

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**LEİSHMANİA PARAZİTİNİN DERİN ÖĞRENME
TABANLI GÖRÜNTÜ ANALİZİ İLE TESPİTİ**

**Hazırlayan
Enes ŞEKER**

**Danışman
Dr. Öğr. Üyesi İlker DALKIRAN**

Yüksek Lisans Tezi

**Temmuz 2023
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**LEİSHMANİA PARAZİTİNİN DERİN ÖĞRENME
TABANLI GÖRÜNTÜ ANALİZİ İLE TESPİTİ**

(Yüksek Lisans Tezi)

**Hazırlayan
Enes ŞEKER**

**Danışman
Dr. Öğr. Üyesi İlker DALKIRAN**

**Temmuz 2023
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Enes ŞEKER

İmza

“Leishmania Parazitinin Derin Öğrenme Tabanlı Görüntü Analizi ile Tespiti” adlı Yüksek Lisans Tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.

Hazırlayan

Enes ŞEKER

İmza

Danışman

Dr. Öğr. Üyesi İlker DALKIRAN

İmza

Elektrik-Elektronik Mühendisliği ABD Başkanı

Prof. Dr. Ferhat DALDABAN

TEŞEKKÜR

Bana çalışmalarım süresince her türlü yardımı ve bilgi birikimini esirgemeyip desteğini sunan, fikir ve tavsiyeleriyle bu çalışmanın ortaya çıkmasını sağlayan sayın hocam Dr. Öğr. Üyesi İlker DALKIRAN'a teşekkürü bir borç bilirim.

Veri setinin toplanmasında ve onaylanmasında desteğini veren sayın Doç. Dr. Ülfet ÇETİNKAYA'ya teşekkürlerimi sunarım.

Ayrıca; çalışmalarım süresince sabır göstererek bana her türlü maddi ve manevi destek veren değerli annem Handan ŞEKER'e ve babam Duran ŞEKER'e en içten teşekkürlerimi sunarım. Tez yazımında bana vermiş olduğu destekten ötürü meslektaşım sayın Furkan Said DURMUŞ'a da ayrıca teşekkür ederim.

Enes ŞEKER

Temmuz 2023, KAYSERİ

LEISHMANIA PARAZİTİNİN DERİN ÖĞRENME TABANLI GÖRÜNTÜ ANALİZİ İLE TESPİTİ

Enes ŞEKER

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Haziran 2023
Danışman: Dr. Öğr. Üyesi İlker DALKIRAN

ÖZET

Tıbbi görüntülerin analizi ve hastalık teşhisi için uzmanlık bilgisi gerekmektedir. Günümüz teknolojilerinin gelişmesiyle birlikte bu tıbbi görüntülerin analizinde ve sınıflandırılmasında yapay zekâ, makine öğrenmesi ve makine öğrenmesinin bir alt dalı olan derin öğrenme yöntemleri yaygın olarak kullanılmaktadır. Mikroskop görüntülerinde bulunan mikroorganizma ve parazitlerin teşhisi ve sınıflandırılması derin öğrenme algoritmaları sayesinde daha kısa sürede ve daha az uzmanlık bilgisi ile yapılabilmektedir. Bu çalışmada leishmania görüntüleri parazitli ve parazitsiz olmak üzere iki sınıfa ayrılmıştır. Evrişimli sinir ağları (Convolutional Neural Networks, CNN) algoritması kullanılarak sınıflandırma için modeller oluşturulmuştur. CNN tabanlı modeller ADAM ve RMSProp optimize edicileri ve dropout kullanılarak eğitilmiştir. Geri çağırma, kesinlik, özgüllük, F1 skoru ve Matthew korelasyon katsayısı (Matthews Correlation Coefficient, MCC) skoru gibi performans ölçüm metrikleri kullanılarak modellerin performansları karşılaştırılmıştır. %91.30'luk başarı gösteren bir model elde edilmiştir.

Anahtar Kelimeler: Derin Öğrenme, Evrişimli sinir ağı, Leishmania, Sınıflandırma, Görüntü İşleme

DETECTION OF LEISHMANIA PARASITE BY DEEP LEARNING BASED IMAGE ANALYSIS

Enes ŞEKER

**Erciyes University, Graduate School of Natural and Applied Sciences
Master Thesis, June 2023
Supervisor: Dr. İlker DALKIRAN**

ABSTRACT

Expert knowledge is required for analysis of medical images and diagnosis of disease. With the development of today's technologies artificial intelligence, machine learning and deep learning which is a sub-branch of machine learning are widely used in the analysis and classification of these medical images. Diagnosis and classification of microorganisms and parasites in microscope images can be done in a shorter time and with less specialized knowledge thanks to deep learning algorithms. In this study, leishmania images were divided into two classes as parasitic and non-parasitic. Models were created for classification using the Convolutional Neural Networks (CNN) algorithm. CNN-based models are trained using ADAM and RMSProp optimizers and dropout. The performances of the models were compared using performance measurement metrics such as recall, precision, specificity, F1 score and MCC (Matthew correlation coefficient) score. A model with a success rate of 91.30% was obtained.

Keywords: Deep Learning, Convolutional Neural Network, Leishmania, Classification, Image Processing

İÇİNDEKİLER

LEISHMANIA PARAZİTİNİN DERİN ÖĞRENME TABANLI GÖRÜNTÜ ANALİZİ İLE TESPİTİ

BİLİMSEL ETİĞE UYGUNLUK	ii
YÖNERGEYE UYGUNLUK.....	iii
KABUL VE ONAY	iv
TEŞEKKÜR.....	v
ÖZET	vi
ABSTRACT.....	vii
İÇİNDEKİLER	viii
KISALTMALAR	x
TABLolar LİSTESİ.....	xi
ŞEKİLLER LİSTESİ	xii
GİRİŞ	1

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

1.1. Hastalığın Durumu	3
1.2. Literatür Taraması	6
1.3. Tezin Amacı ve Önemi.....	9

2. BÖLÜM

MATERYAL ve YÖNTEM

2.1. Derin Öğrenme	10
2.1.1. Tekrarlayan Sinir Ağları.....	10
2.1.2. Uzun Kısa-Dönem Bellek (Long Short-Term Memory, LSTM).....	11
2.1.3. Evrişimli Sinir Ağları (CNN)	12
2.2. Aktivasyon Fonksiyonları.....	15
2.2.1. Sigmoid Aktivasyon Fonksiyonu	15
2.2.2 Hiperbolik Tanjant (tanh) Aktivasyon Fonksiyonu	16
2.2.3 ReLU (Rectified Linear Unit) Aktivasyon Fonksiyonu	17
2.2.4. Leaky ReLU Aktivasyon Fonksiyonu	18
2.4. Optimizasyon Algoritmaları	19
2.4.1. RMSprop.....	20
2.4.2. ADAM	21

