

**DERİN ÖĞRENME İLE AKCİĞER HASTALIKLARININ TESPİTİ**

**AYŞEGÜL ALEYNA ÜNSAL**

**EMİRHAN AYDIN**

**SEVDENUR YILMAZ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

**BİTİRME PROJESİ**

**DR. ÖĞR. ÜYESİ MAHİR KAYA**

**Haziran- 2022**

**Her hakkı saklıdır**

**T.C.**

**TOKAT GAZİOSMANPAŞA ÜNİVERSİTESİ**

**MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

**BİTİRME PROJESİ**

**DERİN ÖĞRENME İLE AKCİĞER HASTALIKLARININ TESPİTİ**

**AYŞEGÜL ALEYNA ÜNSAL**

**EMİRHAN AYDIN**

**SEVDENUR YILMAZ**

TOKAT

Haziran- 2022

Her hakkı saklıdır.

**PROJE BEYANI**

Proje yazım kurallarına uygun olarak hazırlanan bu projenin yazılmasında bilimsel ahlak kurallarına uyulduğunu, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, projenin içerdiği yenilik ve sonuçların başka bir yerden alınmadığını, kullanılan verilerde herhangi bir tahrifat yapılmadığını, projenin herhangi bir kısmının bu üniversite veya başka bir üniversitedeki başka bir proje çalışması olarak sunulmadığını beyan ederim.

**Ayşegül Aleyna ÜNSAL**

**Emirhan AYDIN**

**Sevdenur YILMAZ**

**Haziran 2022**

**ÖZET**

**BİTİRME PROJESİ**

**DERİN ÖĞRENME İLE AKCİĞER HASTALIKLARININ TESPİTİ**

**Ayşegül Aleyna ÜNSAL**

**Emirhan AYDIN**

**Sevdenur YILMAZ**

**TOKAT GAZİOSMANPAŞA ÜNİVERSİTESİ**

**MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

(DANIŞMANI: Dr. Öğr. Üyesi Mahir KAYA)

Röntgen (X-Ray) tıbbi görüntüleme teknikleri uzun yıllardan beri hastalıkların teşhisi,tanısı ve tedavisi için kullanılmıştır. Akciğer hastalıklarında başarılı ve erken teşhis hayati önem taşımaktadır. Akciğer hastalılarında röntgen(X-Ray) görüntülerideki hastalıklı dokuların tanısında kullanılacak olan derin öğrenme metotları ile, akciğer hastalıklarının teşhisinde derin öğrenme alanında araştırmalar mevcuttur. Yapılan çalışmada insan/X-Ray görüntüleri üzerinden akciğer üzerinde hastalıklı doku tespiti için birden fazla derin öğrenme modelleri önerilmiştir. Oluşturulan derin öğrenme modellerinde veri seti olarak hastalıklı doku bulunan ve bulunmayan 112120 tane röntgen görüntüsü; oluşturulacak olan derin öğrenme modeline göre farklı sayılarda kullanılan sınıflara göre kullanılmıştır. Lisans bitirme çalışmasında, akciğer hastalıklarının teşhisi öncesi sınıflandırma için derin öğrenme modelleri kullanılmıştır. Veri seti önce hastalıklı doku bulunup bulunmama durumlarına göre etiketlenmiştir. Kullanılan farklı modellerde birden fazla sayıda eğitim modeli oluşturulmuş olup aralarındaki başarı oranlarının farkları gözlemlenerek, transfer öğrenme metotunun özgün modellere kıyasla çok daha yüksek başarı oranlarına sahip olduğu gözlemlenmiştir.

Anahtar kelimeler: Derin Öğrenme, Medikal Görüntü, CNN, Sınıflandırma, Görüntü İyileştirme, Akciğer Hastalıkları Teşhisi, Python, Görüntü İşleme

**ÖNSÖZ**

Bu araştırma için bizi yönlendiren, karşılaştığımız zorlukları bilgi ve tecrübesi ile aşmamıza yardımcı olan değerli danışman hocamız Dr. Öğr. Üyesi Mahir Kaya’ya teşekkürlerimizi sunarız. Hayatımız boyunca yanımızda olan desteklerini ve sevgilerini bizlerden hiç esirgemeyen değerli anne ve babalarımıza yürekten sevgi ve teşekkürlerimizi sunuyoruz.

**Ayşegül Aleyna ÜNSAL**

**Emirhan AYDIN**

**Sevdenur YILMAZ**

**Haziran 2022**

**İÇİNDEKİLER**

**ÖZET ……………………………………………………………………………………………….: 4**

**ÖNSÖZ ……………………………………………………………………………………………..: 5**

**İÇİNDEKİLER ..…………………………………………………………………………………...: 6**

**KISALTMALAR .………………………………………………………………………………….: 9**

**ŞEKİL LİSTESİ …………………………………………………………………………………...: 10**

**1. GİRİŞ ……………………………………………………………………………………………: 14**

**2. GENEL BİLGİLER …………………………………………………………………………....: 15**

**3. MATERYAL VE YÖNTEM …………………………………………………………………..: 16**

3.1Materyal **……………………………………………………………………………..…: 16**

3.1.1 Python **……………………………………………………………………….: 16**

3.1.2 Tensorflow **……………………………………………………………..……: 16**

3.1.3. Keras **……………………………………………………………………..…: 16**

3.1.4 CUDA **………………………………………………………………….……: 16**

3.1.5. cuDNN **……………………………………………………………………...: 17**

3.1.6. NIH Chest X-Rays **………………………………………………………….: 17**

3.1.7 Anaconda **………………………………………………………...………….: 17**

3.1.8 Spyder **……………………………………………………………………….: 18**

3.2 Yöntem **………………………………………………………………………………...: 18**

3.2.1 Genel Yöntem **……………………………………..………………………...: 18**

3.2.1.1 Yapay Zeka **…………..…………………………………………...: 18**

3.2.1.2 Makine Öğrenmesi **..………………………….…………………...: 18**

3.2.1.2.1 Denetimli Öğrenme **..…………………………………...: 19**

3.2.1.3 Yapay Sinir Ağları **..……………………………………………....: 19**

3.2.1.4 Derin Öğrenme **..………………………………………………......: 20**

3.2.1.4.1 CNN **..……………………………………………….......: 20**

3.2.1.4.2 Sequential **..…………………………………………......: 23**

3.2.1.4.3 Image Data Generator **..…...……………………….........: 23**

3.2.1.5 Transfer Öğrenme **..…...………………………..................: 23**

3.2.1.6 Hiperparametreler **..…...………………………..................: 23**

3.2.1.7 Algoritmalar **..…...………………………………..….........: 25**

3.2.2 Uygulama **..…...………………………………..…..........................................: 26**

3.2.2.1 Verisetlerinin Hazırlanması **..…...…………………..…..................: 26**

3.2.2.1.1 4 Sınıflı Model **..…...…………………..…......................: 26**

3.2.2.1.2 3 Sınıflı Model **..…...…………………..…......................: 26**

3.2.2.1.3 Binary Model **..…...…………………..…........................: 26**

3.2.2.2 CNN Modellerinin Hazırlanması **..…...……..…..............................: 26**

3.2.2.2.1 4 Sınıflı Model **…...…………………..…........................: 26**

3.2.2.2.2 3 Sınıflı Model **..…...……………..…..…........................: 33**

3.2.2.2.3 Binary Model **..…...…………………..…........................: 37**

**4. BULGULAR …………………………………………………………………………………….: 56**

4.1 4 Sınıflı Model **..…...…………………………………………….…..…........................: 56**

4.1.1 Model – 1 **..…...…………………..………………………………………….: 56**

4.1.2 Model – 2 **..…...…………………..….............................................................: 57**

4.1.3 Model – 3 **..…...…………………..….............................................................: 58**

4.1.4 Model – 4 **..…...…………………..….............................................................: 59**

4.1.5 Transfer Model – 1 **..…...…………………..…..............................................: 60**

4.1.6 Transfer Model – 2 **..…...…………………..…..............................................: 61**

4.1.7 Transfer Model – 3 **..…...…………………..…..............................................: 62**

4.2 3 Sınıflı Model **..…...…………………..…....................................................................: 63**

4.2.1 Model – 1 **..…...…………………..…............................................................: 63**

4.2.2 Model – 2 **..…...…………………..…............................................................: 64**

4.2.3 Model – 3 **..…...…………………..…............................................................: 65**

4.2.4 Transfer Model – 1 **..…...…………………..…..............................................: 66**

4.3 Binary Model **..…...…………………..…......................................................................: 67**

4.3.1 Model – 1 **..…...…………………..…............................................................: 67**

4.3.2 Model – 2 **..…...…………………..…............................................................: 68**

4.3.3 Model – 3 **..…...…………………..…............................................................: 69**

4.3.4 Model – 3.1 **..…...…………………..….........................................................: 70**

4.3.5 Model – 3.2 **..…...…………………..….........................................................: 71**

4.3.6 Model – 3.2.1 **..…...…………………..…......................................................: 72**

4.3.7 Model – 3.2.2 **..…...…………………..…......................................................: 73**

4.3.8 Model – 3.2.3 **..…...…………………..…......................................................: 74**

4.3.9 Model – 3.2.4 **..…...…………………..…......................................................: 75**

4.3.10 Model – 3.2.2.1 **..…...…………………..….................................................: 76**

4.3.11 Model – 3.2.2.2 **..…...…………………..….................................................: 77**

4.3.12 Model – 3.2.2.3 **..…...…………………..….................................................: 78**

4.3.13 Model – 3.2.2.4 **..…...…………………..….................................................: 79**

4.3.14 Model – 3.2.2.0.1 **..…...…………………..…..............................................: 80**

4.3.15 Model – 3.2.2.0.2 **..…...…………………..…..............................................: 81**

4.3.16 Transfer Model – 1 **..…...…………………..…...........................................: 82**

4.3.17 Transfer Model – 2 **..…...…………………..…...........................................: 83**

4.3.18 Transfer Model – 3 **..…...…………………..…...........................................: 84**

**5. SONUÇ ………………………...……………………………………………………………….: 85**

5.1 4 Sınıflı Model **………...………………………………………………………………: 85**

5.2 3 Sınıflı Model **………...………………………………………………………………: 86**

5.3 Binary Model **………...…………………………………………………………..……: 87**

5.4 Çıkarımlar **……...……...………………………………………………………………: 88**

**6. KAYNAKLAR …………………………………………………………………………………: 89**

**7. ÖZGEÇMİŞ ……………………………………………………………………………………: 91**

**KISALTMALAR**

**Kısaltmalar** **Açıklama**

CNN Convolution Neural Network

NIH National Institutes of Health

CPU Central Process Unit

GPU Graphics Processing Unit

CUDA Compute Unified Device Architecture

ONEIROS Open-ended Neuro-Electronic Intelligent Robot Operating System

CNTK Cognitive Kit

SDK Software Development Kit

BT Bilgisayarlı Tomografi

cuDNN CUDA Deep Neural Network Library cuDNN

MIT Massachusetts Institute of Technology

IDE Integrated Development Environment

GUI Graphical User Interface

CLI Command Line Interface

YSA Yapay Sinir Ağları

AI Artifical Intelligence

ReLU Rectifier Linear Unit

**ŞEKİL LİSTESİ**

**Şekil** **Sayfa**

Şekil 1. Göğüs Röntgenlerinde Gözlenen 8 Yaygın Göğüs Hastalıkları …………………..……: 14

Şekil 2. Tek Gizli Katmanlı ve Çok Gizli Katmanlı Sinir Ağı Yapısı …………………..………: 19

Şekil 3. CNN Mimarisi ……………………….…………………..………………….……….…: 20

Şekil 4. ReLU Fonksiyon Grafiği ……………………………………………………………….: 22

Şekil 5. Sigmoid Fonksiyon Grafiği …………………………………………………………….: 22

Şekil 6. Geleneksel Makine Öğrenmesi ile Transfer Öğrenmenin Farkını Gösteren Örnek …....: 23

Şekil 7. Hiperparametrelerin Model Sonuçlarına Etkilerinin Örneklendirilmesi …………….....: 25

Şekil 8. Çalışmayan Bir Lambayı Çalışır Hale Getiren Algoritma ………………………......…: 25

Şekil 9. Model – 1 CNN Mimarisi ………………………....................................................……: 27

Şekil 10. Model – 2 CNN Mimarisi ………………………......................................................…: 28

Şekil 11. Model – 3 CNN Mimarisi ………………………....................................................…..: 29

Şekil 12. Model – 4 CNN Mimarisi ………………………....................................................…..: 30

Şekil 13. Transfer Model – 1 CNN Mimarisi ………………………....................................…...: 30

Şekil 14. Transfer Model – 2 CNN Mimarisi ………………………....................................…...: 31

Şekil 15. Transfer Model – 3 CNN Mimarisi ………………………...................................……: 32

Şekil 16. Model – 1 CNN Mimarisi ……………………...………….......................................…: 33

Şekil 17. Model – 2 CNN Mimarisi ……………………...………….......................................…: 34

Şekil 18. Model – 3 CNN Mimarisi ……………………...………….......................................…: 35

Şekil 19. Transfer Model – 1 CNN Mimarisi ……………………….......................................…: 36

Şekil 20. Model – 1 CNN Mimarisi ……………………...………….......................................…: 38

Şekil 21. Model – 2 CNN Mimarisi ……………………...………….......................................…: 39

Şekil 22. Model – 3 CNN Mimarisi ……………………...………….......................................…: 40

Şekil 23. Model – 3.1 CNN Mimarisi ……………………...…………....................................…: 41

Şekil 24. Model – 3.2 CNN Mimarisi ……………………...…………........................................: 42

Şekil 25. Model – 3.2.1 CNN Mimarisi ……………………...………….............................……: 43

Şekil 26. Model – 3.2.2 CNN Mimarisi ……………………...………….................................…: 44

Şekil 27. Model – 3.2.3 CNN Mimarisi ……………………...………….............................……: 45

Şekil 28. Model – 3.2.4 CNN Mimarisi ……………………...………….................................…: 46

Şekil 29. Model – 3.2.2.1 CNN Mimarisi ……………………......….......................................…: 47

Şekil 30. Model – 3.2.2.2 CNN Mimarisi …………...…...………….......................................…: 48

Şekil 31. Model – 3.2.2.3 CNN Mimarisi ………...……...………….......................................…: 49

Şekil 32. Model – 3.2.2.4 CNN Mimarisi ……………………...……......................................…: 50

Şekil 33. Model – 3.2.2.0.1 CNN Mimarisi ……………………...…………...........................…: 51

Şekil 34. Model – 3.2.2.0.2 CNN Mimarisi ……………………...…………...........................…: 52

Şekil 35. Transfer Model – 1 CNN Mimarisi ……………………...…………........................…: 53

Şekil 36. Transfer Model – 2 CNN Mimarisi ……………………...…………....................……: 54

Şekil 37. Transfer Model – 3 CNN Mimarisi ……………………...…………........................…: 55

Şekil 38. Model – 1 Accuracy – Validation Accuracy Grafik Tablosu …………………………: 56

Şekil 39. Model – 1 Loss – Validation Loss Grafik Tablosu ……………………………….......: 56

Şekil 40. Model – 2 Accuracy – Validation Accuracy Grafik Tablosu ……………………..…..: 57

Şekil 41. Model – 2 Loss – Validation Loss Grafik Tablosu …………………………………...: 57

Şekil 42. Model – 3 Accuracy – Validation Accuracy Grafik Tablosu ………………………....: 58

Şekil 43. Model – 3 Loss – Validation Loss Grafik Tablosu ……………………..…………….: 58

Şekil 44. Model – 4 Accuracy – Validation Accuracy Grafik Tablosu ………………………....: 59

Şekil 45. Model – 4 Loss – Validation Loss Grafik Tablosu ……………………..………….....: 59

Şekil 46. Transfer Model – 1 Accuracy – Validation Accuracy Grafik Tablosu …………..…...: 60

Şekil 47. Transfer Model – 1 Loss – Validation Loss Grafik Tablosu ……………………..…...: 60

Şekil 48. Transfer Model – 2 Accuracy – Validation Accuracy Grafik Tablosu …………..…...: 61

Şekil 49. Transfer Model – 2 Loss – Validation Loss Grafik Tablosu ………………..………...: 61

Şekil 50. Transfer Model – 3 Accuracy – Validation Accuracy Grafik Tablosu …………….…: 62

Şekil 51. Transfer Model – 3 Loss – Validation Loss Grafik Tablosu …………..…………...…: 62

Şekil 52. Model – 1 Accuracy – Validation Accuracy Grafik Tablosu …………..……………..: 63

Şekil 53. Model – 1 Loss – Validation Loss Grafik Tablosu ……………...………………...…: 63

Şekil 54. Model – 2 Accuracy – Validation Accuracy Grafik Tablosu …………………....……: 64

Şekil 55. Model – 2 Loss – Validation Loss Grafik Tablosu …………….…………….....……: 64

Şekil 56. Model – 3 Accuracy – Validation Accuracy Grafik Tablosu …...……..…………......: 65

Şekil 57. Model – 3 Loss – Validation Loss Grafik Tablosu …………..…………….………...: 65

Şekil 58. Transfer Model – 1 Accuracy – Validation Accuracy Grafik Tablosu ….……..…..…: 66

Şekil 59. Transfer Model – 1 Loss – Validation Loss Grafik Tablosu …..…...……………..…: 66

Şekil 60. Model – 1 Accuracy – Validation Accuracy Grafik Tablosu ..……...……...………..: 67

Şekil 61. Model – 1 Loss – Validation Loss Grafik Tablosu ………………...………………...: 67

Şekil 62. Model – 2 Accuracy – Validation Accuracy Grafik Tablosu …...………….…..……: 68

Şekil 63. Model – 2 Loss – Validation Loss Grafik Tablosu ………………………..…………: 68

Şekil 64. Model – 3 Accuracy – Validation Accuracy Grafik Tablosu ….…….………...……: 69

Şekil 65. Model – 3 Loss – Validation Loss Grafik Tablosu …………...……..………....….…: 69

Şekil 66. Model – 3.1 Accuracy – Validation Accuracy Grafik Tablosu ………….………….: 70

Şekil 67. Model – 3.1 Loss – Validation Loss Grafik Tablosu …………………..………….…: 70

Şekil 68. Model – 3.2 Accuracy – Validation Accuracy Grafik Tablosu ……..…..…………..: 71

Şekil 69. Model – 3.2 Loss – Validation Loss Grafik Tablosu …………..…...……………….: 71

Şekil 70. Model – 3.2.1 Accuracy – Validation Accuracy Grafik Tablosu …...……..………..: 72

Şekil 71. Model – 3.2.1 Loss – Validation Loss Grafik Tablosu ……………..……………….: 72

Şekil 72. Model – 3.2.2 Accuracy – Validation Accuracy Grafik Tablosu …………..……….: 73

Şekil 73. Model – 3.2.2 Loss – Validation Loss Grafik Tablosu …………..…………………..: 73

Şekil 74. Model – 3.2.3 Accuracy – Validation Accuracy Grafik Tablosu …………..……….: 74

Şekil 75. Model – 3.2.3 Loss – Validation Loss Grafik Tablosu …………..….……………….: 74

Şekil 76. Model – 3.2.4 Accuracy – Validation Accuracy Grafik Tablosu …….……..………: 75

Şekil 77. Model – 3.2.4 Loss – Validation Loss Grafik Tablosu …………..…..…………...…: 75

Şekil 78. Model – 3.2.2.1 Accuracy – Validation Accuracy Grafik Tablosu ……..……..……: 76

Şekil 79. Model – 3.2.2.1 Loss – Validation Loss Grafik Tablosu …………..……...…………: 76

Şekil 80. Model – 3.2.2.2 Accuracy – Validation Accuracy Grafik Tablosu ………..…..……: 77

Şekil 81. Model – 3.2.2.2 Loss – Validation Loss Grafik Tablosu …………..………...………: 77

Şekil 82. Model – 3.2.2.3 Accuracy – Validation Accuracy Grafik Tablosu ………........……: 78

Şekil 83. Model – 3.2.2.3 Loss – Validation Loss Grafik Tablosu …………..…………...……: 78

Şekil 84. Model – 3.2.2.4 Accuracy – Validation Accuracy Grafik Tablosu …...……….……: 79

Şekil 85.Model – 3.2.2.4 Loss – Validation Loss Grafik Tablosu …………..……...……….…: 79

Şekil 86. Model – 3.2.2.0.1 Accuracy – Validation Accuracy Grafik Tablosu ……...…..……: 80

Şekil 87. Model – 3.2.2.0.1 Loss – Validation Loss Grafik Tablosu …………..……...…….…: 80

Şekil 88. Model – 3.2.2.0.2 Accuracy – Validation Accuracy Grafik Tablosu ………….……: 81

Şekil 89.Model – 3.2.2.0.2 Loss – Validation Loss Grafik Tablosu ……………………...……: 81

Şekil 90. Transfer Model – 1 Accuracy – Validation Accuracy Grafik Tablosu …...…….……: 82

Şekil 91. Transfer Model – 1 Loss – Validation Loss Grafik Tablosu …………..………...…..: 82

Şekil 92. Transfer Model – 2 Accuracy – Validation Accuracy Grafik Tablosu ………………: 83

Şekil 93. Transfer Model – 2 Loss – Validation Loss Grafik Tablosu …………..……...…..…: 83

Şekil 94. Transfer Model – 3 Accuracy – Validation Accuracy Grafik Tablosu ………....……: 84

Şekil 95. Transfer Model – 3 Loss – Validation Loss Grafik Tablosu …………………...……: 84

Şekil 96. 4 Sınıflı Modellerin Accuracy Grafik Tablosu ………………………………………: 85

Şekil 97. 4 Sınıflı Modellerin Loss Grafik Tablosu ……………………………………………: 85

Şekil 98. 3 Sınıflı Modellerin Accuracy Grafik Tablosu …………………………..……..……: 86

Şekil 99. 3 Sınıflı Modellerin Loss Grafik Tablosu ………………………………...….....……: 86

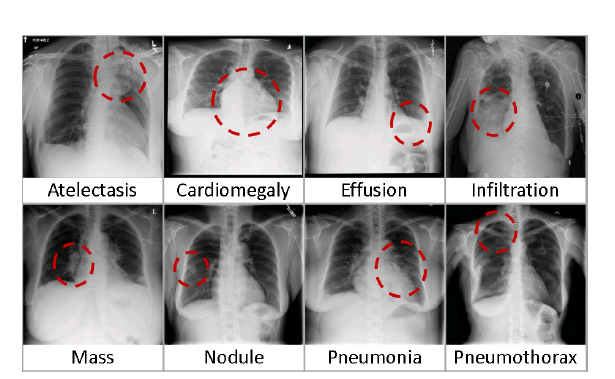
Şekil 100. Binary Modellerin Accuracy Grafik Tablosu ……………………………...…..……: 87

Şekil 101. Binary Modellerin Loss Grafik Tablosu ………………………………...……..……: 87

**1. GİRİŞ**

Radyolojik görüntüleme teknolojisi akciğer hastalıklarının görüntülenmesi ve tespit edilmesinde kullanılır. Röntgen filmleri ilk olarak 1895 yılında Alman fizikçi Wilhelm Röntgen tarafından çekilmiştir.**[1]** Kullanacağımız verisetinde**[2]** 13 farklı akciğer hastalığı bulunmaktadır.

