

Şekil 5.2’de, EfficientNet5 modelde “adam” optimizasyonları için confusion matrix grafiği verilmiştir.



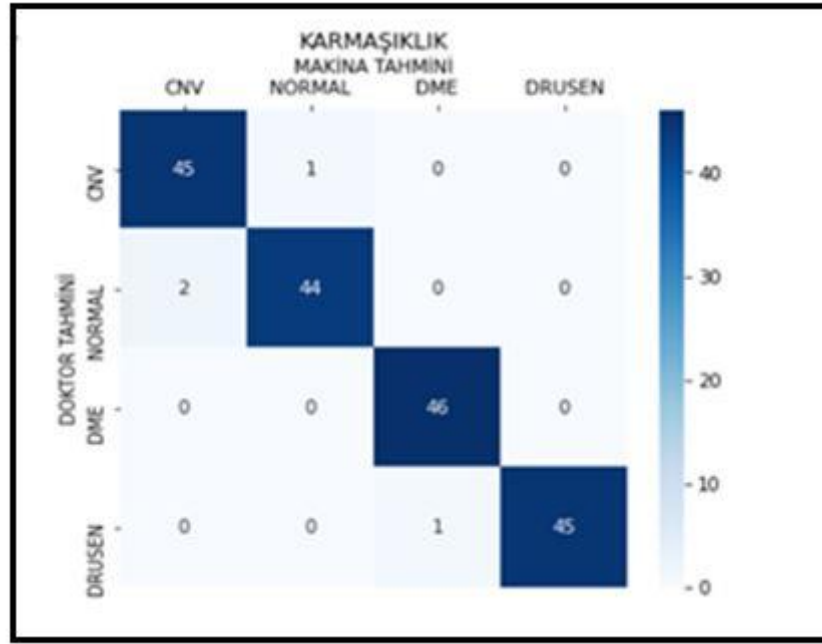
Şekil 5.2: EfficientNet5 model Adam optimizelerin confusion matrix grafiği.

Şekil 5.3’de, InceptionV3 modelde “Adam” optimizasyonu için başarı ve kayıp grafiği verilmiştir. Grafiğe göre epoch sayısı 20, batch size değeri 16 olarak alınmıştır . Eğitim başarı oranı %81.52, test başarısı ise %98.56 olarak elde edilmiştir.



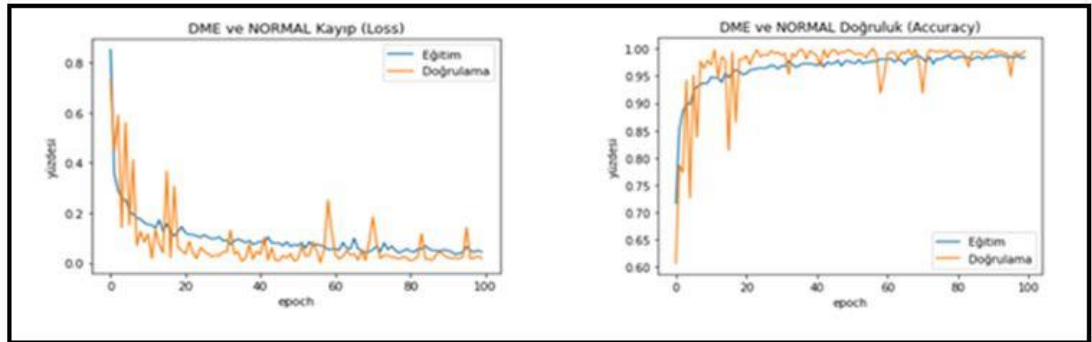
Şekil 5.3: InceptionV3 model Adam optimizeler başarı-loss grafikleri

Şekil 5.4’de, InceptionV3 modelde “Adam” optimizasyonu için confusion matrix grafiği aşağıda belirtilmiştir.



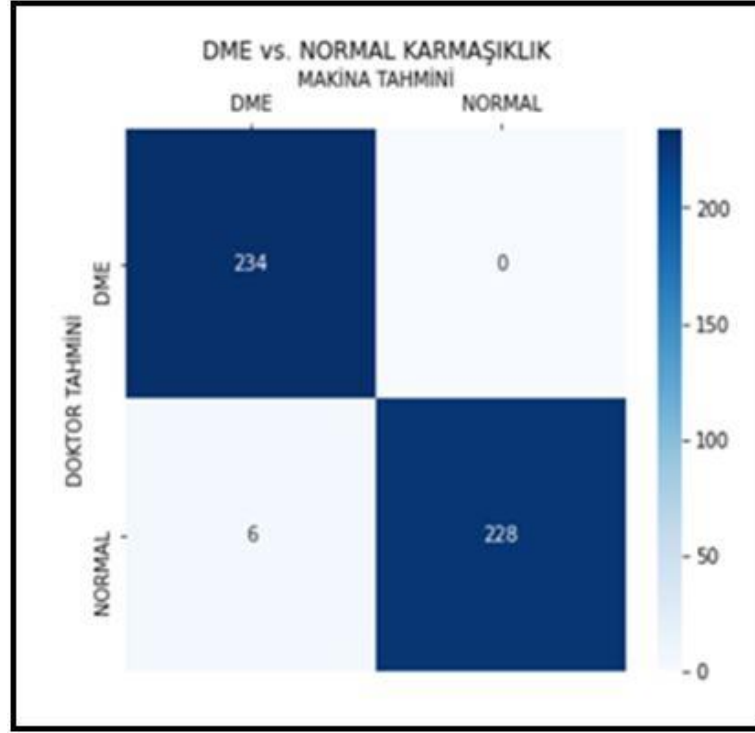
Şekil 5.4: InceptionV3 model Adam optimizer confusion matrix grafiği

Şekil 5.5’de, CNN modelde “Adam” optimizasyonu için başarı ve kayıp grafiği verilmiştir. Grafiğe göre epoch sayısı 100, batch size değeri 16 olarak alınmıştır . Eğitim başarı oranı %99.15 , test başarıısı ise %98.74 olarak elde edilmiştir.