2.5. Hata Matrisi (Confusion Matrix)	21
2.6. Model Geliştirmede Kullanılan Kütüphaneler ve Yazılım Dili	24

3. BÖLÜM

MODELİN OLUŞTURULMASI ve PERFORMANSININ ÖLÇÜLMESİ

3.1. Veri Seti Hazırlanması.....	25
3.2. Derin Öğrenme Modelinin Oluşturulması.....	27
3.4. Modellerin performansları.....	28
3.5. Önerilen Model Sonuçları	28

4. BÖLÜM

TARTIŞMA, SONUÇ ve ÖNERİLER

4.1.Tartışma	38
4.2.Sonuç ve Öneriler.....	39
KAYNAKÇA	40
ÖZGEÇMİŞ.....	44

KISALTMALAR

ADAM	: Uyarlanabilir moment tahmini (Adaptive Moment Estimation)
CNN	: Convolutional Neural Network (Evrişimli Sinir Ağı)
DNA	: Deoksiribo Nükleik Asit
DSÖ	: Dünya Sağlık Örgütü
FC	: Tam Bağlantı (Fully connected)
FN	: False Negative
FP	: False Positive
GPU	: Grafik İşleme Ünitesi (Graphics Processing Unit)
KL	: Kutanöz Leishmaniasis
Leaky ReLU	: Sızan Doğrultulmuş Doğrusal Ünite (Leaky Rectified Linear Unit)
LSTM	: Uzun Kısa-Dönem Hafıza (Long Short-Term Memory)
MCC	: Matthews Correlation Coefficient (Matthew Korelasyon Katsayısı)
ML	: Mukokutanöz Leishmaniasis
NLP	: Doğal Dil İşleme (Natural Language Processing)
PCR	: Polymerase Chain Reaction (Polimeraz Zincir Reaksiyonu)
PIL	: Python Imaging Library
RAN	: Residual Attention Network (Artık Dikkat Ağı)
ReLU	: Doğrultulmuş Doğrusal Ünite (Rectified Linear Unit)
ResNetV2	: Residual Network (Artık Ağ)
RMSprop	: Karelerin ortalamasının karekökü yayılımı (Root Mean Square Prop)
RNN	: Tekrarlayan sinir ağları (Recurrent neural network)
SGD	: Stokastik Gradyan İniş (Stochastic Gradient Descent)
TN	: True Negative
TNR	: True Negative Rate
TP	: True Positive
TPR	: True Positive Rate
VGG16	: The Visual Geometry Group (Görsel Geometri Grubu)
VL	: Visceral Leishmaniasis
YOLOv5	: You Look Only Once

TABLÖLAR LİSTESİ

Tablo 3.1.	CNN tabanlı oluşturulan derin öğrenme modellerinin hiperparametre değerleri.	27
Tablo 3.2.	Farklı hiperparametre değerlerine göre oluşturulan modellerin eğitimlerinin sonucunda elde edilen hata matrisi ile hesaplanan performans değerleri.	28



ŞEKİLLER LİSTESİ

Şekil 1.1	Viseral Leishmaniasis	4
Şekil 1.2.	Mukokutanöz Leishmaniasis	4
Şekil 1.3.	Kutanöz Leishmaniasis	5
Şekil 1.4.	Kutanöz Leishmaniasis	5
Şekil 1.5.	Jacques S.M kullanmış olduğu U-net algoritması	6
Şekil 1.6.	Qanbar M.M. kurmuş olduğu RAN ağı	8
Şekil 2.1.	Tekrarlayan Sinir Ağı (RNN) yapısı.....	11
Şekil 2.2.	LSTM hücresinin iç yapısı.....	11
Şekil 2.3.	LSTM modelinin mimarisi	12
Şekil 2.4.	Evrişimli Sinir Ağı (CNN) Yapısı	12
Şekil 2.5.	Dolgulama (Padding) işlemi	13
Şekil 2.6.	Maksimum Havuzlama (MaxPool) İşlemi.....	14
Şekil 2.7.	Ortalama Havuzlama (Average Pool) işlemi	14
Şekil 2.8.	Tam Bağlı Katman.....	15
Şekil 2.9.	Sigmoid Fonksiyonun Grafiği	16
Şekil 2.10.	Tanh Aktivasyon Fonksiyonu Grafiği	16
Şekil 2.11.	ReLU Aktivasyon Fonksiyonu Grafiği.....	17
Şekil 2.12.	Leaky ReLU Aktivasyon Fonksiyonun Grafiği.....	18
Şekil 2.13.	Düğüm Seyreltme Sinir Ağı Modeli (a) Standart Sinir Ağı Modeli (b) Düğüm Seyreltme Uygulanmış Sinir Ağı Modeli	19
Şekil 2.14.	Hata Matrisi	22
Şekil 3.1.	Veri Setinden Örnek Görüntü.	26
Şekil 3.2.	Veri setinde parazit içeren görüntüler.....	26
Şekil 3.3.	Veri setinde parazit içermeyen görüntüler.....	26
Şekil 3.4.	Önerilen Model 5'e ait akış şeması.....	29
Şekil 3.5.	Model 5'e ait eğitim doğruluk grafiği.....	29
Şekil 3.6.	Model 5'e ait eğitim kayıp grafiği.	30
Şekil 3.7.	Model 5'e ait hata matrisi.	31
Şekil 3.8.	Model 1'e ait eğitim doğruluk grafiği.....	32
Şekil 3.9.	Model 1'e ait eğitim kayıp grafiği.	32
Şekil 3.10.	Model 2'ye ait eğitim doğruluk grafiği.....	33
Şekil 3.11.	Model 2'ye ait eğitim kayıp grafiği.	33

Şekil 3.12.	Model 1'e ait hata matrisi.	34
Şekil 3.13.	Model 2'ye ait hata matrisi.	34
Şekil 3.14.	Model 3'e ait eğitim doğruluk grafiği.....	35
Şekil 3.15.	Model 3'e ait eğitim kayıp grafiği.	35
Şekil 3.16.	Model 4'e ait eğitim doğruluk grafiği.....	36
Şekil 3.17.	Model 4'e ait eğitim kayıp grafiği	36
Şekil 3.18.	Model 3'e ait hata matrisi.	37
Şekil 3.19.	Model 4'e ait hata matrisi.	37



GİRİŞ

Teknolojinin gelişmesiyle birlikte, bilişim teknolojilerinin insan hayatına etkisi gün geçtikçe artmaktadır. Çalışma hayatında, evlerde, okullarda, sağlık alanlarında bu teknolojilerin kullanımı gündelik işleri kolaylaştırmaktadır. Tıp alanında da kullanılan bu teknolojiler, hastalıkların teşhisi ve tedavisinde sağlık çalışanlarına yardımcı olmaktadır.