Diğer bir çok hastalıkta olduğu gibi akciğer hastalıklarında da erken teşhis önemli bir etmendir. Günümüzde erken teşhisin yapılması için röntgen filmlerinin derin öğrenme modelleri ile yorumlanmasından yararlanılmaktadır. Bu projede Infıltration, Atelectasis, Nodule, Mass ve Pneumothorax isimli akciğer hastalıklarının röntgen filmlerinin derin öğrenme modelleri ile yorumlanması ve model mimarilerinde kullanılan yöntemlerin başarı oranlarının karşılaştırılması amaçlanmıştır. Kullanılacak olan yöntemlerle ilgili bilgiler Genel Bilgiler bölümünde açıklanmıştır.



Şekil 1. Göğüs Röntgenlerinde Gözlenen 8 Yaygın Göğüs Hastalıkları

**2. GENEL BİLGİLER**

Gerçekleştirilen bu çalışmada hastalık tespitinin yapılabilmesi için akciğer röntgen görüntüleri, derin öğrenme yöntemlerinden CNN modeli, transfer öğrenme yöntemi, tek sınıflı ve çok sınıflı sınıflandırma yöntemleri ile işlenmiş olup uygulamalar bu doğrultuda gerçekleştirilmiştir.

İlk olarak NIH Chest X-rays isimli verisetinde bulunan 112.120 adet röntgen görüntüsü içerisinden belirlediğimiz 3 hastalık ve sağlıklı akciğer görüntülerini istenilen sayıda ayrıştıran, test, eğitim ve doğrulama klasörlerine bölen algoritma yazılmıştır.

Daha sonra gerekli kütüphaneler tespit edilip kurulmuştur.

Sonrasında 4 sınıf içeren çok sınıflı sınıflandırma modelleri oluşturulup, doğruluk başarı oranları karşılaştırıldıktan sonra transfer öğrenme yöntemi de kullanılarak 4 sınıflı sınıflandırma için başarı oranları grafiğine ulaşılmıştır.

Daha sonra önceden oluşturduğumuz veriseti hazırlama algoritmasını kullanarak 3 sınıf içeren veriseti oluşturulduktan sonra bu veriseti için çok sınıflı sınıflandırma modelleri oluşturulup, doğruluk başarı oranları karşılaştırıldıktan sonra transfer öğrenme yöntemi de kullanılarak 3 sınıflı sınıflandırma için başarı oranları grafiğine ulaşılmıştır.

Sonrasında önceden oluşturduğumuz veriseti hazırlama algoritmasını kullanarak 2 sınıf içeren veriseti oluşturulduktan sonra bu veriseti için iki sınıflı sınıflandırma modelleri oluşturulup, doğruluk başarı oranları karşılaştırıldıktan sonra transfer öğrenme yöntemi de kullanılarak 2 sınıflı sınıflandırma için başarı oranları grafiğine ulaşılmıştır.

Son olarak 2, 3 ve 4 sınıflı sınıflandırma modellerinin grafikleri karşılaştırılmıştır.

**3. MATERYAL VE YÖNTEM**

**3.1 Materyal**

Bu çalışmada Anaconda Navigator Programı, Python Programlama Dili, Spyder IDE, açık kaynak kodlu TensorFlow ve Keras kütüphaneleri, CUDA, cuDNN teknolojileri ve NIH Göğüs Röntgenleri kullanılarak derin öğrenme ile akciğer hastalıkları tespit uygulaması geliştirilmiştir.

**3.1.1 Python**

İlk sürümü 1991 yılında geliştirilmiş**[3]** , nesne yönelimli, yorumlamalı, birimsel (modüler) ve etkileşimli yüksek seviyeli bir programlama dilidir.**[4]**

Girintilere dayalı basit söz dizimi, dilin öğrenilmesini ve akılda kalmasını kolaylaştırır. Bu da ona söz diziminin ayrıntıları ile vakit yitirmeden programlama yapılmaya başlanabilen bir dil olma özelliği kazandırır.**[4]**

**3.1.2 TensorFlow**

TensorFlow, makine öğrenimi için ücretsiz ve açık kaynaklı bir yazılım kütüphanesidir. Bir dizi görevde kullanılabilir, ancak derin sinir ağlarının eğitimi ve çıkarımına özel olarak odaklanmaktadır.**[5]**

Tensorflow, veri akışına ve türevlenebilir programlamaya dayalı sembolik bir matematik kitaplığıdır. Google'da hem araştırma hem de üretim için kullanılmaktadır.TensorFlow, Google Brain ekibi tarafından Google'ın iç işlerinde kullanımı için geliştirilmiştir. 2015 yılında Apache License 2.0 sürümü altında piyasaya sürülmüştür.TensorFlow, Google Brain'in ikinci nesil sistemidir. 1.0.0 sürümü 11 Şubat 2017'de yayınlanmıştır. Referans uygulama tek cihazlarda çalışırken, TensorFlow birden fazla CPU ve GPU üzerinde çalışabilmektedir (grafik işleme birimlerinde genel amaçlı bilgi işlem için isteğe bağlı CUDA ve SYCL uzantıları ile birlikte). TensorFlow, 64 bit Linux, macOS, Windows ve Android ve iOS dahil mobil bilgi işlem platformlarında mevcuttur.**[5]**

**3.1.3 Keras**

Keras, Python'da yazılmış açık kaynaklı bir sinir ağı kütüphanesidir. Keras TensorFlow, Microsoft Cognitive Toolkit, R, Theano veya PlaidML ile beraber çalışabilir. Derin sinir ağları ile hızlı deney yapabilmek için tasarlanan bu cihaz kullanıcı dostu, modüler ve genişletilebilir olmaya odaklanıyor. ONEIROS (Açık Uçlu Nöro-Elektronik Akıllı Robot İşletim Sistemi), projesinin araştırma çabalarının bir parçası olarak geliştirilmiştir ve ana yazarı ve sürdürücüsü Google mühendisi François Chollet'tir. Chollet ayrıca XCeption derin sinir ağı modelinin yazarıdır.**[6]**

2017 yılında Google'ın TensorFlow ekibi, TensorFlow'un çekirdek kütüphanesinde Keras'ı desteklemeye karar verdi. Chollet, Keras'ın bağımsız bir makine öğrenimi çerçevesi yerine bir arayüz olarak tasarlandığını açıkladı. Kullanılan hesaplamalı arka uçtan bağımsız olarak derin öğrenme modelleri geliştirmeyi kolaylaştıran daha üst düzey, daha sezgisel bir soyutlama seti sunar. Microsoft, CNTK v2.0'dan itibaren mevcut olan Keras'a bir CNTK arka ucu ekledi.**[6]**

**3.1.4 CUDA**

CUDA (Compute Unified Device Architecture), GPU (Graphics Processing Unit) için NVIDIA'nın sunduğu C programlama dili üzerinde eklenti olarak kullanıma sunulan bir mimari ve teknolojidir.**[7]**

PathScale tabanlı bir C derleyicisi ve C ile yazılmış algoritmaların GPU üzerinde çalışmasını sağlayan geliştirme araçları kümesidir. CUDA, nVidia tarafından geliştirilmiştir ve çalışması için nVidia GPU ve son sürüm driverlara ihtiyaç vardır. CUDA G8X üzeri, GeForce, Quadro ve Tesla'yı içeren her GPU da çalışır. nVidia, ekran kartı mimarilerinin ileriye doğru kod uyumluluğu sayesinde, Geforce 8 için geliştirilen programların herhangi bir düzeltme yapılmadan gelecek nesil ekran kartlarında hızlanmalardan otomatik olarak faydalanacak şekilde kullanılabileceğini belirtiyor. CUDA kütüphanesi, geliştiricilerin CUDA özellikli GPU'lar üzerindeki hafızalara ve Stream Processorlara hükmedebilmesini sağlar. İlk CUDA Geliştirici seti (SDK) 15 Şubat 2007 de yayınlandı.**[7]**

**3.1.5 cuDNN**

NVIDIA CUDA® Derin Sinir Ağı kütüphanesi (cuDNN), derin sinir ağları için GPU ile hızlandırılmış bir ilkel kütüphanedir. cuDNN, ileri ve geri evrişim, havuzlama, normalleştirme ve etkinleştirme katmanları gibi standart rutinler için yüksek düzeyde ayarlanmış uygulamalar sağlar.**[8]**

Dünya çapındaki derin öğrenme araştırmacıları ve çerçeve geliştiricileri, yüksek performanslı GPU hızlandırma için cuDNN'yi tercih ediyor. Aynı zamanda cuDNN, araştırmacıların düşük seviyeli GPU performans ayarına zaman harcamak yerine sinir ağlarını eğitmeye ve yazılım uygulamaları geliştirmeye odaklanmalarını sağlar. cuDNN, Caffe2, Chainer, Keras, MATLAB, MxNet, PaddlePaddle, PyTorch ve TensorFlow dahil olmak üzere yaygın olarak kullanılan derin öğrenme framework’lerini hızlandırır. Frameworkler ile entegre edilmiş cuDNN'ye sahip NVIDIA tarafından optimize edilmiş derin öğrenme framework kapsayıcılarına erişim için, daha fazla bilgi edinmek ve başlamak üzere NVIDIA GPU CLOUD'u ziyaret edebilirsiniz.**[8]**

**3.1.6 NIH Chest X-Rays**

Göğüs röntgeni muayeneleri, mevcut en sık ve uygun maliyetli tıbbi görüntüleme muayenelerinden biridir. Bununla birlikte, göğüs röntgeninin klinik teşhisi zor olabilir ve bazen göğüs BT görüntülemesi ile teşhis etmekten daha zor olabilir. Ek açıklamaları olan, kamuya açık büyük veri kümelerinin olmaması, göğüs röntgenleri ile gerçek dünyadaki tıbbi sitelerde klinik olarak ilgili bilgisayar destekli tespit ve teşhis elde etmenin imkansız olmasa da çok zor olduğu anlamına gelir. Büyük X-Ray görüntü veri kümeleri oluşturmanın önündeki en büyük engellerden biri, bu kadar çok görüntüyü etiketlemek için kaynak eksikliğidir. Bu veri setinin yayınlanmasından önce, Openi, mevcut 4.143 görüntü ile göğüs röntgeni görüntülerinin halka açık en büyük kaynağıydı.**[2]**

NIH Göğüs Röntgeni Veri Kümesi, 30.805 benzersiz hastadan hastalık etiketli 112.120 X-ray görüntüsünden oluşur. Bu etiketleri oluşturmak için yazarlar, ilişkili radyolojik raporlardan hastalık sınıflandırmalarını metin haline getirmek için Doğal Dil İşleme'yi kullandılar. Etiketlerin >%90 doğru olması ve zayıf denetimli öğrenme için uygun olması bekleniyor.**[2]**

**3.1.7 Anaconda**

Anaconda ücretsiz ve açık kaynaklı, Python ve R programlama dillerinin bilimsel hesaplama kullanımında paket yönetimini kolaylaştırmayı amaçlayan bir özgür ve açık kaynaklı dağıtımdır. Paket sürümleri conda paket yönetim sistemi ile yönetilir.**[9]**

Anaconda dağıtımı Windows, Linux ve MacOS işletim sistemlerinde kullanılabilen veri bilimi paketleri içerir.Anaconda dağıtımı, 1.500'den fazla paketin yanı sıra conda paketi ve sanal çevre yöneticisiyle birlikte gelir Ayrıca komut satırı arabirimine (CLI) grafiksel bir alternatif olarak bir GUI, Anaconda Navigator içerir.**[9]**

Conda ve pip paket yöneticisi arasındaki en büyük fark, paket gereksinimlerinin yönetim farkıdır, bu da Python veri bilimi ve condanın varlığının nedeni için önemli bir zorluktur.**[9]**

**3.1.8 Spyder**

Spyder , Python dilinde bilimsel programlama için açık kaynaklı , platformlar arası entegre bir geliştirme ortamıdır (IDE) . Spyder, NumPy , SciPy , Matplotlib , pandas , IPython , SymPy ve Cython'un yanı sıra diğer açık kaynaklı yazılımlar dahil olmak üzere bilimsel Python yığınındaki bir dizi önemli paketle entegre olur. MIT lisansı ile yayınlanmıştır.**[10]**

İlk olarak 2009 yılında Pierre Raybaut tarafından yaratılan ve geliştirilen Spyder, 2012'den beri bilimsel Python geliştiricilerinden oluşan bir ekip ve topluluk tarafından sürdürüldü ve sürekli olarak geliştirildi.**[10]**

Spyder birinci taraf ve üçüncü taraf eklentilerle genişletilebilir, veri denetimi için etkileşimli araçlar için destek içerir ve Pyflakes, Pylint ve Rope gibi Python'a özgü kod kalite güvencesi ve iç gözlem araçlarını içerir . Anaconda aracılığıyla , Windows'ta, macOS'ta MacPorts aracılığıyla ve Arch Linux , Debian , Fedora , Gentoo Linux , openSUSE ve Ubuntu gibi büyük Linux dağıtımlarında çapraz platform olarak kullanılabilir.**[10]**

**3.2 Yöntem**

**3.2.1 Genel Yöntem**

**3.2.1.1 Yapay Zeka**

İngilizce “artificial intelligence” tanımlamasının dilimizdeki tercümesi olarak kullanılan yapay zeka terimi, teknoloji dünyasında ise “AI” şeklindeki bir kısaltma ile tabir edilir. En kısa açıklama ile “insan beyninin ve düşünme sisteminin kopya edilmesi” olarak tanımlanabilir. Tanımı biraz daha sade bir şekilde yapmak gerekirse; “bir bilgisayar programını insan gibi düşünebilecek hâle getirme” cümlesi kullanılabilir.**[11]**

Yapay zeka konusunda araştırmalar yapılırken sıklıkla “makine öğrenmesi” ve “derin öğrenme” gibi tanımlarla karşılaşılır. İngilizcede “machine learning” olarak tanımlanan makine öğrenmesi ise yapay zekayı oluşturan unsurlardan biridir. Yine İngilizcede “deep learning” şeklinde ifade edilen ve dilimize “derin öğrenme” olarak tercüme edilebilen teknoloji ise makine öğrenmesinin unsurlarından biridir.**[11]**

**3.2.1.2 Makine Öğrenmesi**

Makine öğrenimi veya makine öğrenmesi, bilgisayarların algılayıcı verisi ya da veritabanları gibi veri türlerine dayalı öğrenimini olanaklı kılan algoritmaların tasarım ve geliştirme süreçlerini konu edinen bir bilim dalıdır. Makine öğrenimi araştırmalarının odaklandığı konu bilgisayarlara karmaşık örüntüleri algılama ve veriye dayalı akılcı kararlar verebilme becerisi kazandırmaktır. Bu, makine öğreniminin istatistik, olasılık kuramı, veri madenciliği, örüntü tanıma, yapay zekâ, uyarlamalı denetim ve kuramsal bilgisayar bilimi gibi alanlarla yakından ilintili olduğunu göstermektedir.**[12]**

Makine öğrenimi algoritmaları, insan müdahalesi olmadan verilerden öğrenebilen ve deneyimler ile geliştirebilen programlardır. Makine öğrenimi algoritmaları denetimli öğrenme ve denetimsiz öğrenme olarak kategorize edilir.**[13]**

**3.2.1.2.1 Denetimli Öğrenme**

Denetimli öğrenmede, veri kümemizin ne olduğunu ve bu verilerden istediğimiz çıktının ne olması gerektiğini biliriz.**[14]**

Denetimli öğrenme, verileri ve o verilerden çıkan sonuçları makineye tekrar baştan vererek bu bilgilerden bir fonksiyon (giriş verileri ile sonuç verileri arasında bir eşleşme) çıkartılmasının sağlamaktadır. Böylece makine veriler arasındaki ilişkiyi öğrenmektedir. Denetimli öğrenme problemleri “regresyon” ve “sınıflandırma” olarak ikiye ayrılır.**[14]**

Bir regresyon probleminde, sonuçları sürekli bir çıktı içinde tahmin etmeye çalışırız; yani, girdi değişkenlerini bazı sürekli fonksiyonlara eşlemeye çalışırız.**[14]**

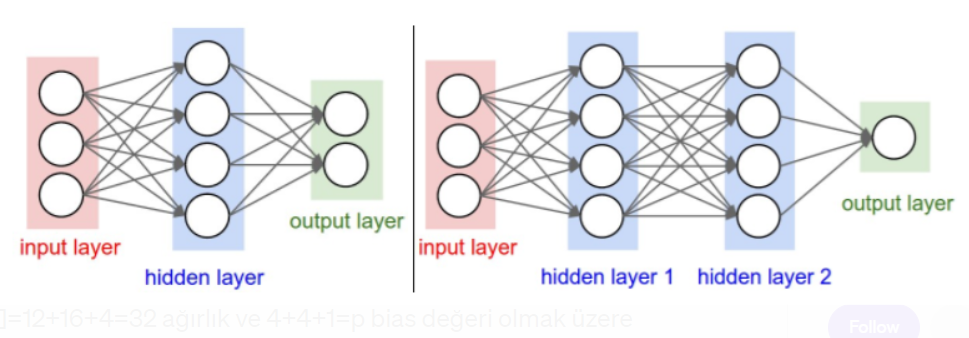
Bir sınıflandırma probleminde, sonuçları ayrı ayrı çıktılarda tahmin etmeye çalışırız. Yani, girdi değişkenlerini ayrı kategorilere atamaya çalışırız.**[14]**

**3.2.1.3 Yapay Sinir Ağları**

Yapay sinir ağları (YSA), insan [beyninin](https://tr.wikipedia.org/wiki/Beyin) bilgi işleme tekniğinden esinlenerek geliştirilmiş bir bilgi işlem teknolojisidir. YSA ile basit [biyolojik](https://tr.wikipedia.org/wiki/Biyolojik) sinir sisteminin çalışma şekli taklit edilir. Yani biyolojik [nöron](https://tr.wikipedia.org/wiki/N%C3%B6ron) hücrelerinin ve bu hücrelerin birbirleri ile arasında kurduğu sinaptik bağın dijital olarak modellenmesidir. Nöronlar çeşitli şekillerde birbirlerine bağlanarak ağlar oluştururlar. Bu ağlar öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiptirler. Diğer bir ifadeyle, YSA'lar, normalde bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir. Bir insanın, düşünme ve gözlemleme yeteneklerini gerektiren problemlere yönelik çözümler üretebilmesinin temel sebebi ise insan beyninin ve dolayısıyla insanın sahip olduğu yaşayarak veya deneyerek öğrenme yeteneğidir.**[15]**

Biyolojik sistemlerde öğrenme, nöronlar arasındaki [sinaptik](https://tr.wikipedia.org/wiki/Sinaptik) (synaptic) bağlantıların ayarlanması ile olur. Yani, insanlar doğumlarından itibaren bir yaşayarak öğrenme süreci içerisine girerler. Bu süreç içinde beyin sürekli bir gelişme göstermektedir. Yaşayıp tecrübe ettikçe sinaptik bağlantılar ayarlanır ve hatta yeni bağlantılar oluşur. Bu sayede öğrenme gerçekleşir. Bu durum YSA için de geçerlidir. Öğrenme, eğitme yoluyla örnekler kullanarak olur; başka bir deyişle, gerçekleşme girdi/çıktı verilerinin işlenmesiyle, yani eğitme algoritmasının bu verileri kullanarak bağlantı ağırlıklarını (weights of the synapses) bir yakınsama sağlanana kadar, tekrar tekrar ayarlamasıyla olur.**[15]**

YSA'lar, ağırlıklandırılmış şekilde birbirlerine bağlanmış birçok işlem biriminden (nöronlar) oluşan [matematiksel](https://tr.wikipedia.org/wiki/Matematik) sistemlerdir. Bir işlem birimi, aslında sık sık transfer fonksiyonu olarak anılan bir denklemdir. Bu işlem birimi, diğer nöronlardan sinyalleri alır; bunları birleştirir, dönüştürür ve sayısal bir sonuç ortaya çıkartır. Genelde, işlem birimleri kabaca gerçek nöronlara karşılık gelirler ve bir ağ içinde birbirlerine bağlanırlar; bu yapı da sinir ağlarını oluşturmaktadır.**[15]**