Şekil 5.5: CNN model Adam optimizer başarı-loss grafikleri

Şekil 5.6’da, CNN modelde “Adam” optimizasyonu için confusion matrix grafiği aşağıda belirtilmiştir.



Şekil 5.6: CNN model Adam optimizier confusion matrix grafiği

## 6. SONUÇ VE ÖNERİLER

Optik Koherens Tomografi tanı sistemi göz hastalıklarının erken teşhisinde önemli rol oynamaktadır. Bu çalışmada derin öğrenme modelleri, 4 sınıf içeren “Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images For Classification” veri seti kullanılmakta. Bu çalışmada modelin daha verimli çalışabilmesi için ve yapılacak olan teşhisleri saptayabilmesi için veri artırımı işlemi uygulanmıştır. Veri artırımı işlemi uygulanan veri seti %87.5 eğitim, %12.5’i ise test verisi olmak üzere ayırma işlemi gerçekleştirilmiştir.

Geliştirilen projede veri setinde yer alan görüntülerin sınıflandırılması için “InceptionV3, EfficiNet5,CNN” derin öğrenme modelleri kullanılmıştır. InceptionV3 ve EfficiNet5 modellerinde aşırı öğrenme (overfitting) sorunu ile karşı karşıya



kalındığı için değerlendirilmeye alınmamıştır. Veri setine uygulanan CNN derin öğrenme modeli ile %98.74 başarı oranı tespit edilmiştir. Elde edilen sonuçlar doğrultusunda CNN derin öğrenme modelinin aşırı öğrenme durumunun olmadığı gözlemlenmiştir. Bu gözlemlene sonucunda; CNN model erken teşhis için önerilebilir. Kullanılmış olan optimizasyon, modelin başarısında ve doğru sonuç verme olasılığını arttırmak için önemli rol oynamaktadır. Elde edilen sonuçlar doğrultusunda uygulama geliştirilerek sağlık alanında kullanılabilir.

Geliştirilen projede sonuç olarak; performans olarak en iyi derin öğrenme modeli bulunmaya çalışılmıştır. Modelin kısa sürede, minimum maliyet ile doğru sonuçları elde etmesidir. Optik Koherens Tomografi (OCT) tespiti için denenen modeller arasında en iyi tahmini veren model ve tahmin tespit edilmiştir. Optik Koherens Tomografi (OCT) göz hastalığı tanısı için CNN derin öğrenme modelinin kullanılması ile göz hastalıklarının doğru ve erken teşhis edilmesi amaçlanmış olup sağlık alanında kullanılabilir olmasını ortaya koymaktadır..

## KAYNAKLAR

1. Yaya, O., Taş, İ., Ayrancıoğlu, B. N., & Önder, F. (2015). Diyabetik Makuler Ödemde Seröz Makula Dekolmanı Sıklığı. *Turkish Journal of Ophthalmology/Turk Oftalmoloji Dergisi*, 45(3).
2. ÖZDEMİR, M. H., ŞENTÜRK, F., KARAÇORLU, S. A., & KARAÇORLU, M. (2011). Okülökutanöz albinizmde fovea hipoplazisinin spektral domain optik koherens tomografi ile gösterilmesi. *Retina-Vitreus*, 19(1), 53-55.
3. KARAKÜÇÜK, Y., & EKER, S. YAPAY ZEKA VE DERİN ÖĞRENME UYGULAMALARI.(2014).
4. ŞAHİN, A. R., & DOĞAN, K. Yapay Zeka.(2016).
5. URL-1. Veri seti, <https://data.mendeley.com/datasets/rsbjbr9sj/2> Erişim Tarihi,2021.

6. URL-2. CNN, <https://teknoloji.org> Erişim Tarihi,2021.
7. URL-3 . EfficiNet5, <https://medium.com> Erişim Tarihi,2021.
8. URL-4 InceptionV3, <https://bilisim.io> Erişim Tarihi,2021.
9. Özgür DOĞAN, (Kasım 26,2020).
10. Bisong, E. (2019). Google colaboratory. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform* (pp. 59-64). Apress, Berkeley, CA.
11. Dierbach, C., "Introduction to Computer Science Using Python: A Computational Problem-solving Focus", Wiley Publishing, (2012).
12. Ayvaz, U., Çoban, A., Gürüler, H., & Peker, M. *Python Dilinin Öznitelikleri, Programlama Eğitiminde ve Yazılım Dünyasındaki Yeri* (2012).
13. ŞİŞMAN, A., ODABAŞI, H. F., & AKKOYUNLU, B. (2019). *Eğitim Teknolojisi Okumaları*.
14. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
15. Tang (May 2018). *Intelligent Mobile Projects with TensorFlow*. Packt Publishing. pp. Chapter 2. ISBN 9781788834544.
16. Milton-Barker, Adam. "Inception V3 Deep Convolutional Architecture For Classifying Acute Myeloid/Lymphoblastic Leukemia". intel.com. Intel. Retrieved 2 February 2019.