Hastalıkların teşhisi, sınıflandırılması ve tedavisinde makine öğrenmesi ve makine öğrenmesinin bir alt dalı olan derin öğrenme yöntemleri kullanılmaktadır. Tıbbi görüntülerin analizinin ve görüntülerden hastalık teşhisinin yapılması uzmanlık bilgisini gerektirmektedir. Derin öğrenme yöntemleri ile yapılan çalışmalar incelendiğinde daha az uzmanlık bilgisiyle daha kısa sürede hastalık teşhisi gerçekleştirilebilmektedir.

Leishmaniasis, leishmania protozoan mikroorganizmalar sebebiyle meydana gelen, dışı kum sineği (tatarcık, yakarca) ısırması sonucu bulaşan bir hastalıktır [1]. Leishmania parazitinin sebep olduğu hastalıklara bakacak olursak dünya genelinde paraziter hastalıklar arasında sıtma (malaria) hastalığından sonra en çok görülen ikinci hastalıktır. Bu hastalıkların tedavisi ve kontrolleri zordur. Dünyada ve ülkemizde görülen bu hastalığın teşhisinde iki yöntem kullanılmaktadır. Birinci yöntem, ciltte oluşan yaradan alınan bir örneğin giemsa boyama tekniği uygulandıktan sonra mikroskop altında incelenmesidir. Giemsa boyama parazitlerin histopatolojik olarak tanısında kullanılan nükleik asit boyasıdır [2]. İkincisi ise Polimeraz Zincir Reaksiyonu (Polymerase Chain Reaction- PCR) yöntemidir [3]. PCR yöntemi, deoksiribo nükleik asit (DNA) içindeki özgün bir bölgeyi enzimatik olarak çoğaltmak üzere uygulanan yöntemin adıdır. Kısaca DNA replikasyonunun belirli bir sayıda sürekli tekrar ettirilmesi işlemidir [4].

Bu tez çalışmasında, mikroskop görüntülerinden elde edilen leishmania verilerinin derin öğrenme algoritmaları kullanılarak sınıflandırılması amaçlanmıştır. Erciyes Üniversitesi Halil Bayraktar M.Y.O'dan ve açık kaynak olarak verilen internet sitesinden elde edilen leishmania mikroskop görüntülerinin, derin öğrenme algoritmasının bir çeşidi olan

evriřimli sinir ađı (Convolutional Neural Networks, CNN) kullanılarak analizleri yapılmıřtır. Veri seti parazitli ve parazitsiz olarak iki sınıfa ayrılmıřtır [5].

Bu alıřmada yer alan grntlerin sınıflandırılması iin hazırlanan derin đrenme modeline ait kodlar Python yazılım diliyle yazılmıřtır. Veri setinin hazırlanmasında Numpy ve Python grntleme kitaplıđı (Python Imaging Library, PIL) ktphaneleri kullanılmıřtır. Derin đrenme modelinin oluřturulmasında Keras ktphanesi altyapısını kullanan Tensorflow ktphanesinden yararlanılmıřtır. Model tasarımı JupyterLab aık kaynaklı web arayznde yapılmıř olup eđitimde NVIDIA GeForce GTX 950M grafik iřlemci nitesi (Graphics Processing Unit, GPU) kullanılmıřtır.



1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

1.1. Hastalığın Durumu

Derin öğrenme yöntemleriyle mikroskop görüntülerinden parazit teşhisi, doktorlar için bir vakit kazancı olmaktadır. Yanlış teşhis yapılmasını ve gereksiz ilaç kullanım oranını azaltabilmektedir. Derin öğrenme tabanlı mikroskobik parazit teşhisi uygulamalarını kullanan klinisyenler, gelişmiş karar verme görevlerine de daha fazla zaman ayırabilir [6]. Ayrıca derin öğrenmenin çoklu parazit teşhisi ve gerçek zamanlı parazit tanıma uygulamalarında, görüntüde düşük parazit yoğunluğu olmasına rağmen yüksek verim alınması avantajı vardır [6].

Leishmania mikroorganizmasının bir yüzyıldan fazla süre önce keşfedilmiş olmasına rağmen özellikle gelişmekte olan ülkeler başta olmak üzere çoğu ülkede bu parazit önemli bir sağlık problemi teşkil etmektedir [4]. Parazitin neden olduğu hastalığın tedavisi ve kontrolünün zorluğu nedeniyle dünyada paraziter hastalıklar içinde sıtmadan (Malaria) sonra hastalık önem bakımından ikinci sırada gelmektedir [4]. Leishmania türüne göre enfeksiyon kutanöz, mukokutanöz veya viseral olarak 3 türde hastalıkla sonuçlanır [7]. Bu hastalıklar aşağıda verilmiştir.

- Viseral Leishmaniasis (VL)
- Mukokutanöz Leishmaniasis (ML)
- Kutanöz Leishmaniasis (KL)

Viseral Leishmaniasis sineğin ısırığından sonra aylar içinde ilerler. Bu hastalık genellikle dalak, karaciğer ve kemik iliği gibi çeşitli iç organlara etki etmektedir [7]. Yüksek ateş, ciddi kilo kayıpları, dalak ve karaciğerde büyüme görülmektedir [7]. Tedavi edilmezse hastalarda %95 gibi bir oranda ölüm görülmektedir. Dünya Sağlık Örgütü (DSÖ)

verilerine göre vakaların çoğu genelde Brezilya, Doğu Afrika ve Hindistan'da görülmektedir. Her yıl tahmini 50.000 ila 90.000 yeni vaka görülmektedir [7]. Şekil 1.1'de viseral leishmaniasis'in etkisi görülmektedir.



Şekil 1.1 Viseral Leishmaniasis [8].