Şekil 2. Tek Gizli Katmanlı ve Çok Gizli Katmanlı Sinir Ağı Yapısı

**3.2.1.4 Derin Öğrenme**

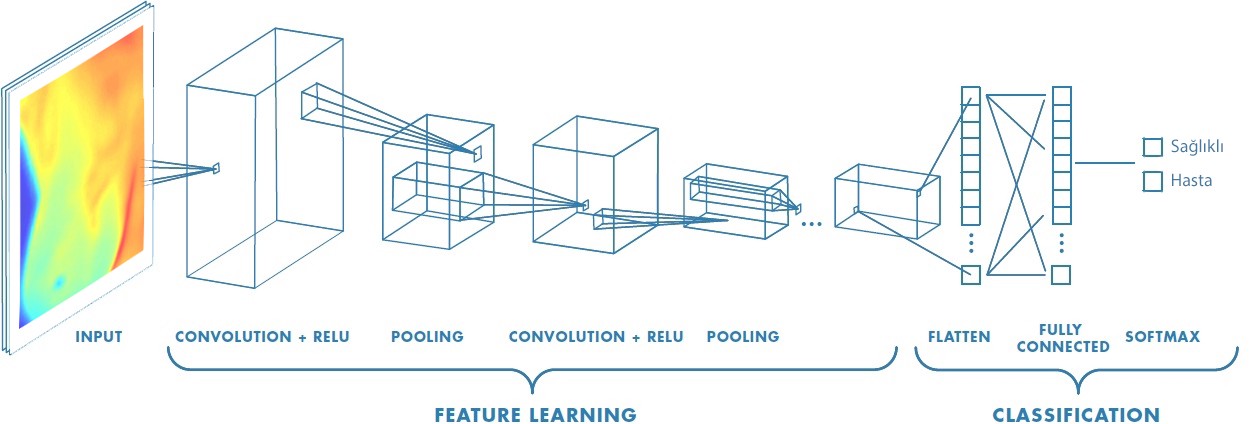
Derin öğrenme (aynı zamanda derin yapılandırılmış öğrenme, hiyerarşik öğrenme ya da derin makine öğrenmesi) bir veya daha fazla gizli katman içeren [yapay sinir ağları](https://tr.wikipedia.org/wiki/Yapay_sinir_a%C4%9Flar%C4%B1) ve benzeri [makine öğrenme](https://tr.wikipedia.org/wiki/Makine_%C3%B6%C4%9Frenimi) [algoritmalarını](https://tr.wikipedia.org/wiki/Algoritma) kapsayan çalışma alanıdır.**[16]**

Yani en az bir adet [yapay sinir ağının](https://tr.wikipedia.org/wiki/Yapay_sinir_a%C4%9Flar%C4%B1) ([YSA](https://tr.wikipedia.org/wiki/YSA)) kullanıldığı ve birçok algoritma ile, bilgisayarın eldeki verilerden yeni veriler elde etmesidir.**[16]**

Derin öğrenme [gözetimli](https://tr.wikipedia.org/wiki/G%C3%B6zetimli_%C3%B6%C4%9Frenme), yarı gözetimli veya [gözetimsiz](https://tr.wikipedia.org/wiki/G%C3%B6zetimsiz_%C3%B6%C4%9Frenme) olarak gerçekleştirilebilir. Derin yapay sinir ağları [pekiştirmeli öğrenme](https://tr.wikipedia.org/wiki/Peki%C5%9Ftirmeli_%C3%B6%C4%9Frenme) yaklaşımıyla da başarılı sonuçlar vermiştir. Yapay sinir ağları, biyolojik sistemlerdeki bilgi işleme ve dağıtılmış iletişim düğümlerinden esinlenilmiştir. Yapay sinir ağlarının biyolojik beyinlerden çeşitli farklılıkları vardır. Özellikle, sinir ağları statik ve sembolik olma eğilimindeyken, çoğu canlı organizmanın biyolojik beyni dinamik(plastik) ve analogtur.**[16]**

Derin öğrenmede çok sayıda veri girişiyle ayırt edici özellikleri kendisi öğrenir. Öğrenme işlemini için ne kadar çok veri girişi olursa o kadar çok başarılı olunur. Veriler birden çok katmandan geçer. Üst katmanlar daha çok ayrıntı çıkaran katmanlardır **[17]**.

**3.2.1.4.1 CNN**



Şekil 3. CNN Mimarisi

Evrişimsel sinir ağları, [derin öğrenmenin](https://tr.wikipedia.org/wiki/Derin_%C3%B6%C4%9Frenme) bir alt dalıdır ve genellikle görsel bilginin analiz edilmesinde kullanılır. Yaygın kullanım alanları [resim ve video tanıma](https://tr.wikipedia.org/wiki/G%C3%B6r%C3%BCnt%C3%BC_tan%C4%B1ma), [önerici sistemler](https://tr.wikipedia.org/wiki/%C3%96nerici_sistem) [resim sınıflandırma](https://tr.wikipedia.org/wiki/Bilgisayarl%C4%B1_g%C3%B6r%C3%BC), [tıbbi görüntü analizi](https://tr.wikipedia.org/w/index.php?title=T%C4%B1bbi_g%C3%B6r%C3%BCnt%C3%BC_analizi&action=edit&redlink=1) ve [doğal dil işleme](https://tr.wikipedia.org/wiki/Do%C4%9Fal_dil_i%C5%9Fleme) olarak sıralanabilir.**[18]**

CNN genellikle görüntü işlemede kullanılan ve girdi olarak görselleri alan bir [derin öğrenme](https://teknoloji.org/derin-ogrenme-nedir-yapay-sinir-aglari-ne-ise-yarar/) algoritmasıdır. Farklı operasyonlarla görsellerdeki featureları (özellikleri) yakalayan ve onları sınıflandıran bu algoritma farklı katmanlardan oluşmaktadır. Convolutional Layer, Pooling ve Fully Connected , Flatten olan bu katmanlardan geçen görsel, farklı işlemlere tabii tutularak derin öğrenme modeline girecek kıvama gelir. CNN modelleri oluştururken, unstructural (düzensiz) veri ile uğraştığımızdan klasik [makine öğrenmesi](https://teknoloji.org/makine-ogrenmesi-nedir-makine-ogrenmesi-algoritmalari/) algoritmalarına kıyasla veri ön işleme kısmında çok uğraşmamaktayız.**[19]**

**Convolutional Layer**

Convolutional (evrişim katmanı) CNN algoritmalarında görüntüyü ele alan ilk katmandır. Bilindiği üzere görseller aslında içlerinde belirli değerler taşıyan piksellerden oluşan matrislerdir. Evrişim katmanında da orijinal görsel boyutlarından daha küçük bir filtre görselin üzerinde gezer ve bu görsellerden belirli özellikleri yakalamaya çalışır.**[20]**

### **Pooling**

Evrişimli katman gibi pooling (havuzlama) katmanı da boyutsallığı azaltma amacındadır. Bu sayede hem gereken işlem gücü azalır hem de yakalanan gereksiz özellikler yok sayılarak daha önemli özelliklere odaklanılır.**[20]**

CNN modellerinde genellikle kullanılan iki farklı pooling tekniği vardır. Bunlardan biri Max (Maksimum) diğeri de Average (Ortalama) pooling’tir.**[20]**

### **Fully Connected Layer (Tam Bağlantılı Katman)**

Fully Connected katmanda birkaç kez evrişimli katmandan ve pooling katmanından geçen ve matris halinde olan görselimiz düz bir vektör haline getirilir.**[20]**

Girdi resmimizi sinir ağları ile eğitebileceğimiz kıvama getirdikten sonra geriye sadece klasik sinir ağlarındaki çalışma mantığı kalıyor. Yine katmanlardaki nodelarda (düğüm) özellikler tutuluyor ve weight (ağırlık) ve bias değiştirilerek öğrenme sürecine giriliyor.**[20]**

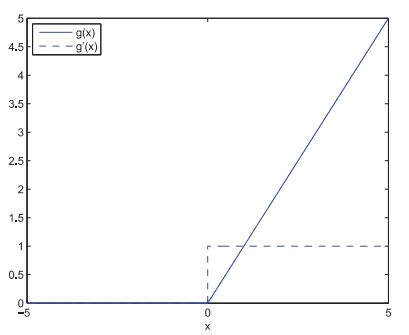
**Flatten**

Convolution ve pooling işlemlerinden sonra ortaya çıkan matrisleri n satır 1 sütundan oluşan vektörlere dönüştürme işlemine flattening deniyor. Bu vektörler artificial neural network’ün inputları olacaktır.**[21]**

**ReLU**

Genellikle Convolutional Neural Network (CNN)’te ve ara katmanlarda çok sık kullanılan ReLU fonksiyonunun ana avantajı aynı anda tüm nöronları aktive etmemesidir. Yani bir nöron negatif değer üretirse, aktive edilmeyeceği anlamına gelir.**[21]**

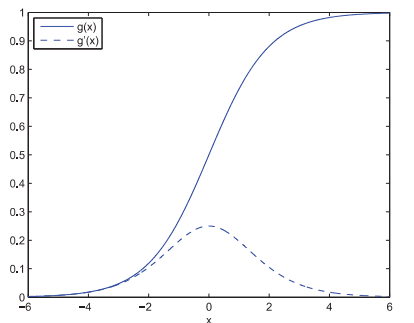
Negatif değerler üreten nöronlar sıfır değerini alır. Bu durum, ReLU’nun Hiperbolik Tanjant ve Sigmoid fonksiyonundan daha verimli ve hızlı çalışmasını sağlar. Bu nedenle ReLU, çok katmanlı sinir ağlarında daha çok tercih edilir.**[21]**



Şekil 4. ReLU Fonksiyon Grafiği

**Sigmoid**

En çok kullanılan aktivasyon fonksiyonlarından birisidir. Şimdiye kadar bahsettiğimiz ilk doğrusal olmayan(non-linear) fonksiyondur. Karar vermeye yönelik olasılıksal bir yaklaşımdır ve değer aralığı [0,1] arasındadır. Yani çıktının hangi sınıfa ait olduğuna dair olasılıksal bir değer verir bize. Ve iyi haber, Sigmoid fonksiyonu sürekli yani türevlenebilir bir fonksiyon olduğundan öğrenme işlemi gerçekleşir.**[21]**



Şekil 5. Sigmoid Fonksiyon Grafiği

**3.2.1.4.2 Sequential**

Sequential, Keras kütüphanesinde model oluşturmak için kullanılan dinamik bir sınıftır. Bu sınıf kendisine eklenen sıralı yapıları kaydederek, dinamik ve özgün CNN mimarileri oluşturmamıza olanak sağlar.

**3.2.1.4.3 Image Data Generator**

Image Data Generator verilen görüntüleri, verilen parametre değerlerine göre işleyip modele sunmak için kullanılan Keras kütüphanesine ait bir sınıftır. Bu sınıf veri sayısını artırmanın yanında overfitting’i (aşırı öğrenme / ezberleme) engelleme amacıyla da kullanılır. Bunu resmi çevirme ve yakınlaştırma-uzaklaştırma gibi işlemlere tabii tutarak yapabilir.

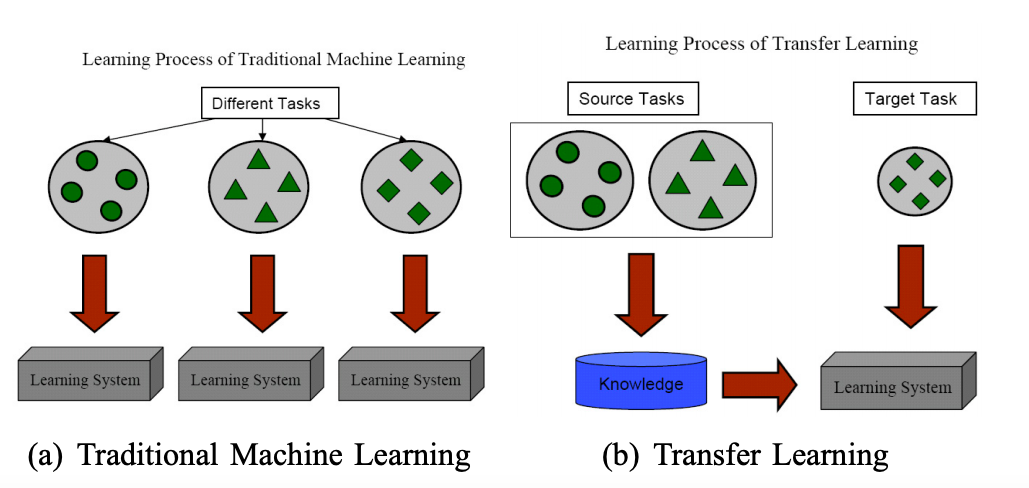
**3.2.1.5 Transfer Öğrenme**

Transfer öğrenmesi, bir problemi çözerken elde edilen bilgiyi saklamak ve daha sonra farklı ama ilgili bir probleme uygulamak üzerine odaklanan makine öğreniminde bir araştırma problemidir. Örneğin, otomobilleri tanımayı öğrenirken kazanılan bilgi, kamyonları tanımaya çalışırken uygulanabilir. Bu alandaki araştırmalar, iki alan arasındaki resmi bağlar sınırlı olmakla birlikte, öğrenmenin aktarımı üzerine psikolojik literatürün uzun tarihi ile bazı ilişkileri taşımaktadır.**[22]**

Transfer öğrenimi, bir görev için eğitilmiş bir modelin, ilgili ikinci bir görevde yeniden tasarlandığı bir makine öğrenmesi tekniğidir.**[22]**

Öğrenme ve alan adı uyarlamasını aktarma, bir ortamda öğrenilenin başka bir ortamda genelleşmeyi iyileştirmek için kullanılmasıyla ilgilidir.**[22]**

<https://dergipark.org.tr/en/download/article-file/1643941> bağlantısında bulunan makalede görüleceği üzere DenseNet121 mimarisi biyomedikal veriler üzerinde diğer modellere kıyasla daha başarılıdır. Dolayısıyla uygulamada DenseNet121 modeli kullanılarak transfer öğrenme gerçekleştirilecektir.



Şekil 6. Geleneksel Makine Öğrenmesi ile Transfer Öğrenmenin Farkını Gösteren Örnek

**3.2.1.6 Hiperparametreler**

Hiperparametreler, ağ yapısını (Örn: gizli birimlerin sayısı) ve ağın nasıl eğitildiğini belirleyen değişkenleri belirleyen değişkenlerdir(Örn: öğrenme hızı).**[23]**

Hiperparametreler eğitimden önce ayarlanır(ağırlıkları ve sapmaları optimize etmeden önce).

Bu uygulamada gizli katmanlar, dropout, aktivasyon fonksiyonları, epoch sayısı, batch boyutu, batch normalization, optimizasyon algoritması ve loss fonksiyonu kullanılan hiperparametreler arasında yer almaktadır.**[23]**

**Gizli Katmanlar**

Giriş katmanı ve çıkış katmanı arasındaki katmanlardır.**[23]**

**Dropout**

Dropout, overfittingi(aşırı uydurmayı) önlemek için (validation doğruluğunu arttırmak) ve genelleştirme gücünü arttırmak için düzenli bir tekniktir.**[23]**

**Aktivasyon Fonksiyonları**

Aktivasyon fonksiyonları, doğrusal olmayan kestirim sınırlarını öğrenmek için derin öğrenme modellerine izin veren modellere doğrusal olmayanlığı tanıtmak için kullanılır.**[23]**

**Epoch Sayısı**

Epoch(döngü) sayısı, eğitim sırasında tüm eğitim verilerinin ağa gösterilme sayısıdır.**[23]**

**Batch Boyutu**

Mini batch size(boyutu), parametre güncellemesinin gerçekleştiği ağa verilen alt örneklerin sayısıdır.**[23]**

**Batch Normalization**

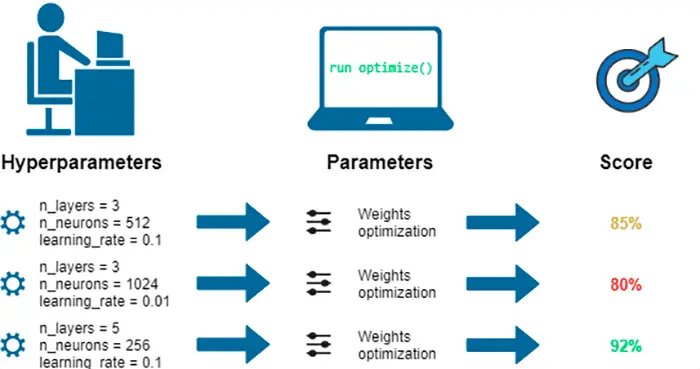
Batch Normalization ile ağdaki katmanlar, önceki katmanın öğrenmesini beklemek zorunda kalmaz. Eş zamanlı olarak öğrenime olanak sağlar. Eğitimin hızlanmasını sağlar.**[24]**

**Optimizasyon Algoritması**

Derin öğrenme uygulamalarında, öğrenme işleminin sağlıklı bir şekilde sonuçlanması için hata fonksiyonunun mutlak minimum değerinin bulunması gerekmektedir. Bu işlem, optimizasyon yöntemleri kullanılarak gerçekleştirilmektedir. Optimizasyon, ağın ürettiği çıkış değeri ile gerçek değerarasındaki farkı yani hatayı en küçük yapmak için kullanılan yöntemlerdir.**[25]**

**Loss Fonksiyonu**

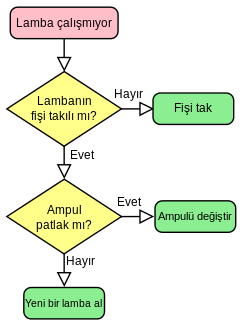
Matematiksel optimizasyon ve karar teorisinde , bir kayıp fonksiyonu veya maliyet fonksiyonu (bazen bir hata fonksiyonu olarak da adlandırılır), bir veya daha fazla değişkenin bir olayını veya değerlerini gerçek bir sayı üzerine eşleyen bir fonksiyondur . olay. Bir optimizasyon problemi , bir kayıp fonksiyonunu en aza indirmeye çalışır. Bir amaç işlevi ya bir kayıp işlevidir ya da bunun tersidir, bu durumda maksimize edilecektir. Kayıp fonksiyonu, hiyerarşinin çeşitli seviyelerindeki terimleri içerebilir.**[26]**



Şekil 7. Hiperparametrelerin Model Sonuçlarına Etkilerinin Örneklendirilmesi

**3.2.1.7 Algoritmalar**

Algoritma, belli bir problemi çözmek veya belirli bir amaca ulaşmak için tasarlanan yoldur. Matematikte ve bilgisayar biliminde bir işi yapmak için tanımlanan, bir başlangıç durumundan başladığında, açıkça belirlenmiş bir son durumunda sonlanan, sonlu işlemler kümesidir. Genellikle bilgisayar programlamada kullanılır ve tüm programlama dillerinin temeli algoritmaya dayanır. Aynı zamanda algoritma tek bir problemi çözecek davranışın, temel işleri yapan komutların veya deyimlerin adım adım ortaya konulmasıdır ve bu adımların sıralamasına dikkat edilmelidir. Bir problem çözülürken algoritmik ve sezgisel olmak üzere iki yaklaşım vardır. Algoritmik yaklaşımda da çözüm için olası yöntemlerden en uygun olan seçilir ve yapılması gerekenler adım adım ortaya konulur.**[27]**



Şekil 8. Çalışmayan Bir Lambayı Çalışır Hale Getiren Algoritma

Uygulamada kullanılan algoritma veriseti içerisinden belirlenmiş hastalıkları seçen ve alt klasörlere ayırıp CNN modeline sunmaya hazır hale getirme işlemini yapan bir algoritmadır.

**3.2.2 Uygulama**

**3.2.2.1 Verisetlerinin Hazırlanması**

**3.2.2.1.1 4 Sınıflı Model**

Hazırlanan algoritma ile Pneumothorax, Nodule, Mass ve Hastalıksız röntgen filmleri test, validation ve train klasörlerine ayrıştırıldı. Ayrıştırılan her röntgen sınıfı, train için 750, test ve validation için ise 250’şer adet röntgen filmi içermektedir. Toplamda train için 3000, validation ve test için ise 1000’er adet röntgen filmi hazırlanmış oldu.

**3.2.2.1.2 3 Sınıflı Model**

Hazırlanan algoritma ile Atelectasis, Infiltration ve Hastalıksız röntgen filmleri test, validation ve train klasörlerine ayrıştırıldı. Ayrıştırılan her röntgen sınıfı, train için 1000, test ve validation için ise 300’er adet röntgen filmi içermektedir. Toplamda train için 3000, validation ve test için ise 900’er adet röntgen filmi hazırlanmış oldu.