Mukokutanöz leishmaniasis bu üç hastalık arasında en az görülen formudur. Kutanöz leishmaniasis'e neden olan parazitlerin bazı türleri bu forma sebep olmaktadır [7]. Burun, ağız ve boğazdaki mukoza zarlarının tahrip olmasına yol açmaktadır. DSÖ verilerine göre Mukokutanöz leishmaniasis vakalarının %90'ından fazlası Brezilya, Etiyopya, Bolivya ve Peru'da görülmektedir [7]. Mukokutanöz leishmaniasis'in insan vücuduna etkileri Şekil 1.2'de verilmiştir.



Şekil 1.2. Mukokutanöz Leishmaniasis [8].

Halk arasında şark çıbanı olarak da bilinen Kutanöz leishmaniasis, leishmaniasis'in en yaygın şeklidir ve cilt lezyonlarına neden olmaktadır [1]. Ömür boyu ciltte yara izleri ve ciddi sakatlanmalar bırakmaktadır [1]. DSÖ verilerine göre Kutanöz leishmaniasis vakalarının yaklaşık %95'i Amerika, Akdeniz Havzası, Orta Doğu ve Orta Asya'da görülmektedir. Her yıl dünya çapında 700.000 ila 1 milyon yeni vakanın ortaya çıktığı belirtilmiştir [7]. Kutanöz leishmaniasis'in deri üzerinde açtığı yaralar, Şekil 1.3 ve 1.4'te gösterilmiştir.

Türkiye'de ise KL çoğunlukla Güneydoğu Anadolu bölgesinde ve Akdeniz bölgesinin Çukurova yöresinde endemik olmakla beraber Şanlıurfa başta olmak üzere birçok ilde görülmüştür [3]. KL birçok deri hastalığı ile karışabilmektedir. Bu yüzden hastalığın tanısı amacıyla iki yöntem kullanılmaktadır. Birinci yöntem, ciltteki oluşan yaradan alınan örneğin giemsa ile boyanmasının ardından mikroskopik olarak incelemesidir. İkinci yöntem ise son yıllarda yapılmaya başlayan PCR yöntemidir [3].



Şekil 1.3. Kutanöz Leishmaniasis [3].

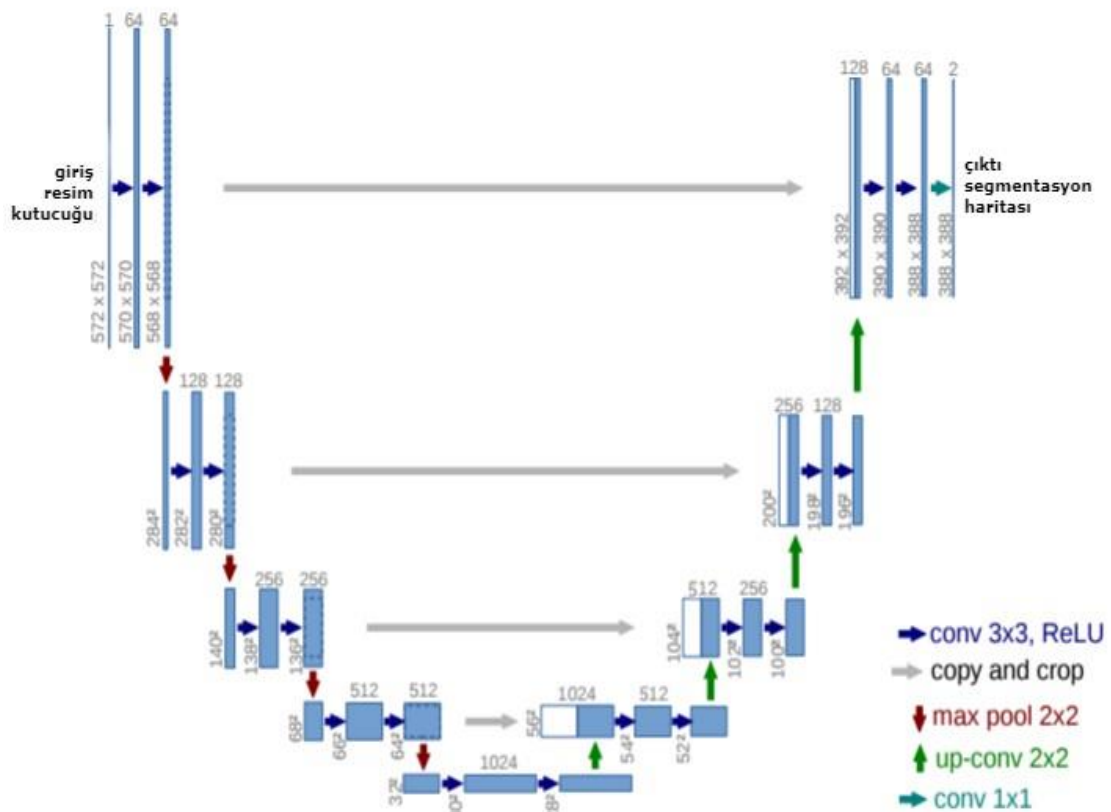


Şekil 1.4. Kutanöz Leishmaniasis [3].

1.2. Literatür Taraması

Literatür incelendiğinde leishmania başta olmak üzere parazitlerin sınıflandırması ve görüntülerin analizi için derin öğrenme algoritmaları ve kütüphaneleri kullanılarak birçok çalışma yapıldığı gözlemlenmiştir.

Chi Zang ve arkadaşları, farklı türde parazitlerin algılama, segmentasyon, izleme ve sınıflandırma analizlerinin yapıldığı ve literatüre geçmiş çalışmaları özetlemişlerdir [6]. Her bir protozoan parazit için ayrı evrişimli sinir ağı mimarileri tercih edilmiştir. Açık kaynaklı veri setleri ve çeşitli mimariler için farklı değerlendirme metrikleri kullanıldığını belirtmişlerdir. Protozoan parazitlerin farklı morfolojik yapılarının ve gelişim evrelerinin olduğunu ve bu nedenle protozoan parazitlerin tespitinde çeşitli yöntemler ve mimariler kullanılarak derin öğrenme çalışmalarının yapılması gerektiği vurgulanmıştır [6].



Şekil 1.5. Jacques S.M kullanmış olduğu U-net algoritması [9].

Jacques S.M. yapmış olduğu tez çalışmasında, kendisinin elde ettiği kırk beş adet mikroskopik görüntüden oluşan veri setine, görüntü ön işleme tekniklerini uygulayıp,

ardından veri etiketleme işlemlerini gerçekleştirmiştir. Evrişimli sinir ağı tabanlı U-net mimarisi ile 20 turda Adam optimizer tercih edilmiş, iki farklı eğitim yapılmıştır. Doğruluk hesabı için metrik olarak kesinlik, geri çağırma ve bu iki ölçümün harmonik ortalaması olan F1 skorunu kullanmıştır. Ancak leishmania parazitinin amastigot ve promastigot segmentasyonunun istediği seviyede olmadığı ve daha fazla geliştirilmesi gerektiğini belirtmiştir [9]. Kullanılan U-net algoritması Şekil 1.5'te gösterilmiştir.

Zare ve arkadaşları, 300 görüntüden oluşan veri setini kullanarak leishmania parazitinin tespiti için makine öğrenmesi tabanlı çalışma yapmışlardır. Eğitim için Viola-Jones algoritması kullanılmış olup, algoritma özellik çıkarma, integral görüntü oluşturma ve sınıflandırma adımlarını takip etmektedir [10]. Öğrenmeyi güçlendirmek ve sınıflandırma performansını artırmak için Adaboost algoritmasını, sistemin performansını değerlendirmek için duyarlılık oranını (sensitivity) ve özgüllük oranını (specificity) kullanmışlardır. Elde ettikleri sonuçlarda %83'lük bir duyarlılık ve %35'lik özgüllük oranı belirtmişlerdir [10].

Qanbar M.M. yapmış olduğu tezinde, protozoan parazitlerin çeşitlerinden birinin neden olduğu malarya (sıtma) hastalığı üzerine çalışmıştır. 27,558 adet veri setini Google'ın açık kaynaklı hizmeti olan Tensoflow kaynaklarından ve ABD ulusal tıp kütüphanesinden almıştır. 20,658 görüntüyü eğitim için 6,900 görüntüyü ise test için ayırmış olup, evrişimli sinir ağları ve destek vektör makinelerinde eğitmiştir. Evrişimli sinir ağı mimari örneklerinden, VGG16, ResNet50, InceptionV3, AlexNet'i ve CNN tabanlı Artık Dikkat Ağlarını (Residual Attention Networks, RAN) kullanmıştır ve bu modellerin birbirlerine göre avantajlarının ve dezavantajlarının incelemesi yapılmıştır. Performans değerlendirmesi için duyarlılık, kesinlik, F-skoru gibi ölçütler kullanmıştır. Ayrıca %95,79'luk bir doğruluk oranı ile RAN en yüksek başarıyı elde etmiştir [11]. RAN ağı Şekil 1.6'da yer almaktadır.

Optimizasyon algoritması olarak Relu, aşırı öğrenmeye karşı dropout kullanılmıştır. Giriş görüntüsü 128x128 boyutunda olup 3 ayrı tür için eğitim yapılmıştır. Eğitim sonucunda ilk tür için kesinlik %100 ve geri çağırma %93.7 , ikinci tür için kesinlik %93.3 ve geri çağırma %87.5, üçüncü tür için ise kesinlik %86.6 ve geri çağırma 82.8 olarak tespit etmişlerdir. Sonuç olarak ise model % 93.3'lük bir doğruluk göstermiştir [13].

1.3. Tezin Amacı ve Önemi

Yapılan literatür taraması neticesinde makine öğrenmesi ve derin öğrenme algoritmaları ile parazitlerin analizi, sınıflandırılması ve tespiti örnekleri gözlemlenmiştir. Çalışmalar derin öğrenmenin sağlık alanında daha etkin bir biçimde kullanıldığını ve hastalıkların erken teşhis edilmesi ve dolayısıyla hastalığın ilerlemeden tedavi sürecine erken başlanmasına fayda sağlanacağını göstermektedir.

Leishmania parazitinin sebep olduğu hastalıklar, sıttmadan sonra dünya genelinde ikinci sırada olması ve ülkemizde de görülmesi sebebiyle bu parazit tercih edilmiştir. Derin öğrenme yöntemlerinden CNN mimarisinin görüntü analizine en uygun olanı olarak belirlenmesinden ötürü çalışma bu mimari ile tamamlanmıştır. Yapılan çalışmanın pratikte kullanılması ve gelecek çalışmalara katkı sağlayacağı değerlendirilmektedir.

2. BÖLÜM

MATERYAL ve YÖNTEM

2.1. Derin Öğrenme

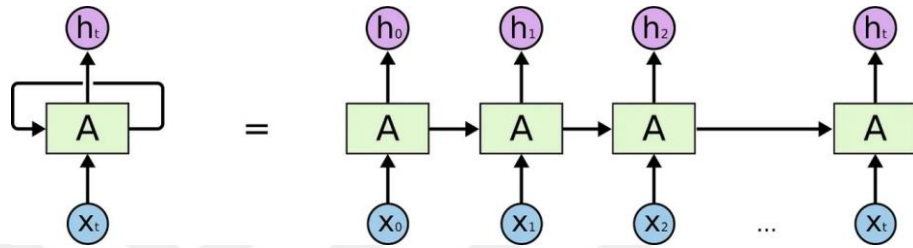
Derin öğrenme (Deep Learning), makine öğrenmesinin (Machine Learning) bir alt dalıdır. Makine öğrenmesinden farklı olarak derin öğrenme insan beyninin davranışını baz alarak sinir hücrelerini taklit etmektedir. İnsanın tecrübelerinden öğrendiği gibi derin öğrenme algoritmaları da sonucu iyileştirebilmek için sürekli gelişim halindedir. Derin öğrenme algoritmaları yapay sinir hücrelerinden meydana gelmektedir. Bu yapay sinir hücreleri birbirine bağlı üç veya daha fazla katmandan meydana gelir ve bu katmanların birçoğu gizli katmanlardır. Giriş ve çıkış katmanları dışındaki ara katmanların tamamı gizli katmanlar olarak isimlendirilmektedir. Gizli katmanların sayısı o ağ yapısını oluşturan kişinin insiyatifindedir. Eğitimin tamamlanması için büyük miktarda veri setine ihtiyaç duyulmaktadır [14, 15].

Yapılan literatür taraması neticesinde derin öğrenme modellerinden evrişimli sinir ağları (Convolutional Neural Network- CNN), yinelemeli sinir ağları (Recurrent Neural Network- RNN) ve uzun kısa süreli bellek (Long short-term memory- LSTM) incelenmiştir. İnceleme sonucunda evrişimli sinir ağlarının görsellerden elde edilen bilgilerin analiz edilmesinde daha etkili olduğu gözlemlenmiştir.