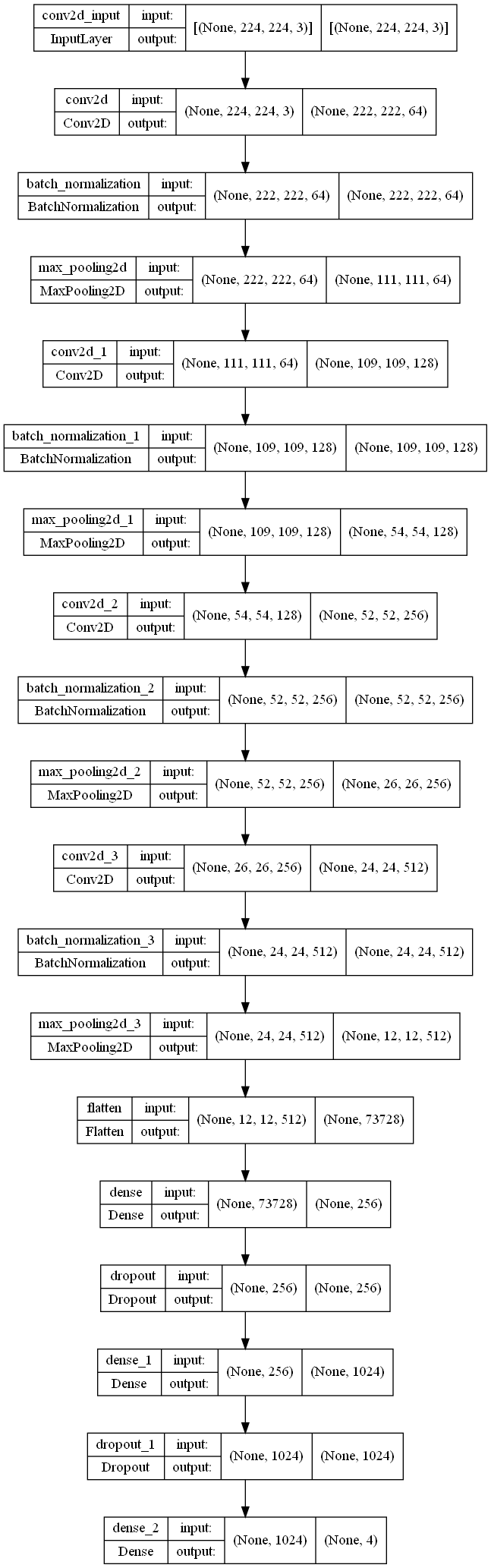
**3.2.2.1.3 Binary Model**

Hazırlanan algoritma ile Infiltration ve Hastalıksız röntgen filmleri test, validation ve train klasörlerine ayrıştırıldı. Ayrıştırılan her röntgen sınıfı, train için 2500, test ve validation için ise 1000’er adet röntgen filmi içermektedir. Toplamda train için 5000, validation ve test için ise 2000’er adet röntgen filmi hazırlanmış oldu.

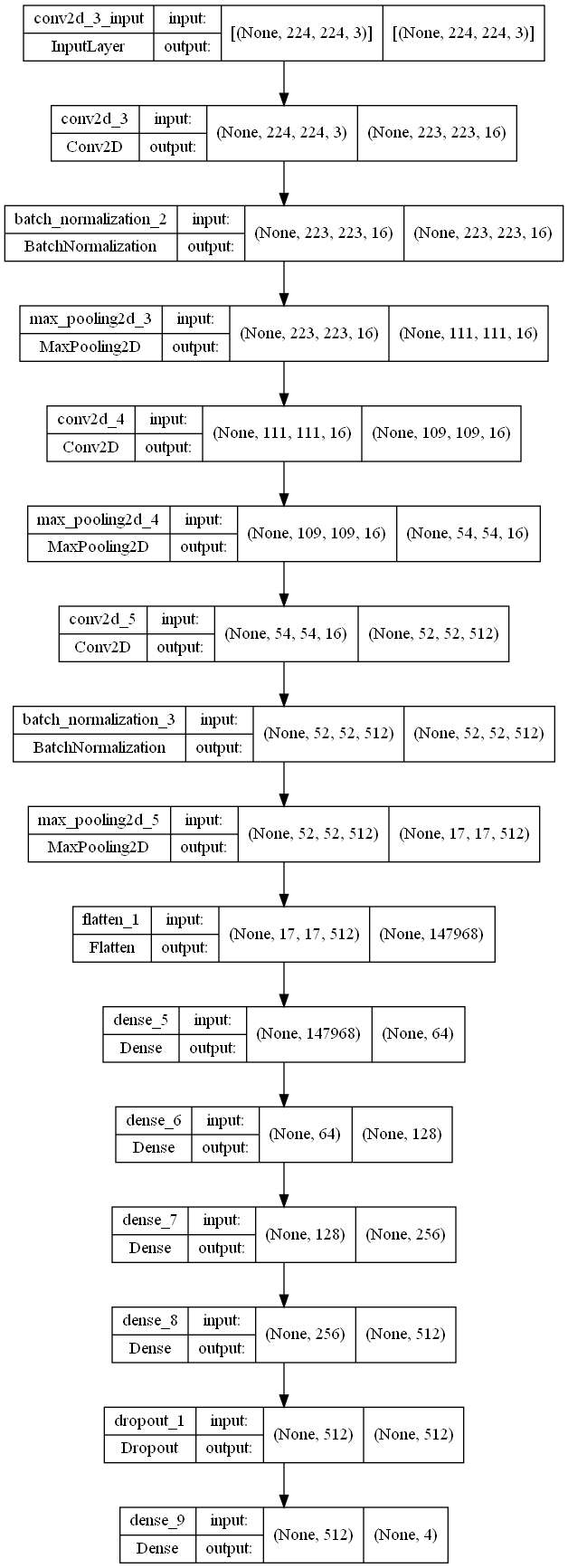
**3.2.2.2 CNN Modellerinin Hazırlanması**

**3.2.2.2.1 4 Sınıflı Model**

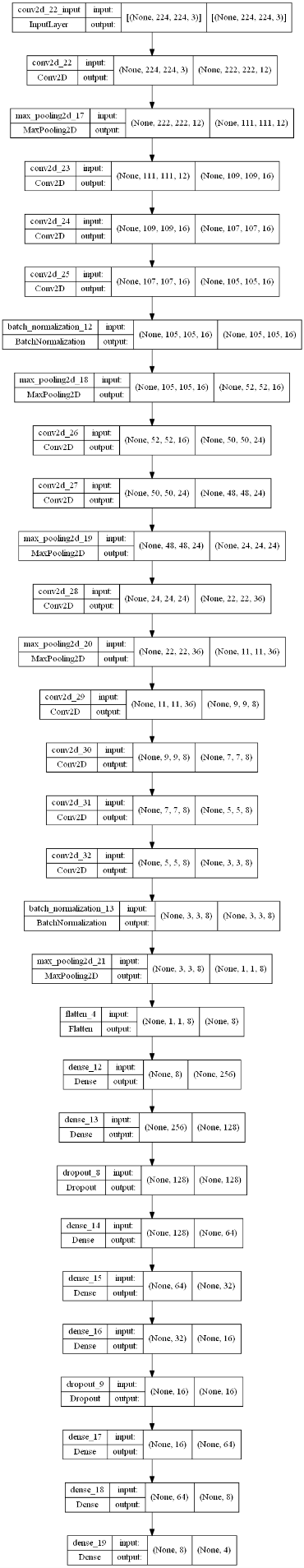
Birbirinden farklı 4 CNN ve 3 farklı transfer öğrenme modeli oluşturuldu. Oluşturulan modeller 3.2.2.1.1’de oluşturulan veriseti ile eğitilerek validation accuracy – accuracy ve validation loss – loss grafikleri çizildi. Modellerin mimarilerinin görüntülenmesi ve incelenmesi için şablonlar oluşturuldu. Oluşturulan tüm modeller test seti ile değerlendirilip, modeller arası karşılaştırma için loss ve accuracy grafiği oluşturuldu.



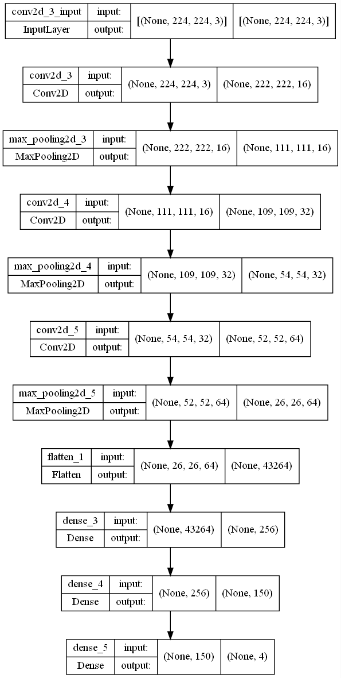
Şekil 9. Model – 1 CNN Mimarisi



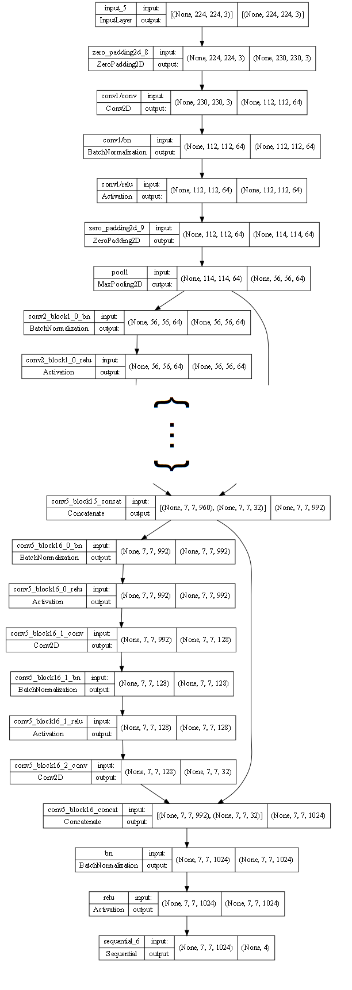
Şekil 10. Model – 2 CNN Mimarisi



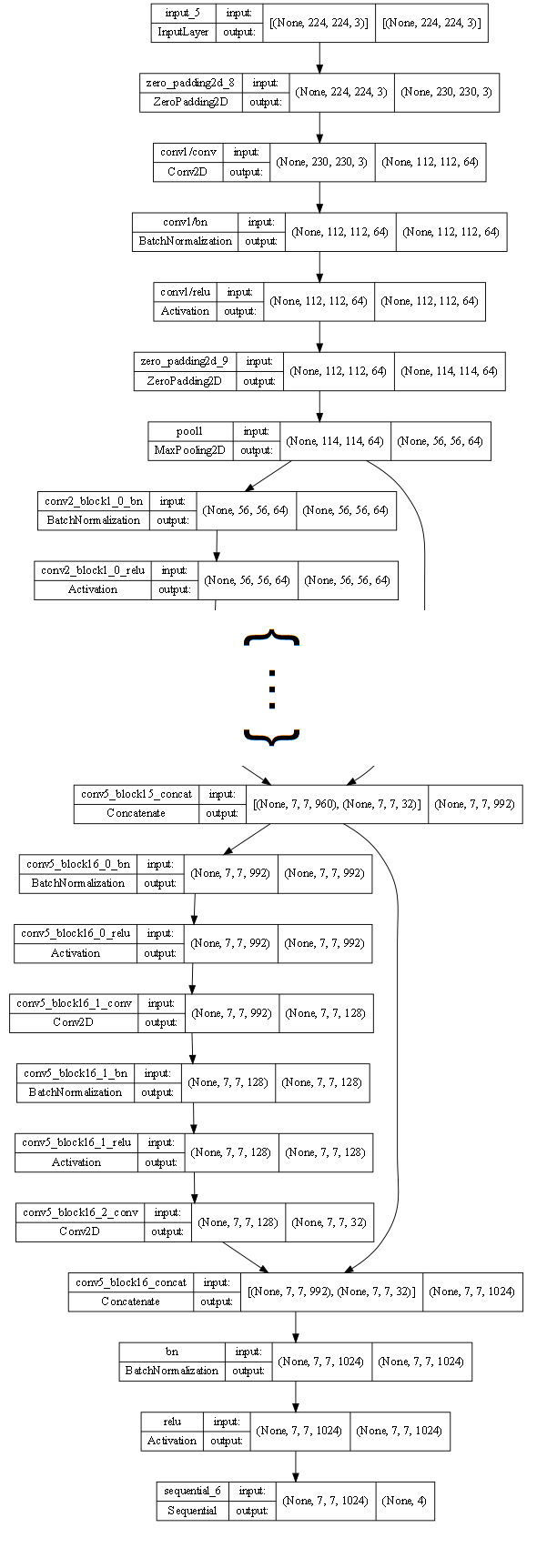
Şekil 11. Model – 3 CNN Mimarisi



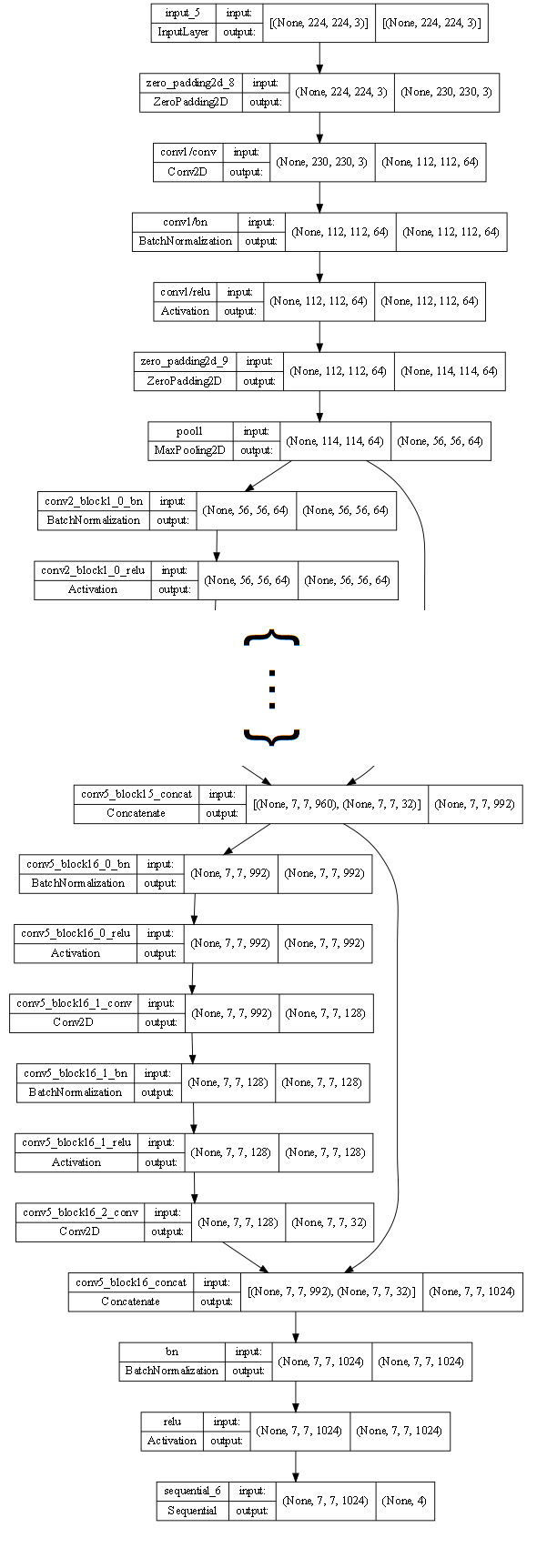
Şekil 12. Model – 4 CNN Mimarisi



Şekil 13. Transfer Model – 1 CNN Mimarisi



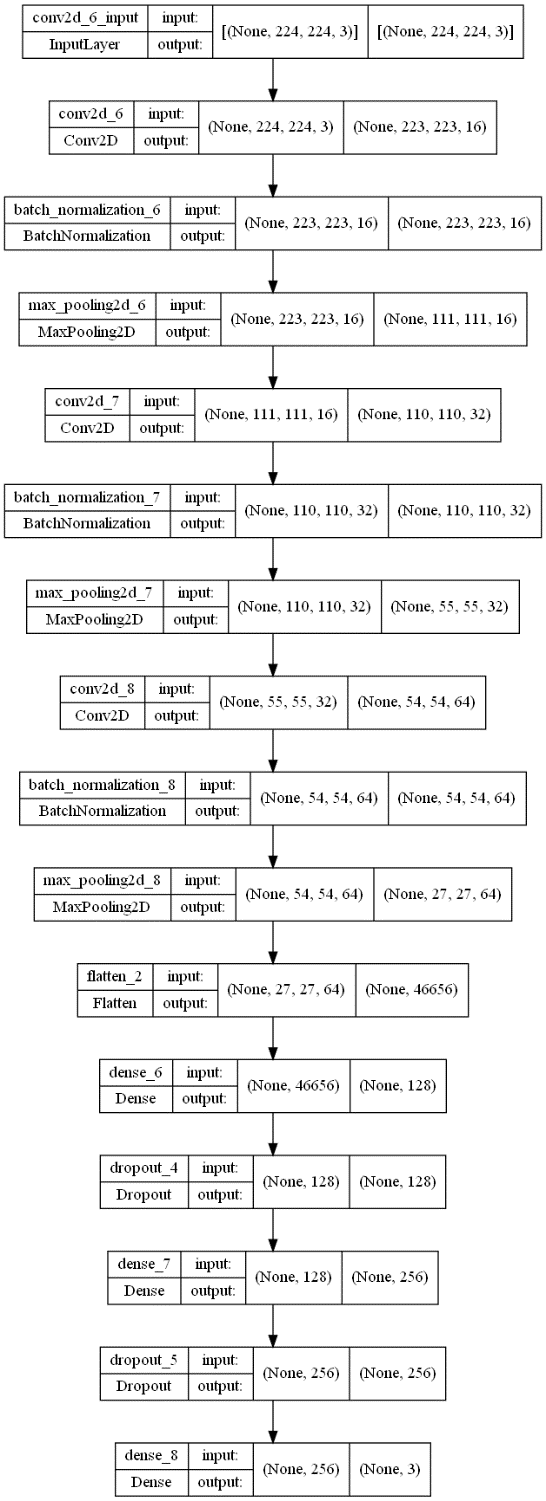
Şekil 14. Transfer Model – 2 CNN Mimarisi



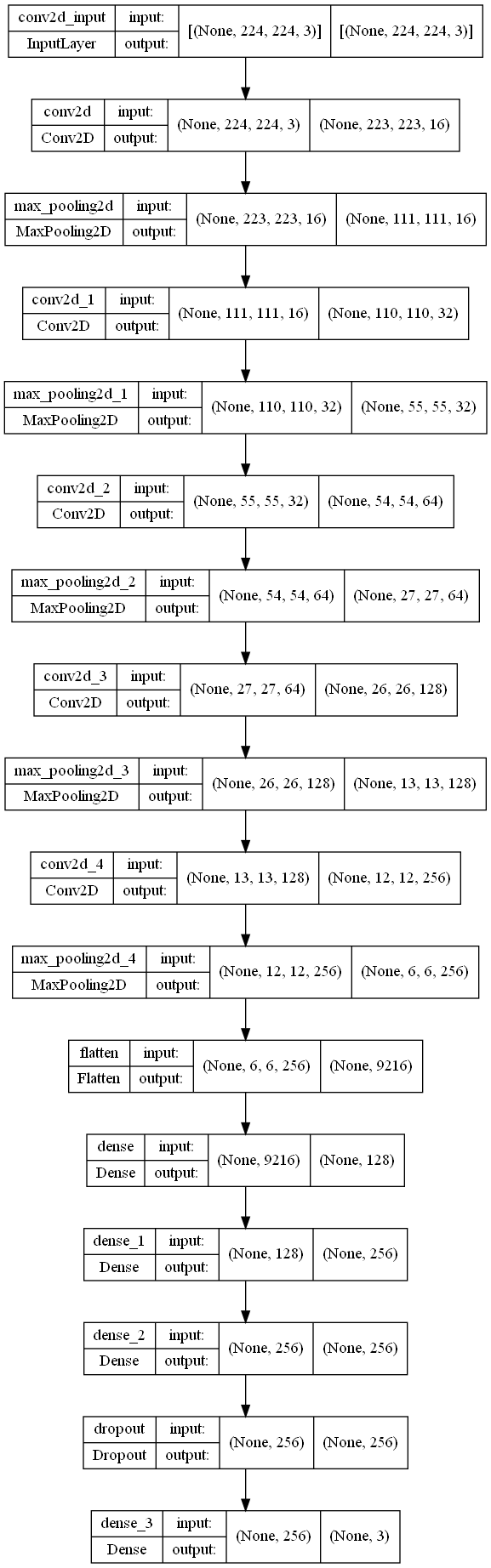
Şekil 15. Transfer Model – 3 CNN Mimarisi

**3.2.2.2.2 3 Sınıflı Model**

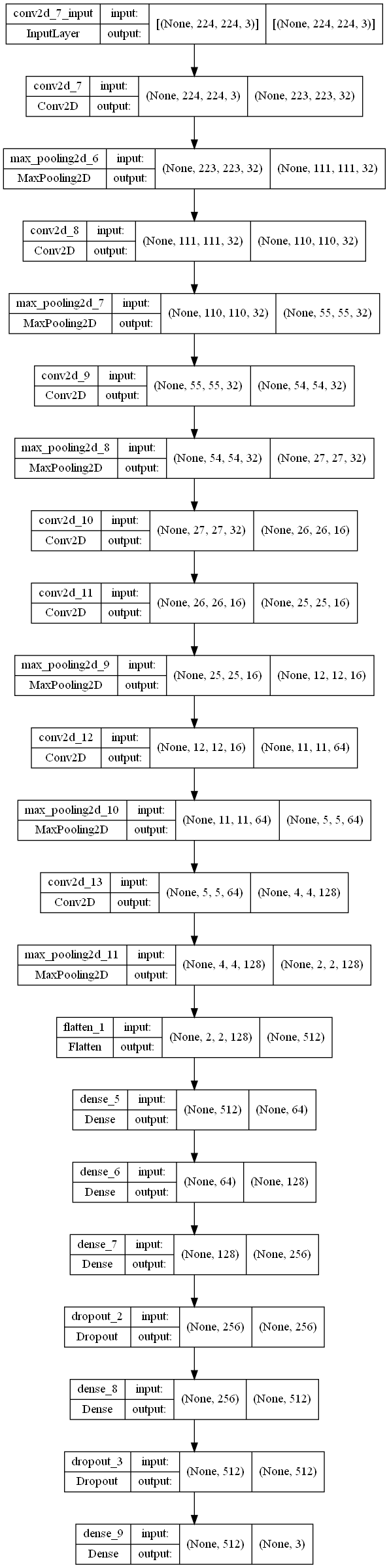
Birbirinden farklı 3 CNN ve 1 transfer öğrenme modeli oluşturuldu. Oluşturulan modeller 3.2.2.1.2’de oluşturulan veriseti ile eğitilerek validation accuracy – accuracy ve validation loss – loss grafikleri çizildi. Modellerin mimarilerinin görüntülenmesi ve incelenmesi için şablonlar oluşturuldu. Oluşturulan tüm modeller test seti ile değerlendirilip, modeller arası karşılaştırma için loss ve accuracy grafiği oluşturuldu.



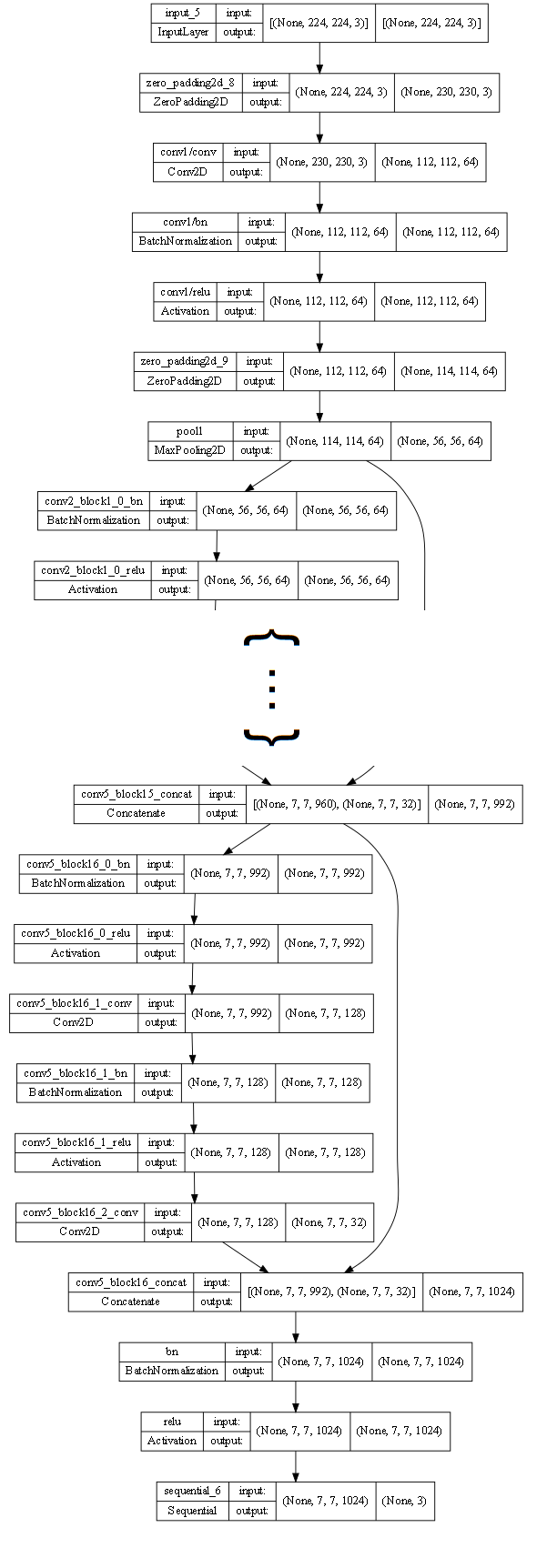
Şekil 16. Model – 1 CNN Mimarisi

****

Şekil 17. Model – 2 CNN Mimarisi

****

Şekil 18. Model – 3 CNN Mimarisi

****

Şekil 19. Transfer Model – 1 CNN Mimarisi

**3.2.2.2.3 Binary Model**

Birbirinden farklı 3 CNN ve 3 farklı transfer öğrenme modeli oluşturuldu ve 3.2.2.1.3’de oluşturulan veriseti ile eğitildi.

3 CNN modeli (özgün) içerisinden en iyi validation accuracy score’a sahip model seçildi (Model - 3). Seçilen bu model üzerinde yalnızca epoch sayısı hiperparametresi değiştirilerek 2 farklı model oluşturuldu ve 3.2.2.1.3’de oluşturulan veriseti ile eğitildi.

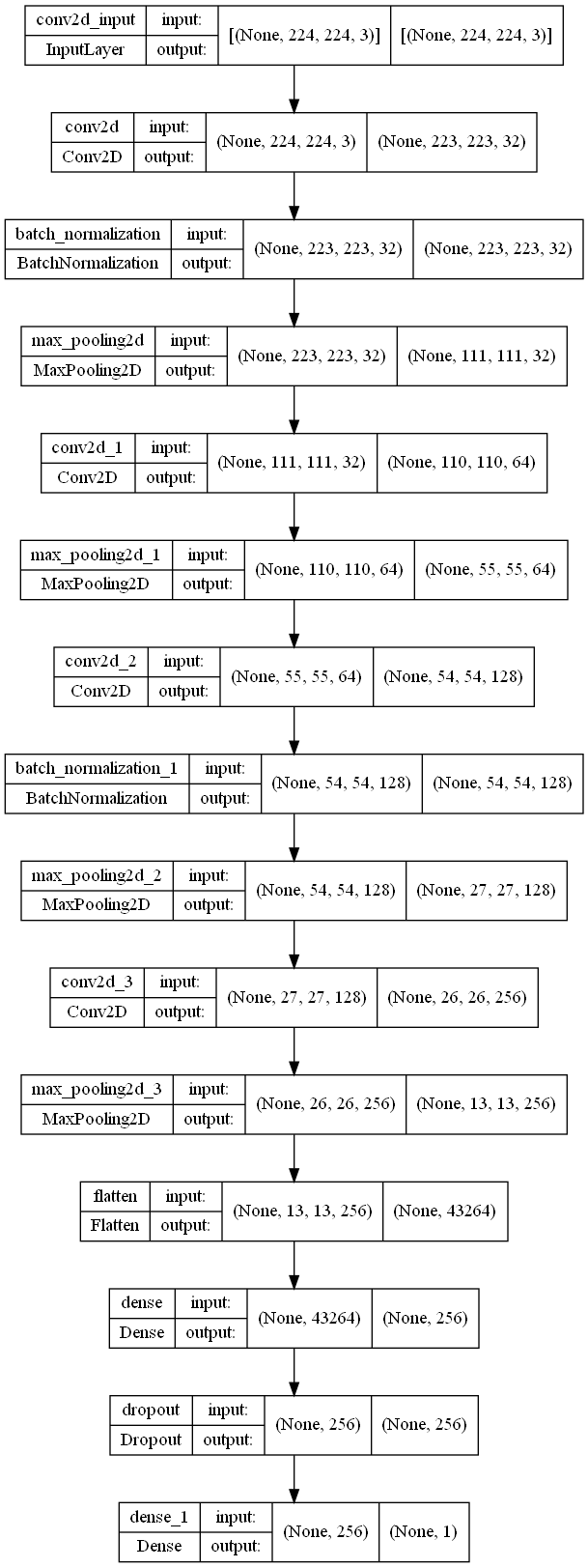
Oluşturulan 2 farklı modelden en iyi validation accuracy score’a sahip model seçildi (Model - 3.2). Seçilen bu model üzerinde yalnızca batch size hiperparametresi değiştirilerek 4 farklı model oluşturuldu ve 3.2.2.1.3’de oluşturulan veriseti ile eğitildi.

Oluşturulan 4 farklı modelden en iyi validation accuracy score’a sahip model seçildi (Model - 3.2.2). Seçilen bu model üzerinde yalnızca image size hiperparametresi değiştirilerek 4 farklı model oluşturuldu ve 3.2.2.1.3’de oluşturulan veriseti ile eğitildi.

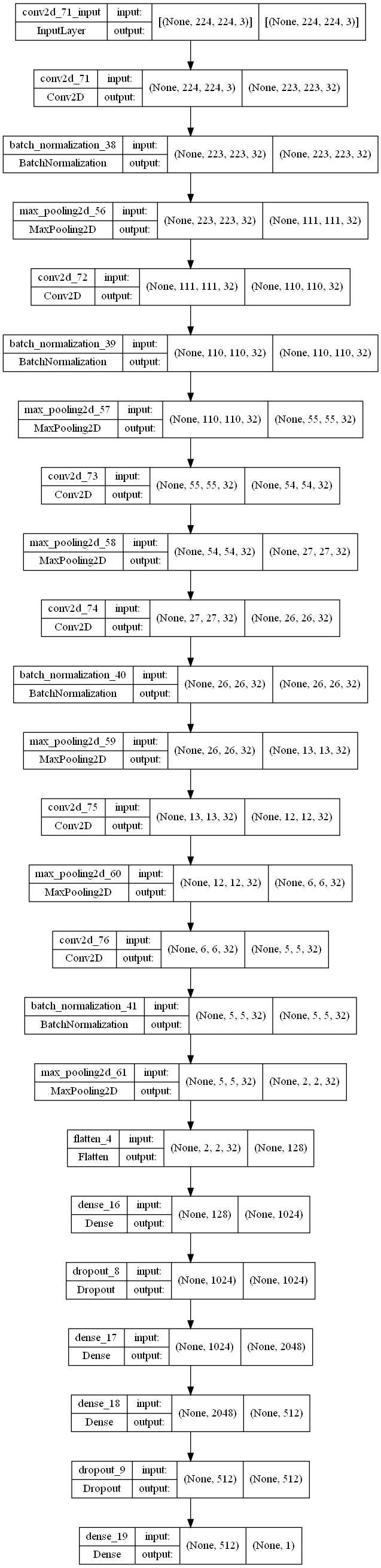
Oluşturulan 4 farklı modelden en iyi validation accuracy score’a sahip model (Model - 3.2.2.2), bir önceki seçilen modelden (Model - 3.2.2) daha yüksek score’a sahip olmadığı için sonraki adımda önceki modelden devam edildi (Model – 3.2.2).

Devam edilen bu model (Model - 3.2.2) üzerinde batch normalization ve dropout metodları çıkarılarak yeni iki model oluşturuldu ve 3.2.2.1.3’de oluşturulan veriseti ile eğitildi.

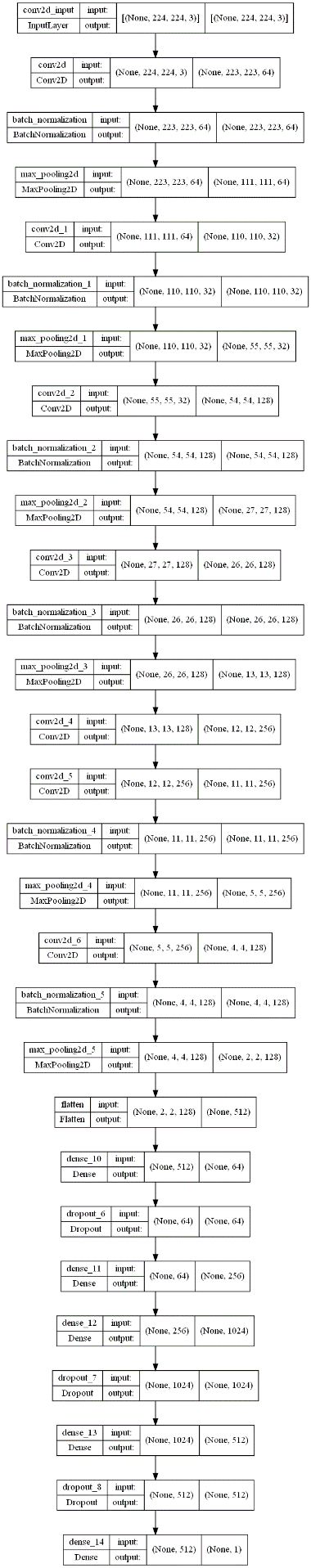
Oluşturulan tüm modellerin validation accuracy – accuracy ve validation loss – loss grafikleri çizildi. Modellerin mimarilerinin görüntülenmesi ve incelenmesi için şablonlar oluşturuldu. Oluşturulan tüm modeller test seti ile değerlendirilip, modeller arası karşılaştırma için loss ve accuracy grafiği oluşturuldu.