2.1.1. Tekrarlayan Sinir Ağları

Tekrarlayan sinir ağları (Recurrent Neural Network- RNN) önceki düğümden gelen bilgiyi kullanarak bir sonraki adımı tahmin etmek için kullanılan derin öğrenme modellerinden biridir. Diğer derin öğrenme modellerinden farkı, tekrarlayan sinir ağlarının hatırlama özelliğine sahip olmasıdır. Başka bir ifade ile tekrarlayan sinir ağları dışındaki sinir ağlarının girdileri birbirinden bağımsız iken, tekrarlayan sinir ağları önceki

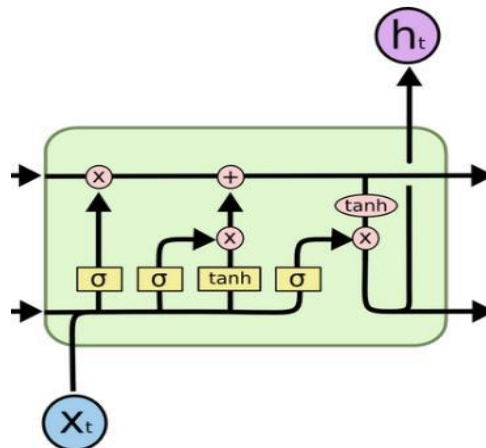
çıktıları girdi olarak kullanmaktadır. Gizli katmanın girişi önceki zaman adımından ve diğer gizli katmandan oluşmaktadır. Bu yapı ise RNN modelinin çoğunlukla doğal dil işleme (Natural Language Processing, NLP) ve konuşma tanıma (Speech Recognition) uygulamalarında kullanılmasını tercih sebebi haline getirmektedir. Ayrıca müzik üretimi, duygu sınıflandırma, karakter üretimi gibi uygulamalarda da kullanılmaktadır [16-18]. Tekrarlayan sinir ağının yapısı Şekil 2.1’de verilmiştir. Şekil 2.1 incelendiğinde çıkan sonuç aslında bir sonraki adımı beslemektir.



Şekil 2.1. Tekrarlayan Sinir Ağı (RNN) yapısı [16].

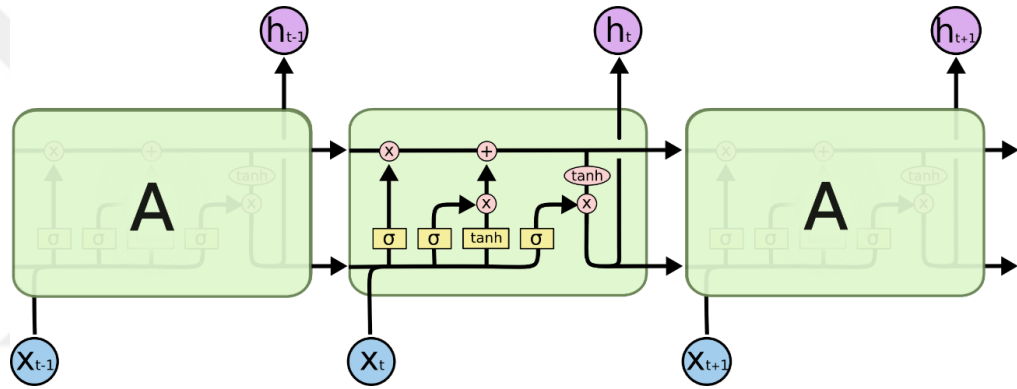
2.1.2. Uzun Kısa-Dönem Bellek (Long Short-Term Memory, LSTM)

Uzun kısa- dönem bellek yapıları, 1997 yılında Sepp Hochreiter ve Jürgen Schmidhuber tarafından kaybolan gradyanlar problemine çözüm olarak geliştirilmiştir. LSTM’in yapısı aynı RNN mimarisindeki gibi önceki katmandan gelen giriş alınarak kullanılmaktadır. RNN mimarisine göre hücre yapısı farklı olup RNN’e göre hafızada daha uzun süre bilgi tutabilmektedir. LSTM yapısı, özel bir RNN türüdür [19,20]. Şekil 2.2’de LSTM hücresinin iç yapısı verilmiştir. Standart ileri beslemeli sinir ağ yapılarının aksine, LSTM’lerin geri bildirim bağlantıları mevcuttur [19].



Şekil 2.2. LSTM hücresinin iç yapısı [17].

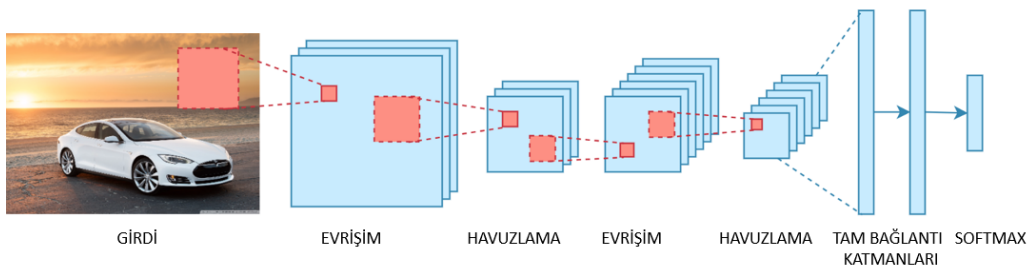
Tekrarlayan sinir ağlarının tekrar eden modüllerinden oluşan yapı, zincir sinir ağı biçimine sahiptir. Tekrarlanan modülün ise basit tek hiperbolik tanjant katmanı olan bir yapısı mevcuttur. LSTM'ler ise aynı RNN'ler gibi bu benzer yapıya sahiptir, fakat tekrar eden modülün farklı yapısı vardır. Şekil 2.3'de gösterilen yapıya bakıldığında, hücrede tek bir sinir ağı katmanına sahip olmak yerine daha gelişmiş etkileşime giren dört adet yapı mevcuttur. LSTM mimarileri konuşma ve metin işleme uygulamalarında iyi sonuçlar göstermektedir. Ayrıca müzik bestesi üretiminde de kullanılmıştır. RNN ile nota tahminleri yapılmasına karşın dinlenebilir bir beste yapmak mümkün olmamıştır. LSTM ile müzik yapılabilecek model tasarlanmıştır [20,21].



Şekil 2.3. LSTM modelinin mimarisi [20].

2.1.3. Evrişimli Sinir Ağları (CNN)

Evrişimli sinir ağları (Convolutional Neural Network- CNN ya da ConvNet) hayvanların görme merkezinden esinlenerek ortaya çıkan bir derin öğrenme mimarisidir. CNN mimarisi bilgisayarlı görme, görüntü işleme, görüntü sınıflandırma, ses işleme, doğal dil işleme ve video tanıma gibi birçok alanda kullanılmaktadır. Biyomedikal görüntülerin işlenmesinde de tercih edilmektedir. Genel bir CNN yapısı Şekil 2.4'de verilmiştir [21].