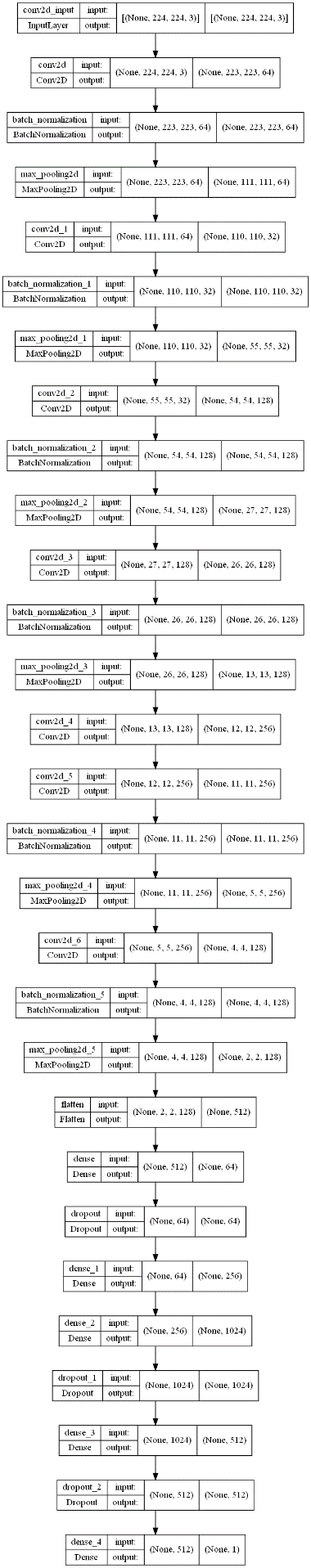
Şekil 20. Model – 1 CNN Mimarisi



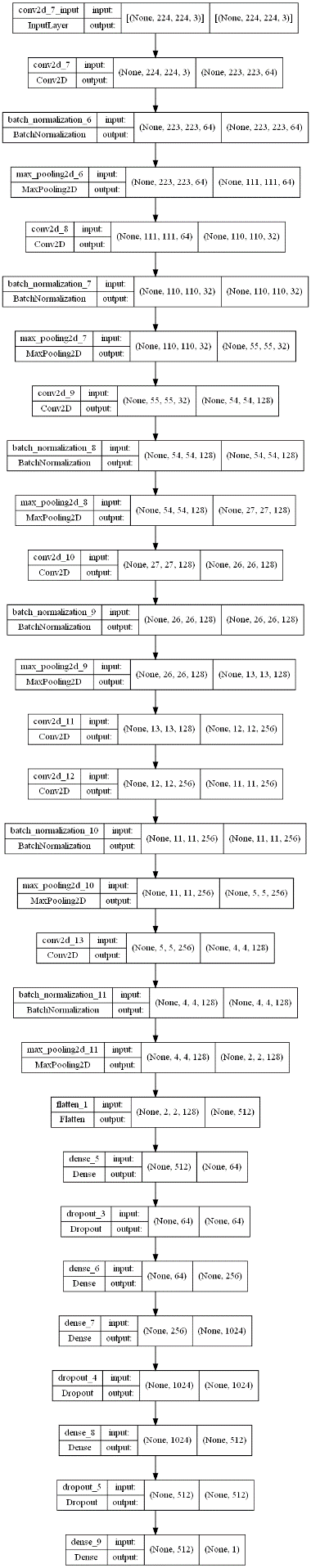
Şekil 21. Model – 2 CNN Mimarisi



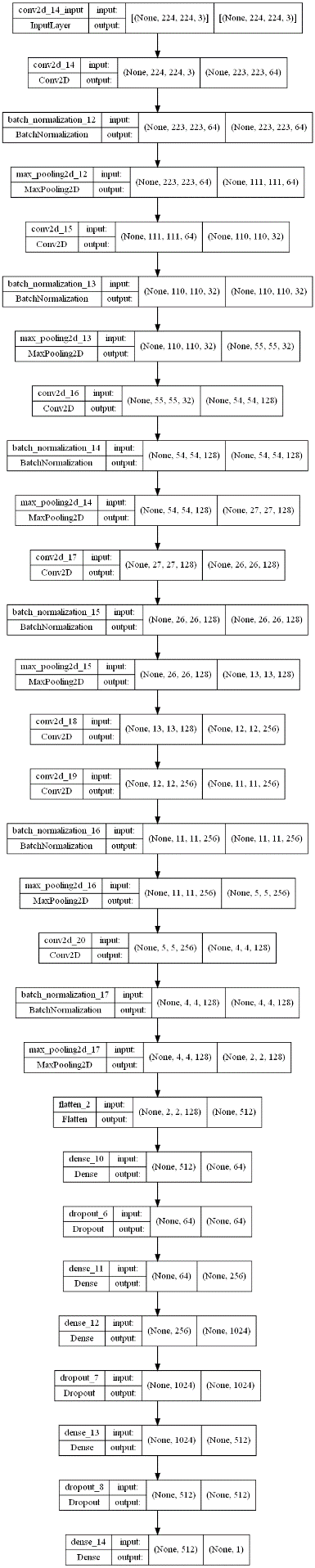
Şekil 22. Model – 3 CNN Mimarisi



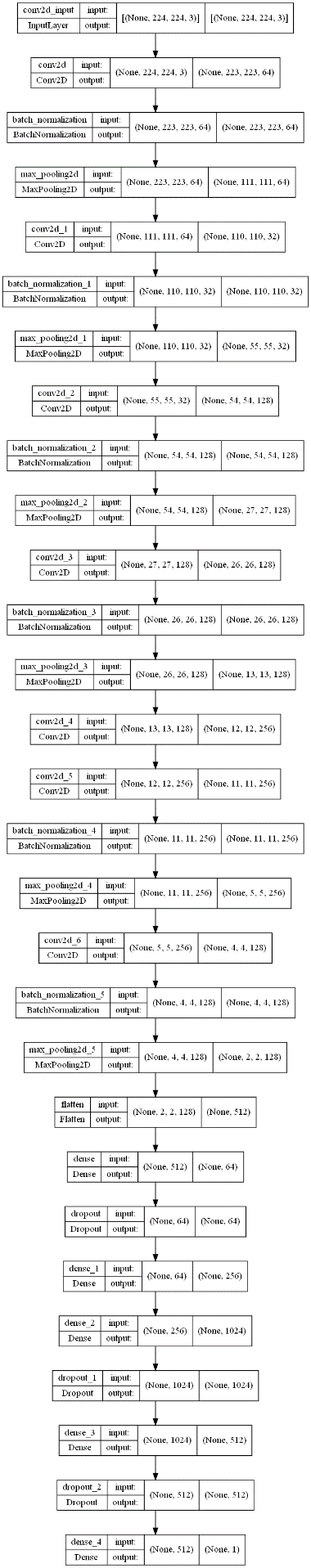
Şekil 23. Model – 3.1 CNN Mimarisi



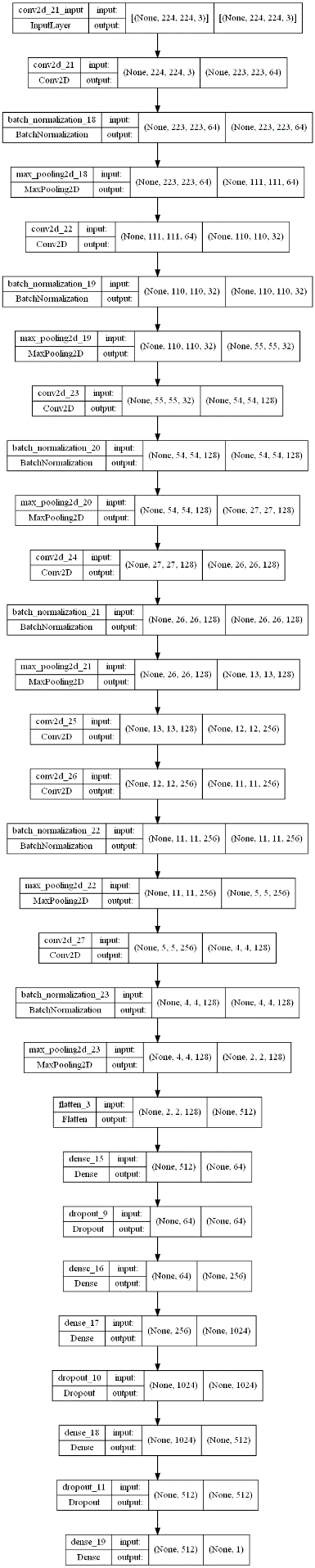
Şekil 24. Model – 3.2 CNN Mimarisi



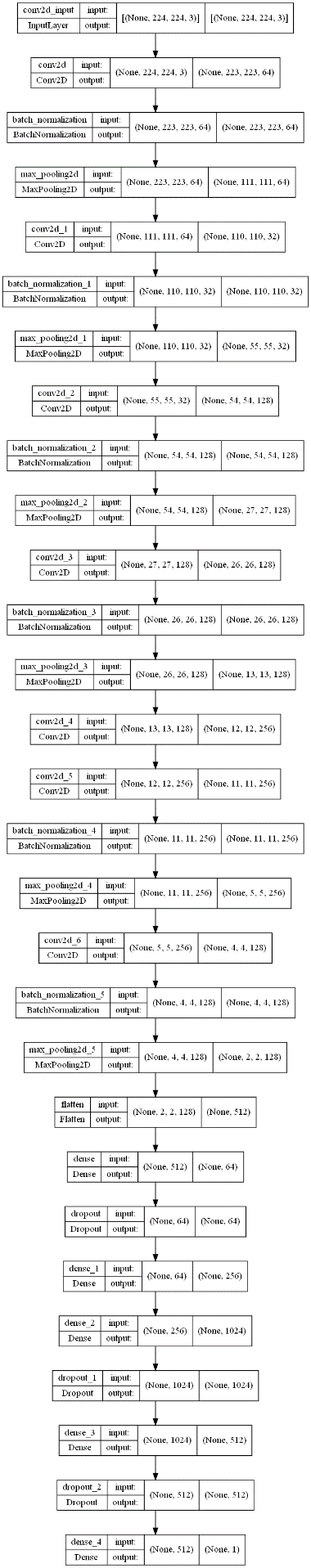
Şekil 25. Model – 3.2.1 CNN Mimarisi



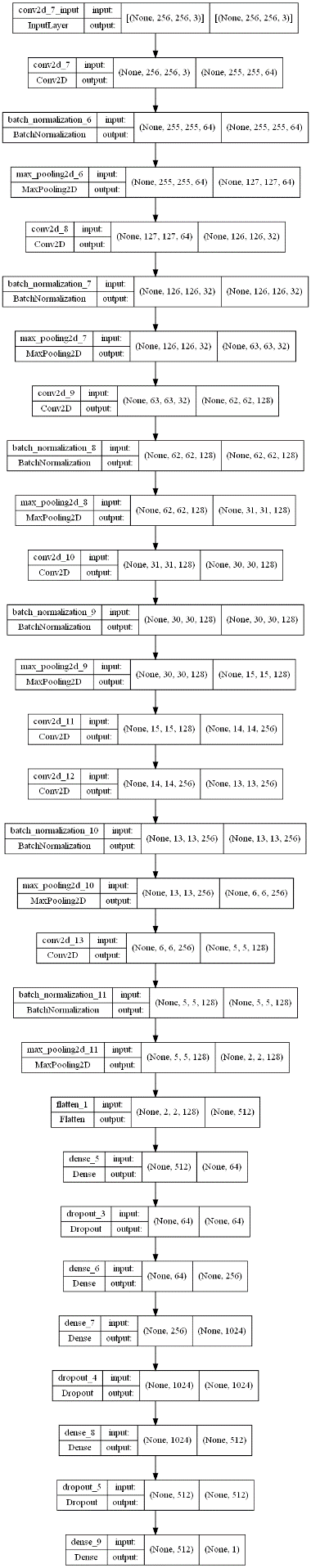
Şekil 26. Model – 3.2.2 CNN Mimarisi



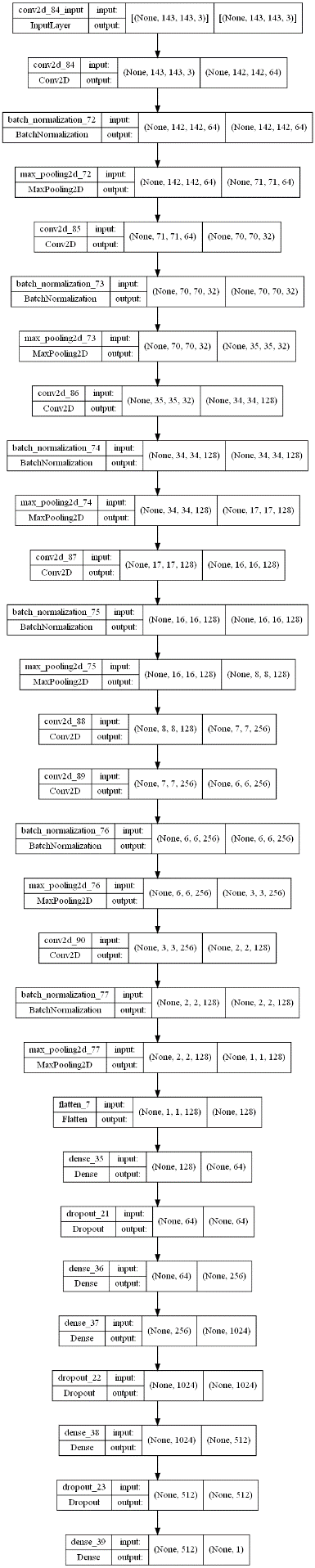
Şekil 27. Model – 3.2.3 CNN Mimarisi



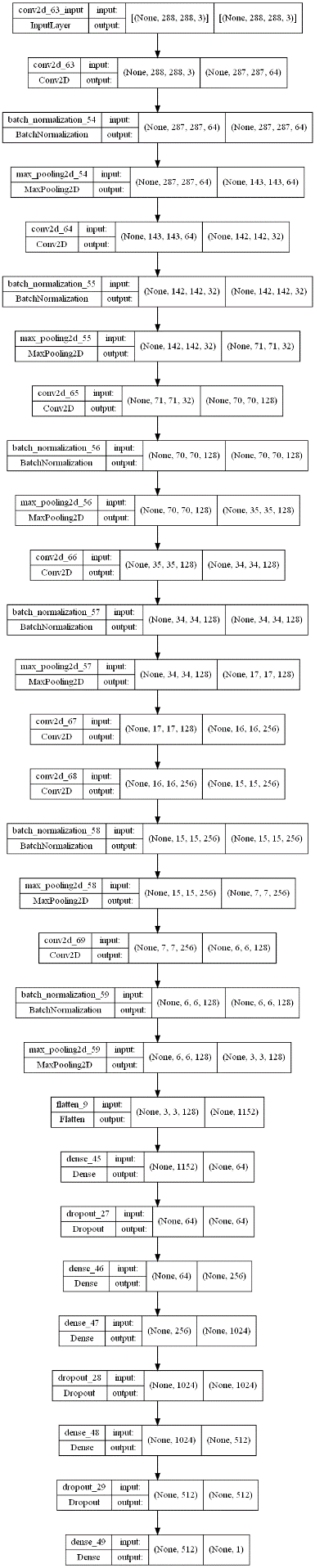
Şekil 28. Model – 3.2.4 CNN Mimarisi



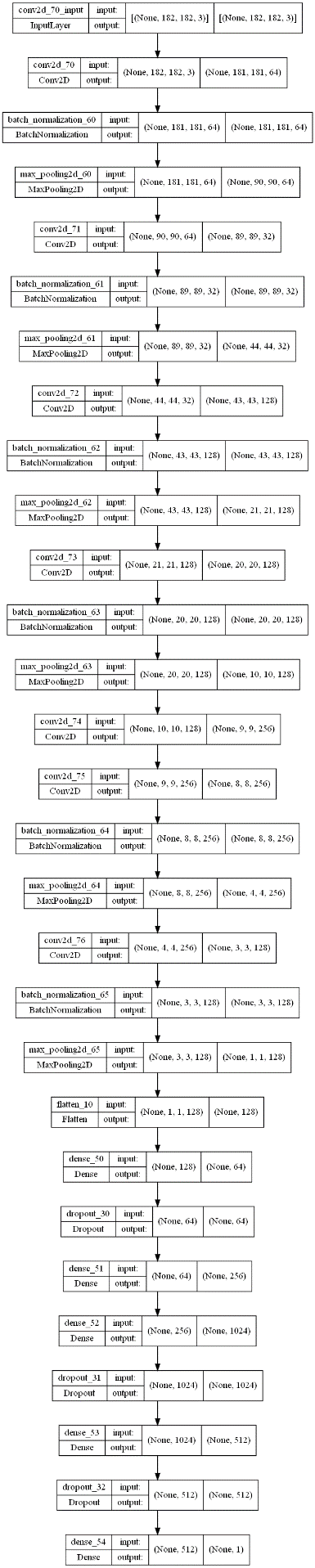
Şekil 29. Model – 3.2.2.1 CNN Mimarisi



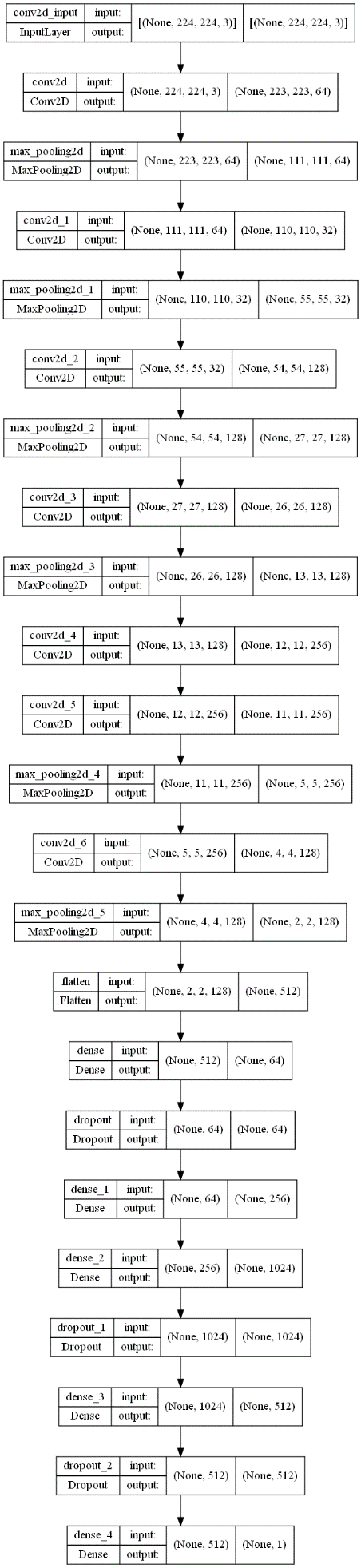
Şekil 30. Model – 3.2.2.2 CNN Mimarisi



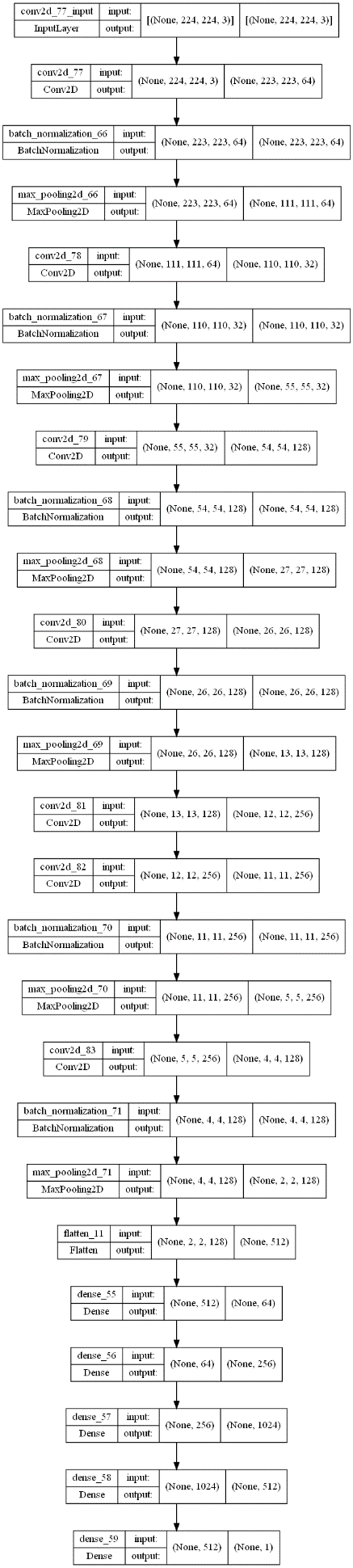
Şekil 31. Model – 3.2.2.3 CNN Mimarisi



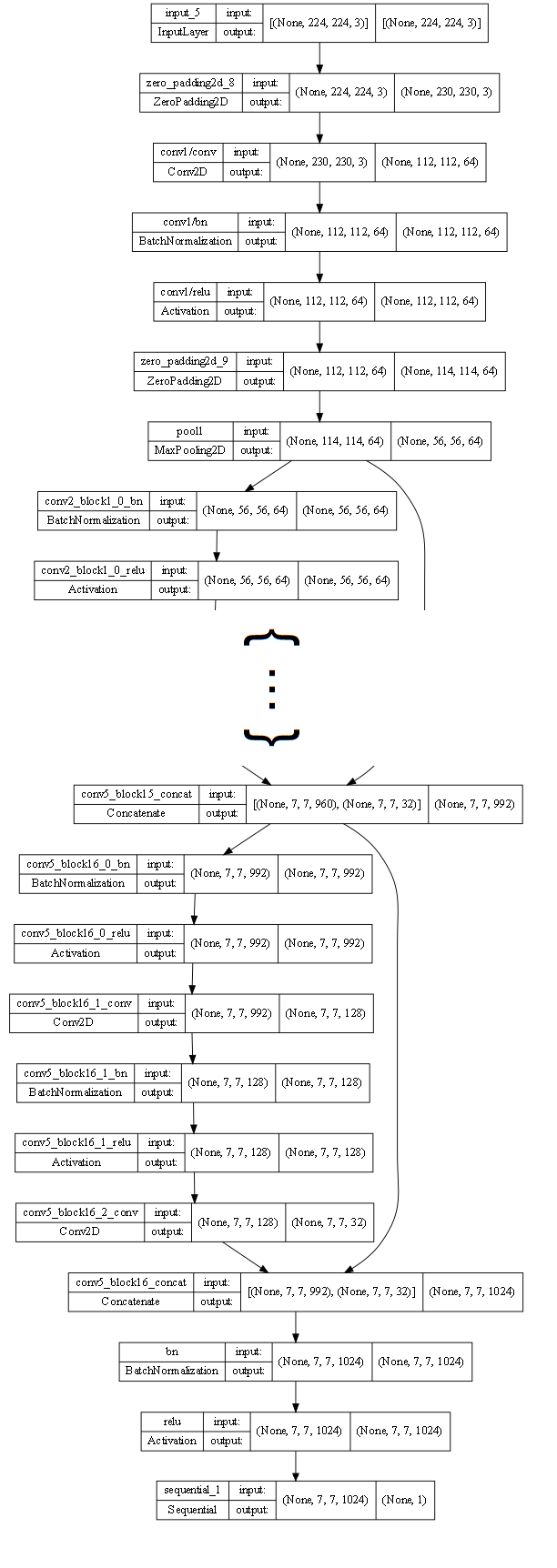
Şekil 32. Model – 3.2.2.4 CNN Mimarisi



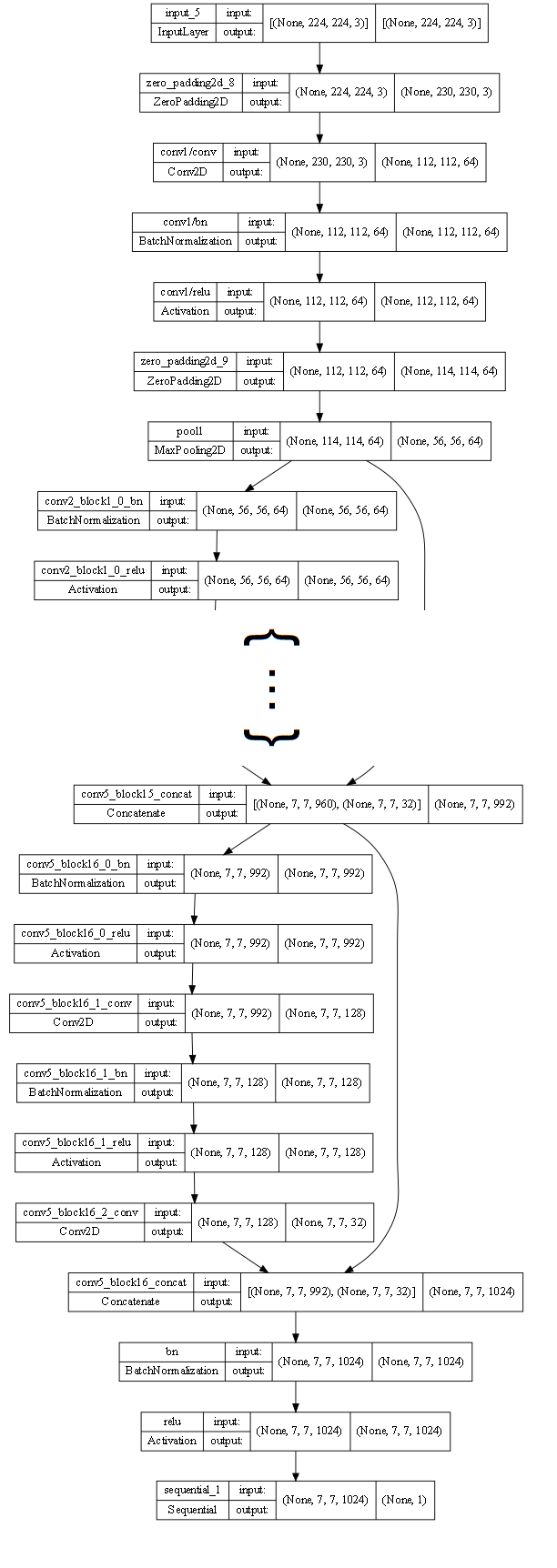
Şekil 33. Model – 3.2.2.0.1 CNN Mimarisi



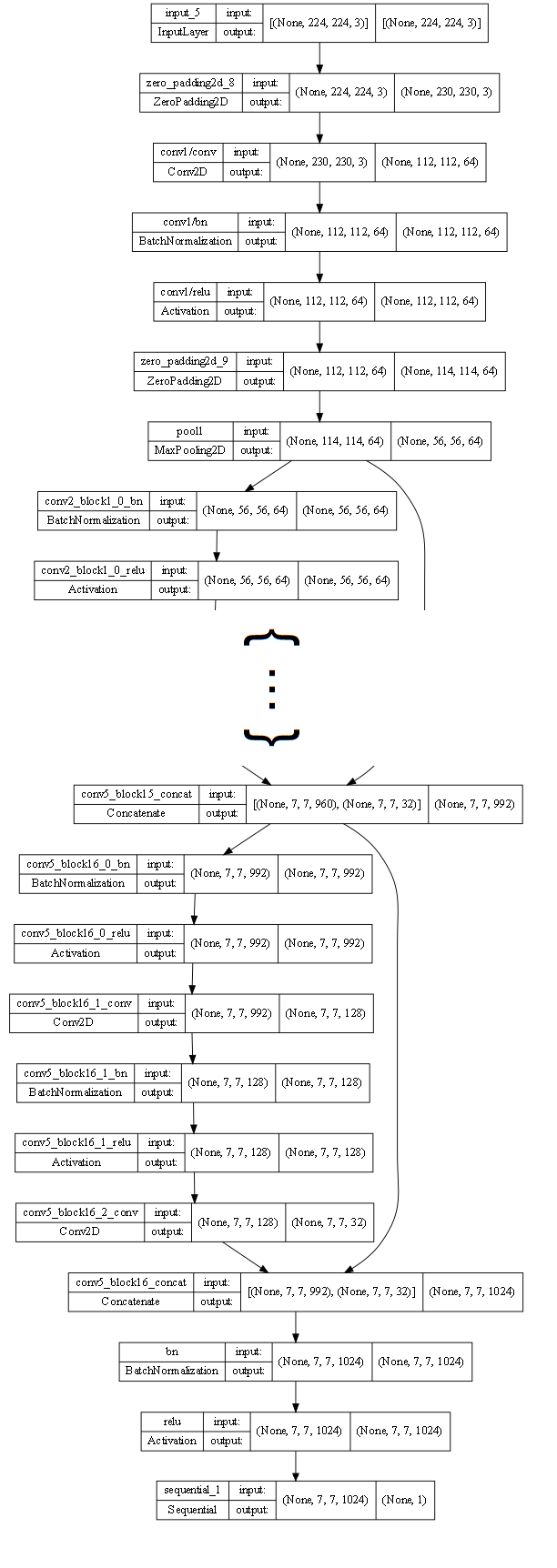
Şekil 34. Model – 3.2.2.0.2 CNN Mimarisi



Şekil 35. Transfer Model – 1 CNN Mimarisi



Şekil 36. Transfer Model – 2 CNN Mimarisi

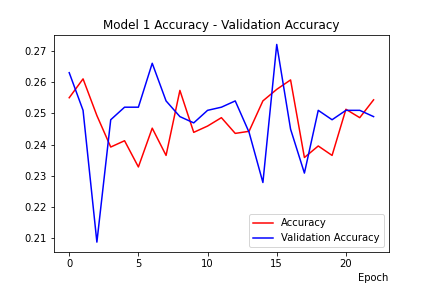


Şekil 37. Transfer Model – 3 CNN Mimarisi

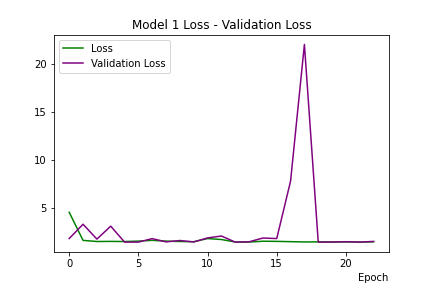
**4. BULGULAR**

**4.1 4 Sınıflı Model**

**4.1.1 Model – 1**

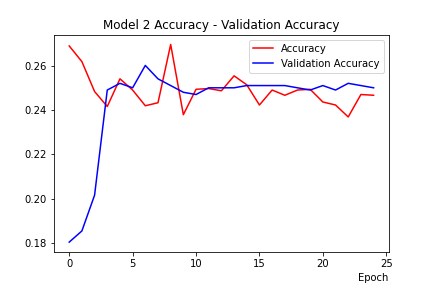
****

Şekil 38. Model – 1 Accuracy – Validation Accuracy Grafik Tablosu

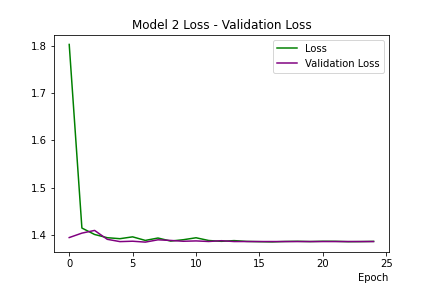


Şekil 39. Model – 1 Loss – Validation Loss Grafik Tablosu

**4.1.2 Model – 2**

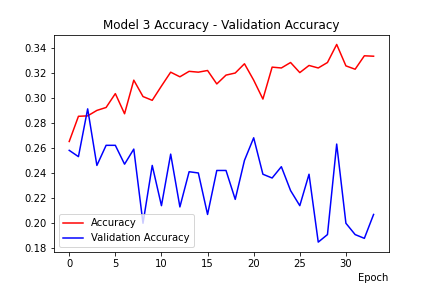


Şekil 40. Model – 2 Accuracy – Validation Accuracy Grafik Tablosu

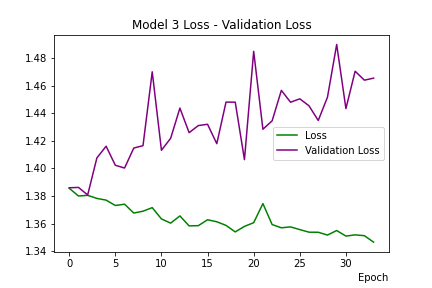


Şekil 41. Model – 2 Loss – Validation Loss Grafik Tablosu

**4.1.3 Model – 3**

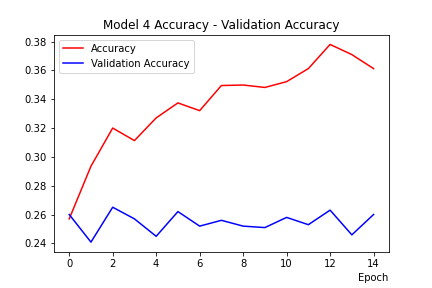


Şekil 42. Model – 3 Accuracy – Validation Accuracy Grafik Tablosu

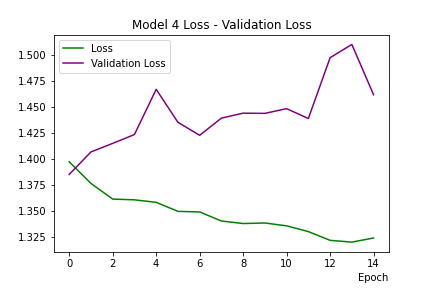


Şekil 43. Model – 3 Loss – Validation Loss Grafik Tablosu

**4.1.4 Model – 4**

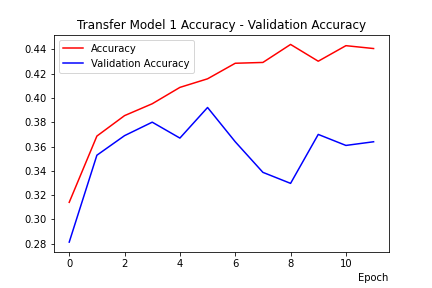


Şekil 44. Model – 4 Accuracy – Validation Accuracy Grafik Tablosu

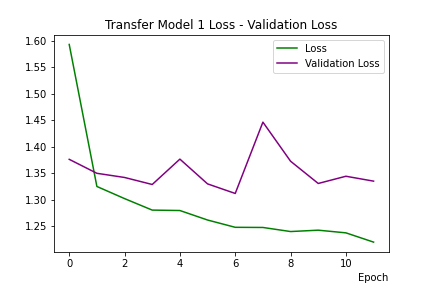


Şekil 45. Model – 3 Loss – Validation Loss Grafik Tablosu

**4.1.5 Transfer Model - 1**

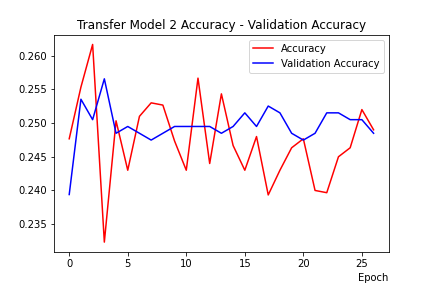


Şekil 46. Transfer Model – 1 Accuracy – Validation Accuracy Grafik Tablosu

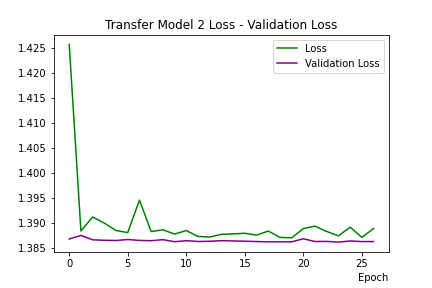


Şekil 47. Transfer Model – 1 Loss – Validation Loss Grafik Tablosu

**4.1.6 Transfer Model - 2**

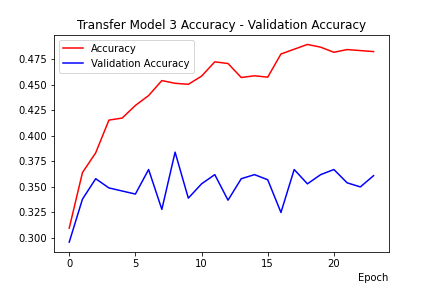


Şekil 48. Transfer Model – 2 Accuracy – Validation Accuracy Grafik Tablosu

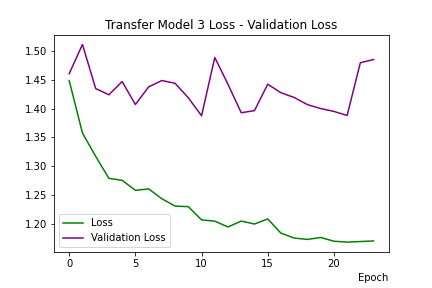


Şekil 49. Transfer Model – 2 Loss – Validation Loss Grafik Tablosu

**4.1.7 Transfer Model – 3**



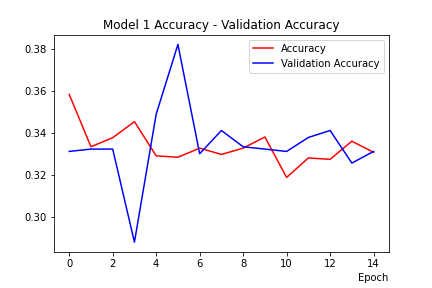
Şekil 50. Transfer Model – 3 Accuracy – Validation Accuracy Grafik Tablosu



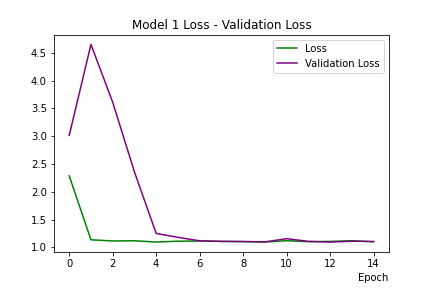
Şekil 51. Transfer Model – 3 Loss – Validation Loss Grafik Tablosu

**4.2 3 Sınıflı Model**

**4.2.1 Model – 1**

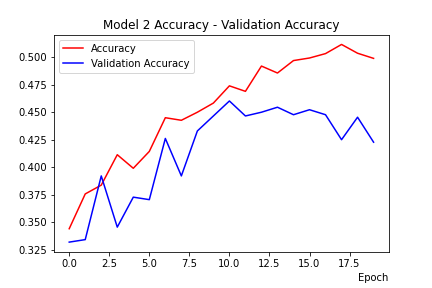


Şekil 52. Model – 1 Accuracy – Validation Accuracy Grafik Tablosu

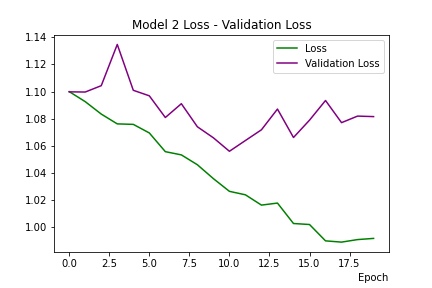


Şekil 53. Model – 1 Loss – Validation Loss Grafik Tablosu

**4.2.2 Model – 2**

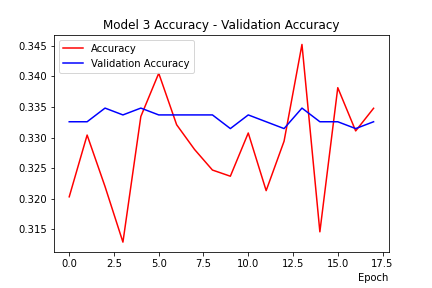


Şekil 54. Model – 2 Accuracy – Validation Accuracy Grafik Tablosu,

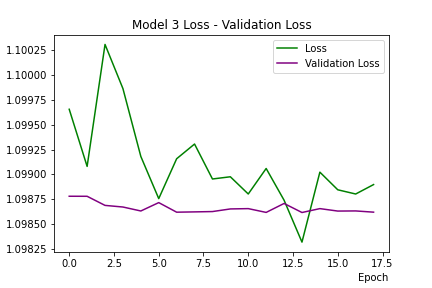


Şekil 55. Model – 2 Loss – Validation Loss Grafik Tablosu

**4.2.3 Model – 3**

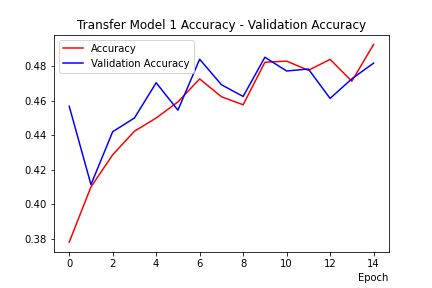


Şekil 56. Model – 3 Accuracy – Validation Accuracy Grafik Tablosu

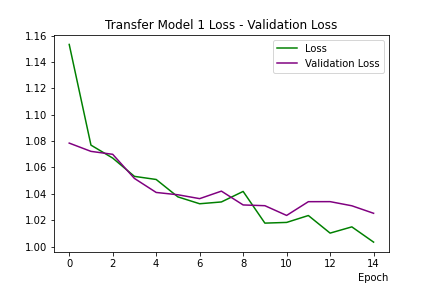


Şekil 57. Model – 3 Loss – Validation Loss Grafik Tablosu

**4.2.4 Transfer Model – 1**



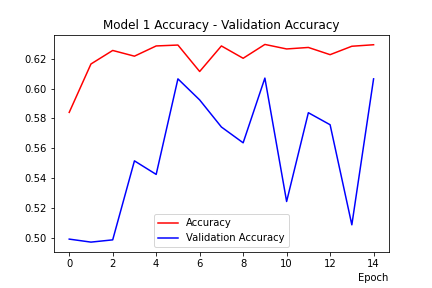
Şekil 58. Transfer Model – 1 Accuracy – Validation Accuracy Grafik Tablosu



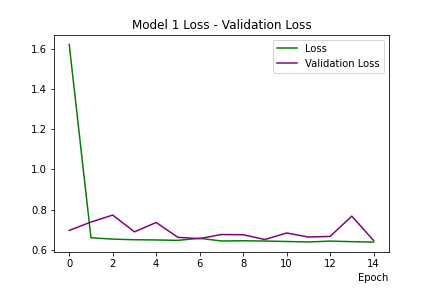
Şekil 59. Transfer Model – 1 Loss – Validation Loss Grafik Tablosu