Şekil 2.4. Evrişimli Sinir Ağı (CNN) Yapısı [22].

Evrişimli sinir ağı yapı olarak 3 bölümden meydana gelmektedir. Bunlar:

- Evrişim katmanı (CONV)
- Ortaklama katmanı (POOL)
- Tam Bağlantı (Fully Connected, FC)

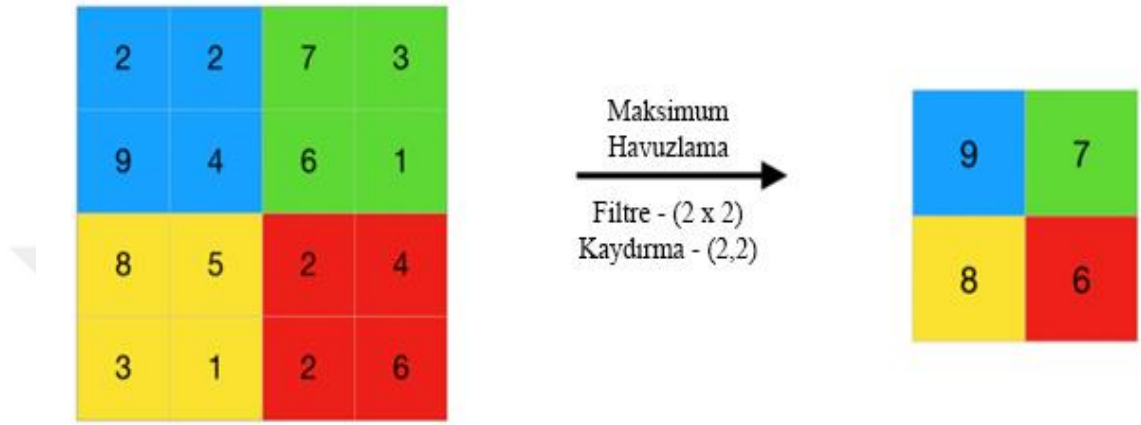
Evrişim katmanı, giriş katmanına evrişim işlemini uygulayarak çıkan sonucu bir sonraki katmana geçirir. Evrişim işlemi görüntü matrisi üzerinde dolaşan, görüntü matrisinden daha küçük boyutta olan filtrelerin noktasal çarpımı işlemidir. Kullanılan filtrenin boyutu ve filtrenin sayısı ayarlanırken, verilen görüntü matrisinin boyutları göz önünde bulundurulmalıdır. CNN algoritmasında modelin öğrendiği parametreler, filtredeki değerlerdir. Dolayısıyla model, sürekli olarak bu değerleri güncel tutup, görüntüden çıkarılan özellikleri daha iyi tespit etmeye çalışır. Ayrıca görüntülerde bazı filtre işlemleri uygulanırken ekstra iki işlem daha uygulanabilmektedir. Bunlar adım aralığı (stride) ve dolgu (padding) işlemleridir. Adım işlemi, filtrenin ana görsel üzerinde kaç birim kayacağını belirler. Şekil 2.5’de gösterilen dolgu işlemi ise görüntünün etrafını sıfırlarla doldurma işlemidir. Filtre uygulandığında çıktı orijinal görüntüden küçük olur. Bunu önlemek için dolgu işlemi uygulanmaktadır [22].



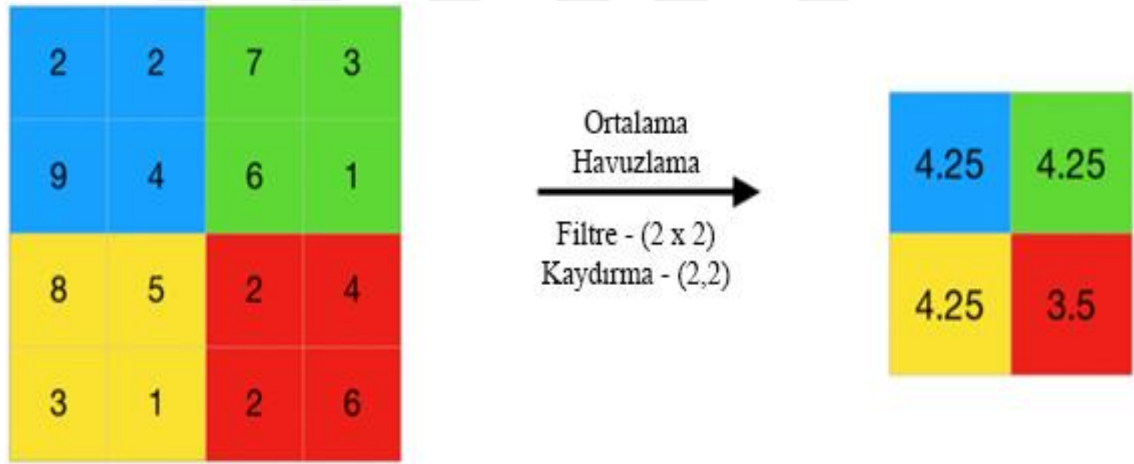
Şekil 2.5. Dolgulama (Padding) işlemi [22].

Havuzlama katmanının amacı, görüntü çok büyük olduğunda parametre sayısı azaltılarak görüntünün küçültülmesidir. Boyut küçülürken önemli bilgiler korunur. İki tür havuzlama yöntemi mevcuttur. Birincisi maksimum havuzlama yöntemidir. İkincisi ise ortalama havuzlama yöntemidir. Şekil 2.6’da gösterilen maksimum havuzlama özellik haritasının

filtre tarafından kapsanan bölgeden maksimum değeri seçen bir işlemdir. Ortalama havuzlama ise özellik haritasının filtre tarafından kapsanan bölgede bulunan değerlerin ortalamasının hesapladığı işlemdir. Şekil 2.7’de ortalama havuzlama işlemi gösterilmiştir. Genellikle maksimum yöntem tercih edilmektedir [22,23].