**4.3 Binary Model**

**4.3.1 Model –1**

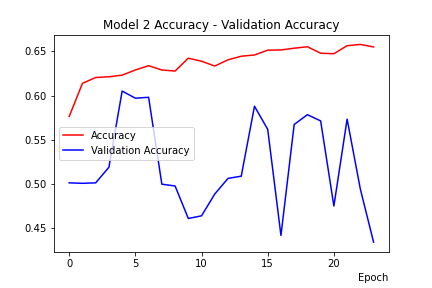


Şekil 60. Model – 1 Accuracy – Validation Accuracy Grafik Tablosu



Şekil 61. Model – 1 Loss – Validation Loss Grafik Tablosu

**4.3.2 Model – 2**

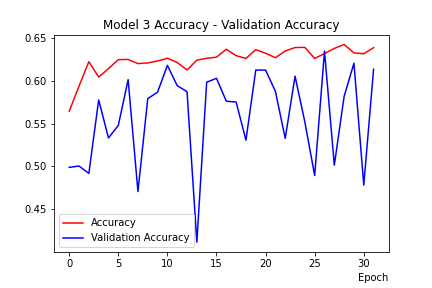


Şekil 62. Model – 2 Accuracy – Validation Accuracy Grafik Tablosu

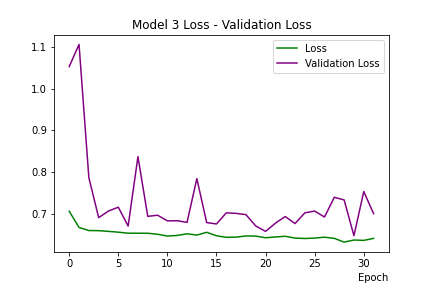


Şekil 63. Model – 2 Loss – Validation Loss Grafik Tablosu

**4.3.3 Model- 3**

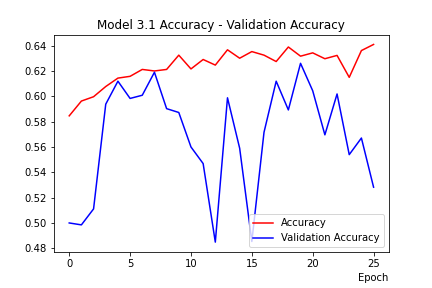


Şekil 64. Model – 3 Accuracy – Validation Accuracy Grafik Tablosu

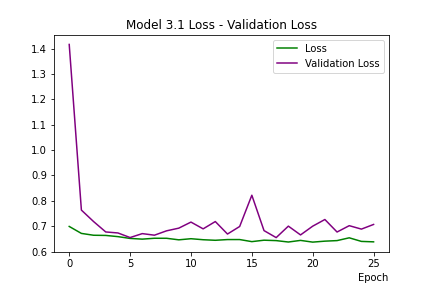


Şekil 65. Model – 3 Loss – Validation Loss Grafik Tablosu

**4.3.4 Model – 3.1**

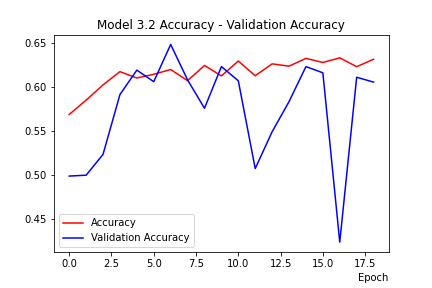


Şekil 66. Model – 3.1 Accuracy – Validation Accuracy Grafik Tablosu

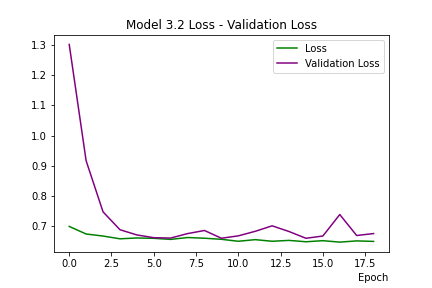


Şekil 67. Model – 3.1 Loss – Validation Loss Grafik Tablosu

**4.3.5 Model – 3.2**

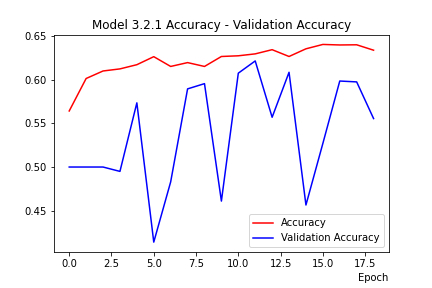


Şekil 68. Model – 3.2 Accuracy – Validation Accuracy Grafik Tablosu

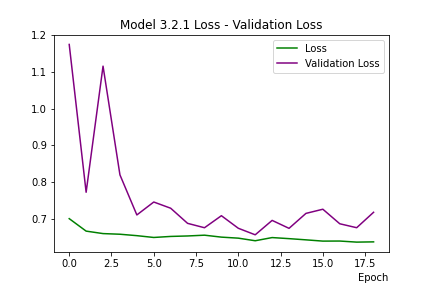


Şekil 69. Model – 3.2 Loss – Validation Loss Grafik Tablosu

**4.3.6 Model – 3.2.1**

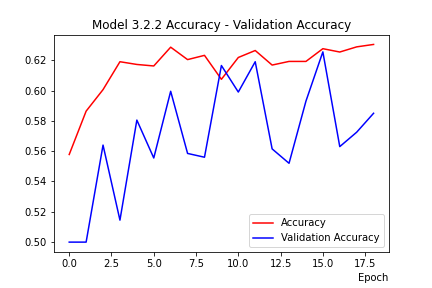


Şekil 70. Model – 3.2.1 Accuracy – Validation Accuracy Grafik Tablosu

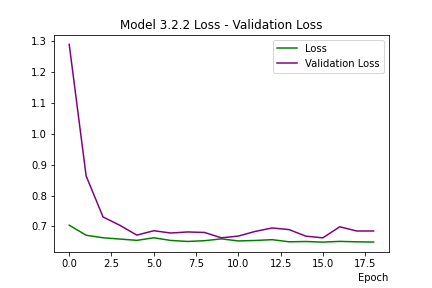


Şekil 71. Model – 3.2.1 Loss – Validation Loss Grafik Tablosu

**4.3.7 Model – 3.2.2**

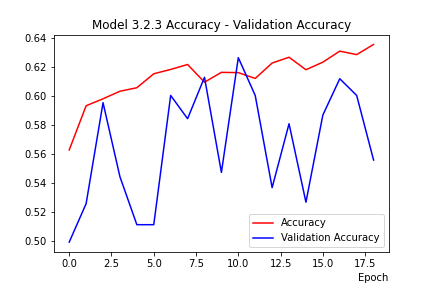


Şekil 72. Model – 3.2.2 Accuracy – Validation Accuracy Grafik Tablosu

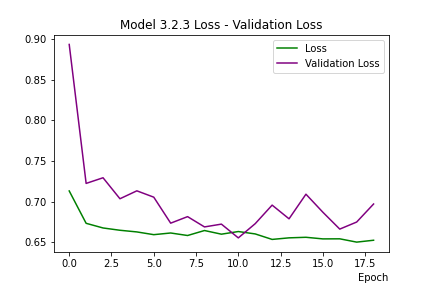


Şekil 73. Model – 3.2.2 Loss – Validation Loss Grafik Tablosu

**4.3.8 Model – 3.2.3**

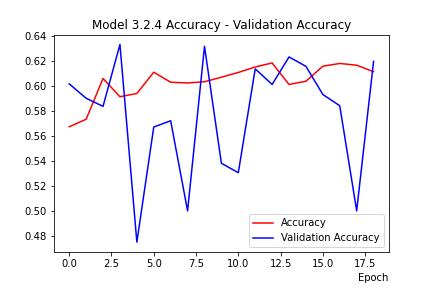


Şekil 74. Model – 3.2.3 Accuracy – Validation Accuracy Grafik Tablosu

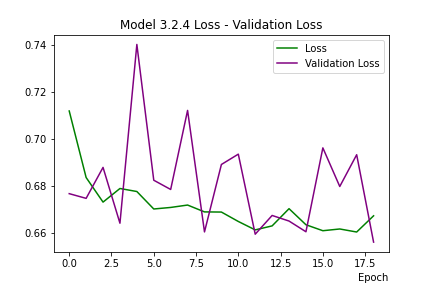


Şekil 75. Model – 3.2.3 Loss – Validation Loss Grafik Tablosu

**4.3.9 Model – 3.2.4**

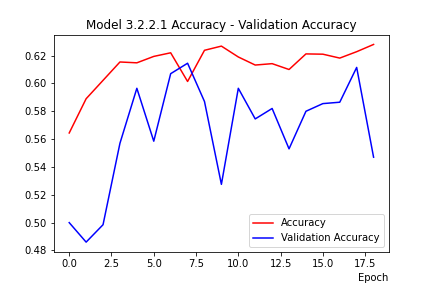


Şekil 76. Model – 3.2.4 Accuracy – Validation Accuracy Grafik Tablosu

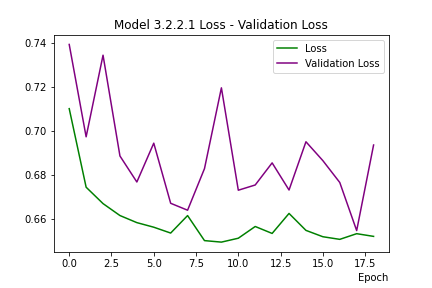


Şekil 77. Model – 3.2.4 Loss – Validation Loss Grafik Tablosu

**4.3.10 Model – 3.2.2.1**

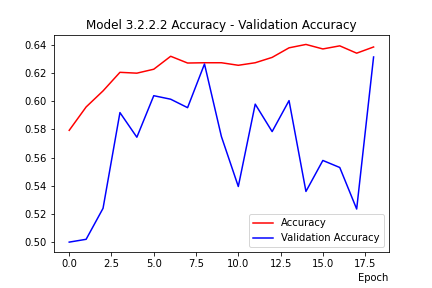


Şekil 78. Model – 3.2.2.1 Accuracy – Validation Accuracy Grafik Tablosu

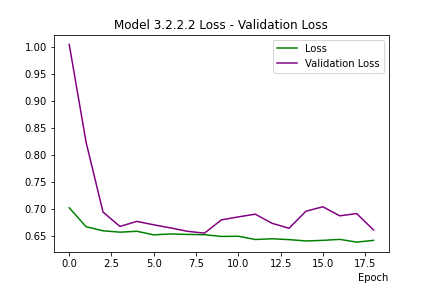


Şekil 79. Model – 3.2.2.1 Loss – Validation Loss Grafik Tablosu

**4.3.11 Model – 3.2.2.2**

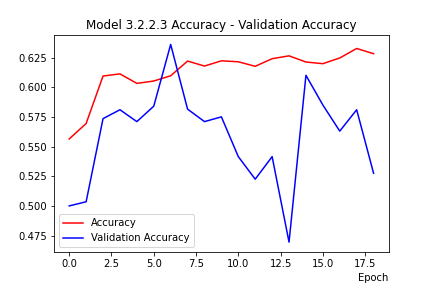


Şekil 80. Model – 3.2.2.2 Accuracy – Validation Accuracy Grafik Tablosu

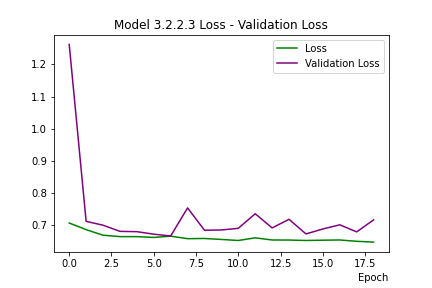


Şekil 81. Model – 3.2.2.2 Loss – Validation Loss Grafik Tablosu

**4.3.12 Model – 3.2.2.3**

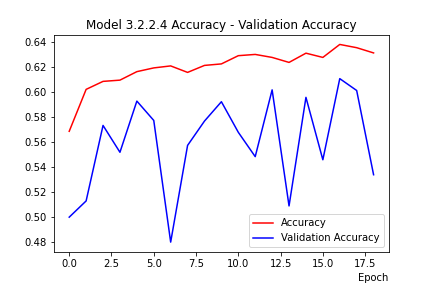


Şekil 82. Model – 3.2.2.3 Accuracy – Validation Accuracy Grafik Tablosu

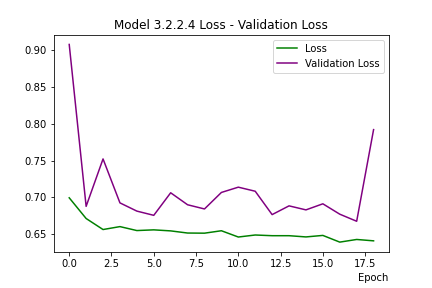


Şekil 83. Model – 3.2.2.3 Loss – Validation Loss Grafik Tablosu

**4.3.13 Model – 3.2.2.4**

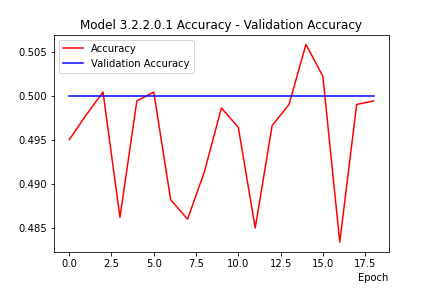


Şekil 84. Model – 3.2.2.4 Accuracy – Validation Accuracy Grafik Tablosu

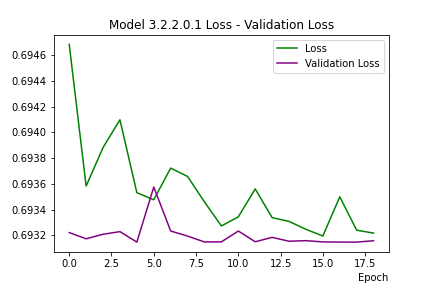


Şekil 85.Model – 3.2.2.4 Loss – Validation Loss Grafik Tablosu

**4.3.14 Model – 3.2.2.0.1**

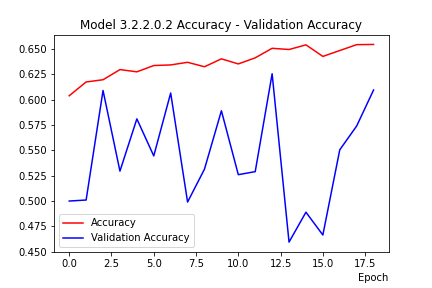


Şekil 86. Model – 3.2.2.0.1 Accuracy – Validation Accuracy Grafik Tablosu

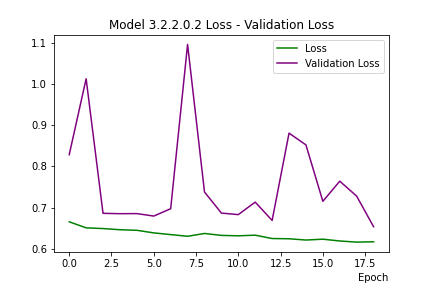


Şekil 87. Model – 3.2.2.0.1 Loss – Validation Loss Grafik Tablosu

**4.3.15 Model – 3.2.2.0.2**

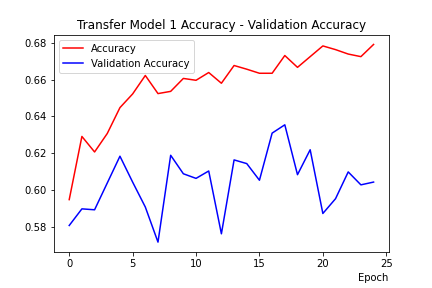


Şekil 88. Model – 3.2.2.0.2 Accuracy – Validation Accuracy Grafik Tablosu

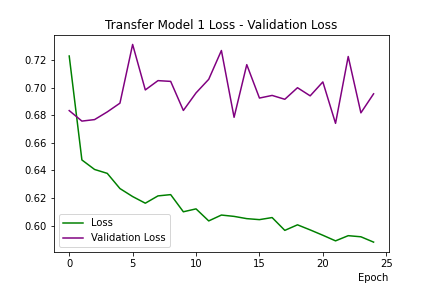


Şekil 89.Model – 3.2.2.0.2 Loss – Validation Loss Grafik Tablosu

**4.3.16 Transfer Model – 1**

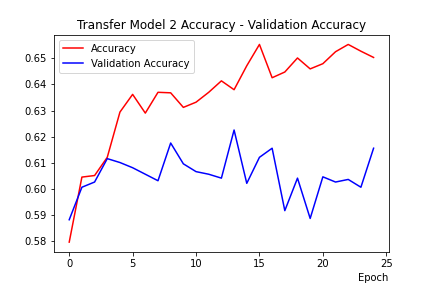


Şekil 90. Transfer Model – 1 Accuracy – Validation Accuracy Grafik Tablosu

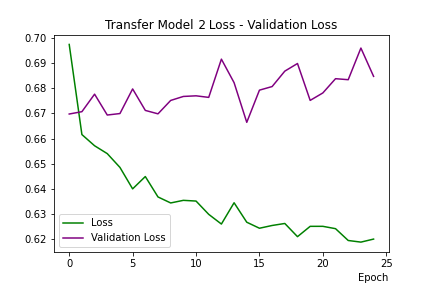


Şekil 91. Transfer Model – 1 Loss – Validation Loss Grafik Tablosu

**4.3.17 Transfer Model – 2**

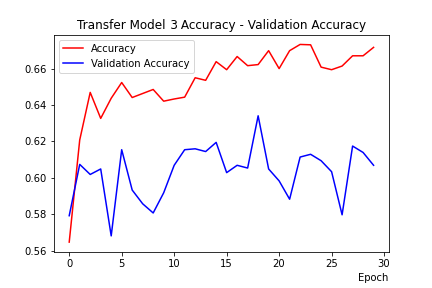


Şekil 92. Transfer Model – 2 Accuracy – Validation Accuracy Grafik Tablosu

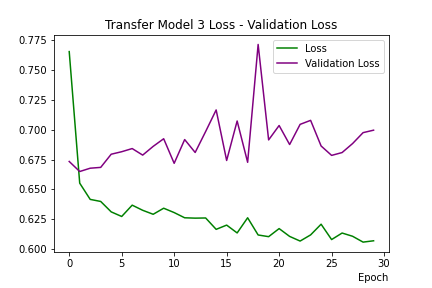


Şekil 93. Transfer Model – 2 Loss – Validation Loss Grafik Tablosu

**4.3.18 Transfer Model – 3**



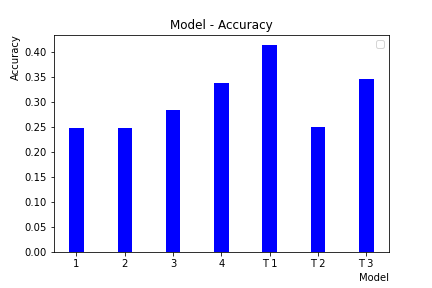
Şekil 94. Transfer Model – 3 Accuracy – Validation Accuracy Grafik Tablosu



Şekil 95. Transfer Model – 3 Loss – Validation Loss Grafik Tablosu

**5. SONUÇ**

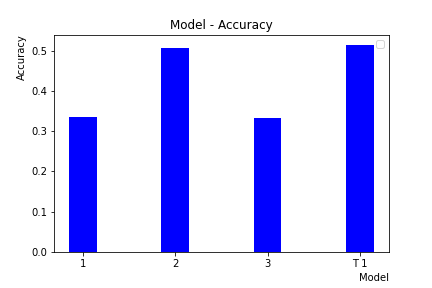
**5.1 4 Sınıflı Model**

****

Şekil 96. 4 Sınıflı Modellerin Accuracy Grafik Tablosu

Şekil 97. 4 Sınıflı Modellerin Loss Grafik Tablosu

**5.2 3 Sınıflı Model**

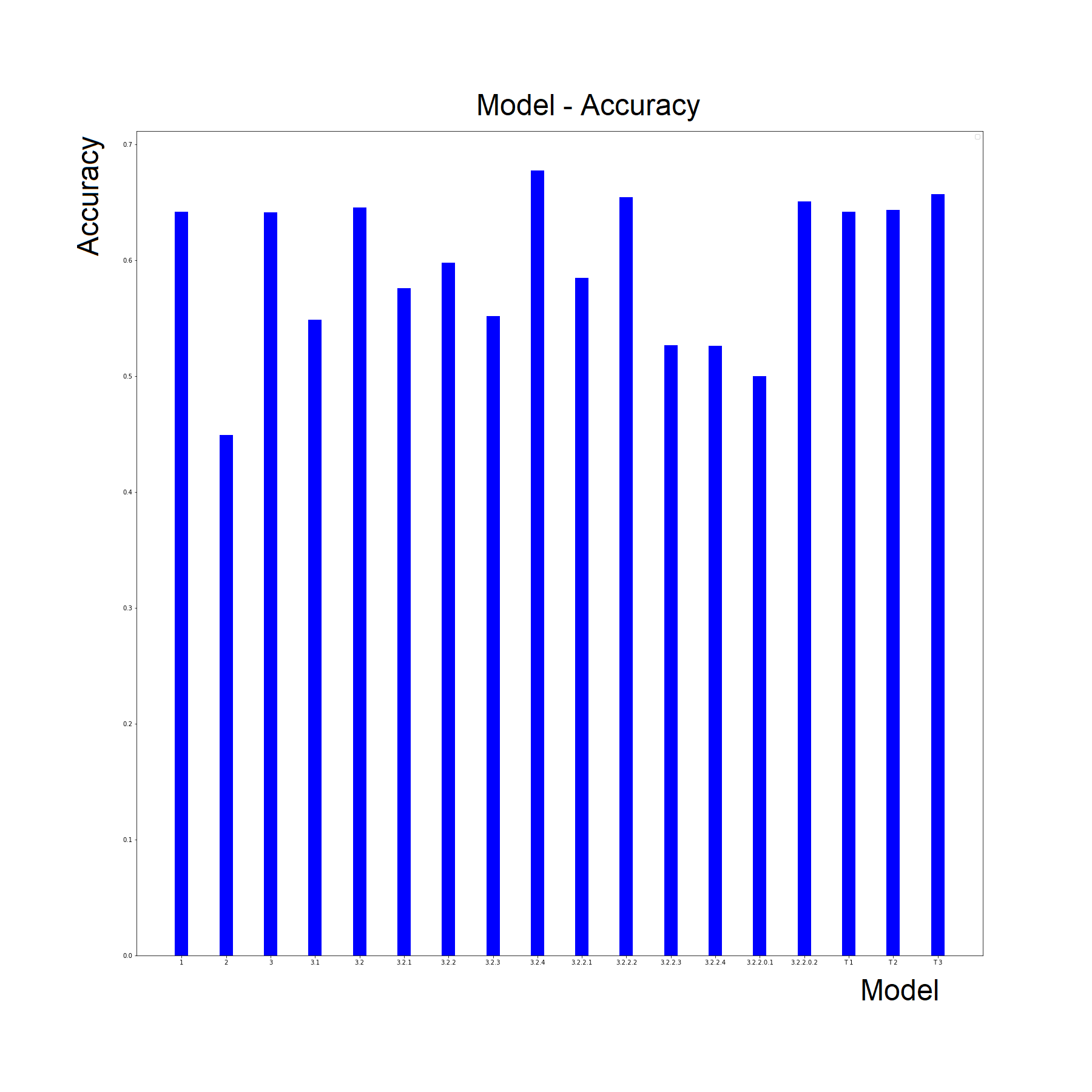


Şekil 98. 3 Sınıflı Modellerin Accuracy Grafik Tablosu

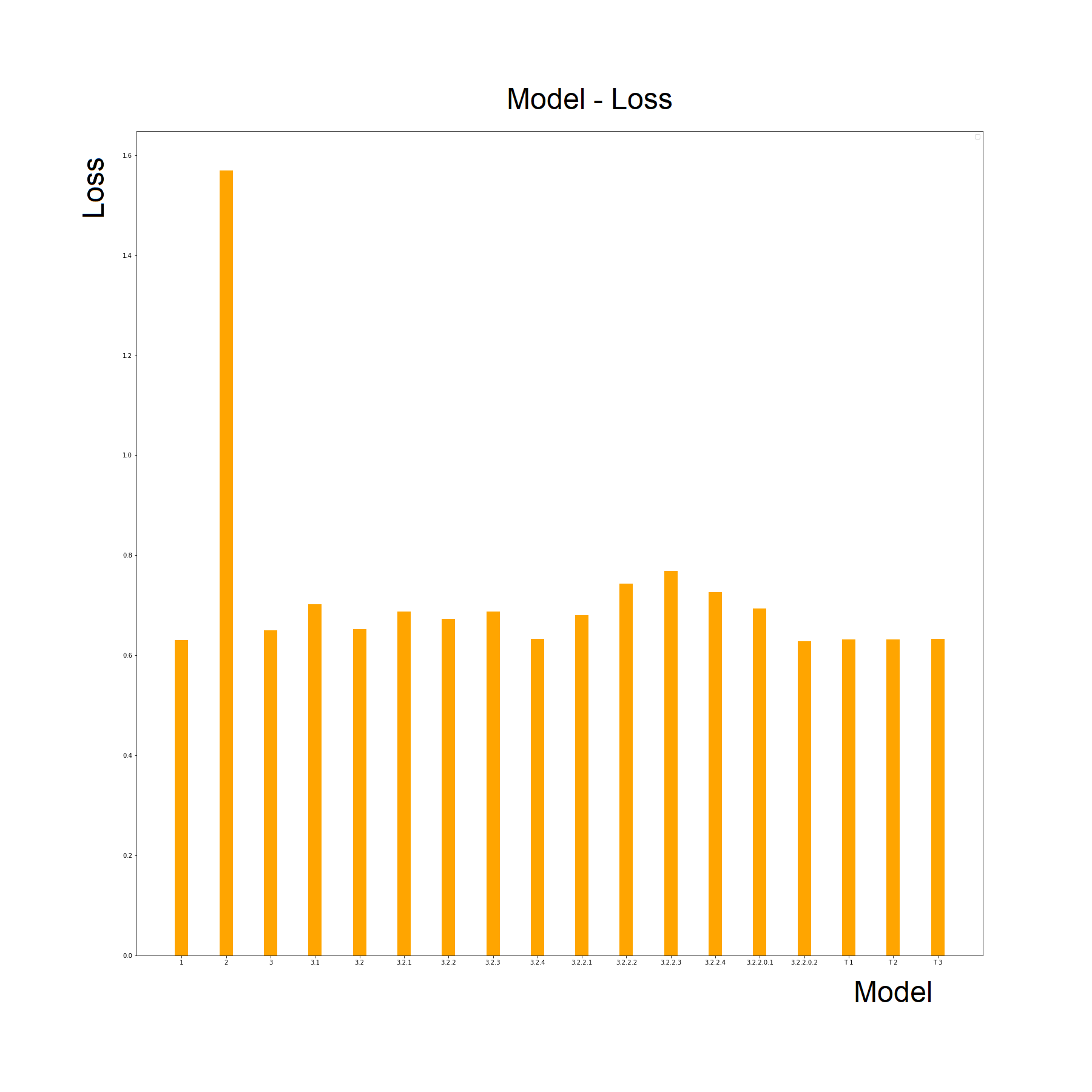


Şekil 99. 3 Sınıflı Modellerin Loss Grafik Tablosu

**5.3 Binary Model**

****

Şekil 100. Binary Modellerin Accuracy Grafik Tablosu



Şekil 101. Binary Modellerin Loss Grafik Tablosu

**5.4 Çıkarımlar**

Yapılan bu çalışmada transfer öğrenme modellerinin, özgün CNN modellerine göre çoğu zaman daha başarılı olduğu ve zaman tasarrufu sağladığı gözlemlenmiştir.

Verisetlerinde bulunan röntgen filmlerindeki hastalıklar arası farkların gözle ayırt edilemeyecek durumda olmasından kaynaklı tüm modellerde loss değerleri yüksek, accuracy değerleri ise düşük oranda elde edilmiştir.

Sonuç olarak akciğer röntgen filmleri gibi bir takım biyomedikal verilerin yalnızca görüntü bazında işlenip değerlendirilmesinde şuanki derin öğrenme teknolojileri yetersiz kalmaktadır.

**6. KAYNAKLAR**

1: <https://tr.wikipedia.org/wiki/Wilhelm_Röntgen>

2: <https://www.kaggle.com/datasets/nih-chest-xrays/data>

3: <https://www.isnet.net.tr/BlogIcerik/Phyton-Programlama-Nedir-isnetblog>

4: <https://tr.wikipedia.org/wiki/Python>

5: <https://tr.wikipedia.org/wiki/TensorFlow>

6: <https://tr.wikipedia.org/wiki/Keras>

7: <https://tr.wikipedia.org/wiki/CUDA>

8: <https://developer.nvidia.com/cudnn>

9: <https://tr.wikipedia.org/wiki/Anaconda_(Python_dağıtımı)>

10: <https://en.wikipedia.org/wiki/Spyder_(software)>

11: <https://berqnet.com/blog/yapay-zeka>

12: <https://tr.wikipedia.org/wiki/Makine_öğrenimi>

13: <https://www.elektrikport.com/makale-detay/denetimli-ve-denetimsiz-makine-ogrenmesi-nedir/22487#ad-image-0>

14: <https://veribilimcisi.com/2017/07/12/denetimli-ogrenme/>

15: <https://tr.wikipedia.org/wiki/Yapay_sinir_ağları>

16: [https://tr.wikipedia.org/wiki/Derin\_öğrenme](https://tr.wikipedia.org/wiki/Derin_%C3%B6%C4%9Frenme)

17: <https://www.beyaz.net/tr/yazilim/makaleler/derin_ogrenme_deep_learning_nedir.html>

18: [https://tr.wikipedia.org/wiki/Evrişimli\_sinir\_ağları](https://tr.wikipedia.org/wiki/Evri%C5%9Fimli_sinir_a%C4%9Flar%C4%B1)

19: <https://teknoloji.org/cnn-convolutional-neural-networks-nedir/>

20: <https://mgminsights.com/2021/09/19/convolutional-neural-network-cnn-nedir/>

21: [https://medium.com/databulls/yapay-sinir-ağlarında-aktivasyon-fonksiyonları-11002b8ac522](https://medium.com/databulls/yapay-sinir-a%C4%9Flar%C4%B1nda-aktivasyon-fonksiyonlar%C4%B1-11002b8ac522)

22: <https://devhunteryz.wordpress.com/2018/07/28/transfer-ogrenimi-transfer-learning/>

23: <https://devhunteryz.wordpress.com/2018/08/27/yapay-sinir-aglarindaki-hiperparametreler>

24: <https://coimer.medium.com/batch-normalization-b7d73c9cc6df>

25: <https://dergipark.org.tr/en/download/article-file/821607>

26: <https://en.wikipedia.org/wiki/Loss_function>

27: <https://tr.wikipedia.org/wiki/Algoritma>

Şekil 1: <https://www.kaggle.com/datasets/nih-chest-xrays/data?select=README_CHESTXRAY.pdf>

Şekil 2: [https://miro.medium.com/max/1400/0\*3rRcXavz8hmHXAFL.png](https://miro.medium.com/max/1400/0*3rRcXavz8hmHXAFL.png)

Şekil 3: [https://miro.medium.com/max/1400/1\*hYN6b13CKaTN1jnafkq5KQ.jpeg](https://miro.medium.com/max/1400/1*hYN6b13CKaTN1jnafkq5KQ.jpeg)

Şekil 4: [https://miro.medium.com/max/818/1\*abqja-rrroNvUkRUto2EIw.png](https://miro.medium.com/max/818/1*abqja-rrroNvUkRUto2EIw.png)

Şekil 5: [https://miro.medium.com/max/828/1\*FXyC5UElEGMGy3Z39qEKhg.png](https://miro.medium.com/max/828/1*FXyC5UElEGMGy3Z39qEKhg.png)

Şekil 6: [https://miro.medium.com/max/1400/0\*GOFqtFbvaeosJ8qf](https://miro.medium.com/max/1400/0*GOFqtFbvaeosJ8qf)

Şekil 7: <https://929687.smushcdn.com/2633864/wp-content/uploads/2021/03/hyperparameter_tuning_scikit_keras_idea.png?lossy=1&strip=1&webp=1>

Şekil 8: <https://upload.wikimedia.org/wikipedia/commons/thumb/a/ac/LampFlowchart_tr.svg/240px-LampFlowchart_tr.svg.png>

**7. ÖZGEÇMİŞ**

**Ayşegül Aleyna ÜNSAL**

Doğum Yeri: Sivas

Doğum Tarihi: 10.01.2000

Lise: 75. Yıl Anadolu Lisesi – Ankara/Çankaya

Üniversite: Tokat Gaziosmanpaşa Üniversitesi

Bölüm: Bilgisayar Mühendisliği

**Emirhan AYDIN**

Doğum Yeri: Tokat/Merkez

Doğum Tarihi: 16.01.1999

Lise: Sami Baklacı Anadolu Lisesi - Tokat/Turhal

Üniversite: Tokat Gaziosmanpaşa Üniversitesi

Bölüm: Bilgisayar Mühendisliği

**Sevdenur YILMAZ**

Doğum Yeri: Sivas/Merkez

Doğum Tarihi: 25.08.1999

Lise: Anadolu İmam Hatip Lisesi - Tokat/Turhal

Üniversite: Tokat Gaziosmanpaşa Üniversitesi

Bölüm: Bilgisayar Mühendisliği