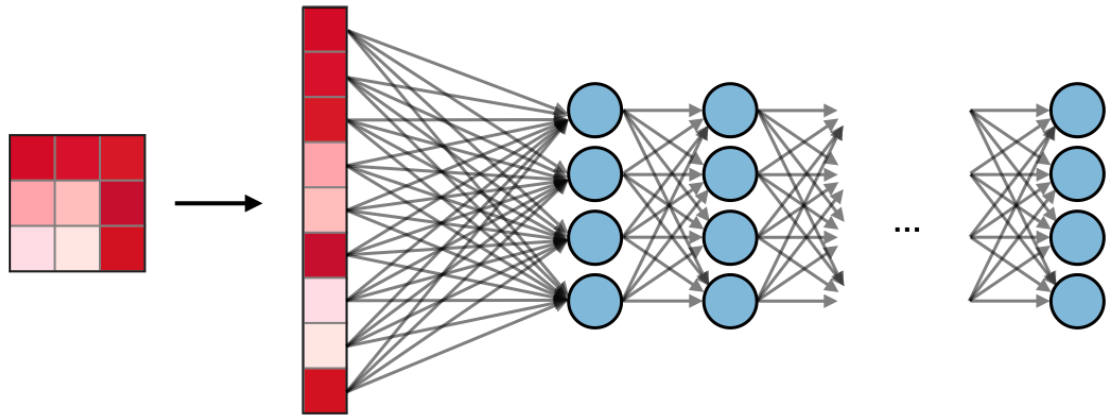


Şekil 2.6. Maksimum Havuzlama (MaxPool) İşlemi [23].



Şekil 2.7. Ortalama Havuzlama (Average Pool) işlemi [23].

Tam bağlı katman CNN mimarisinin son adımıdır. Tam bağlantı katmanları bir katmanda bulunan düğüm ve nöronların sonraki katmanlara aktarıldığı yapay sinir ağı tipidir. Evrişim işleminden sonra düzleştirme (flatten) işlemi uygulanır ve evrişim katmanı düz bir vektör haline geldikten sonra tam bağlı katmanlara beslenir. Bu katmanlar gizli katmanlardan oluşmaktadır ve çıktı bu katmanlardan sonra elde edilir [22].



Şekil 2.8. Tam Bağlı Katman [24].

2.2. Aktivasyon Fonksiyonları

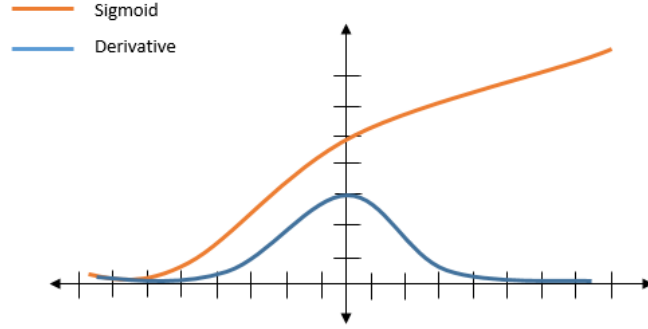
Aktivasyon fonksiyonları, bilginin bir sonraki nörona aktarılıp aktarılmayacağına karar veren fonksiyonlardır. Derin öğrenme algoritmalarında, öğrenme sürecini iyi bir şekilde gerçekleştirme, aşırı öğrenmeyi engelleme, doğruluk performansını artırma ve hesaplama maliyetini düşürme gibi işlemlerin gerçekleştirilmesi için geliştirilmişlerdir. Derin öğrenme modellerinde kullanılmadığı takdirde, derin öğrenme mimarileri doğrusal regresyon gibi davranır ve bu da öğrenme sürecindeki istenilen başarı performansını göstermez. Dolayısıyla bu mimarilerde doğrusal olmayan aktivasyon fonksiyonlarının kullanılması gerekmektedir. Aktivasyon fonksiyonlarından türevlenebilir olma, lineer olmama ve yerel optimuma takılmadan global optimuma ulaşma gibi özelliklere sahip olması beklenmektedir [21].

En çok tercih edilen aktivasyon fonksiyonları sigmoid, hiperbolik tanjant (tanh), doğrultulmuş doğrusal ünite (Rectified Linear Unit -ReLU) ve sızan doğrultulmuş doğrusal ünite (Leaky ReLU).

2.2.1. Sigmoid Aktivasyon Fonksiyonu

Sigmoid fonksiyonu, girişe uygulanan değeri (0,1) aralığına dönüştürmektedir. Sigmoid fonksiyonunun denklemi, Denklem 2.12' de, grafiksel gösterimi ise Şekil 2.9' da gösterilmiştir [25].

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.1)$$

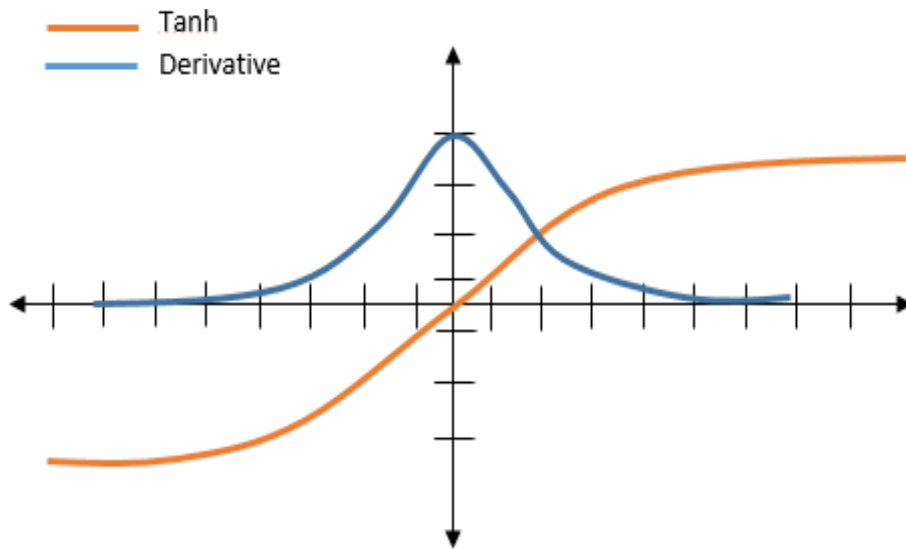


Şekil 2.9. Sigmoid Fonksiyonun Grafiği [25].

2.2.2 Hiperbolik Tanjant (tanh) Aktivasyon Fonksiyonu

Tanh (hiperbolik tanjant) aktivasyon fonksiyonu, sigmoid fonksiyonuna benzer işlem yapan bir fonksiyondur. Sigmoid fonksiyonundan farklı olarak yapılan işleme verilen değerleri (-1,1) aralığında değerlere dönüştürmektedir. Şekil 2.10' da grafiği verilen tanh fonksiyonunun denklemi, Denklem 2.2'deki gibidir [25].

$$f(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.2)$$



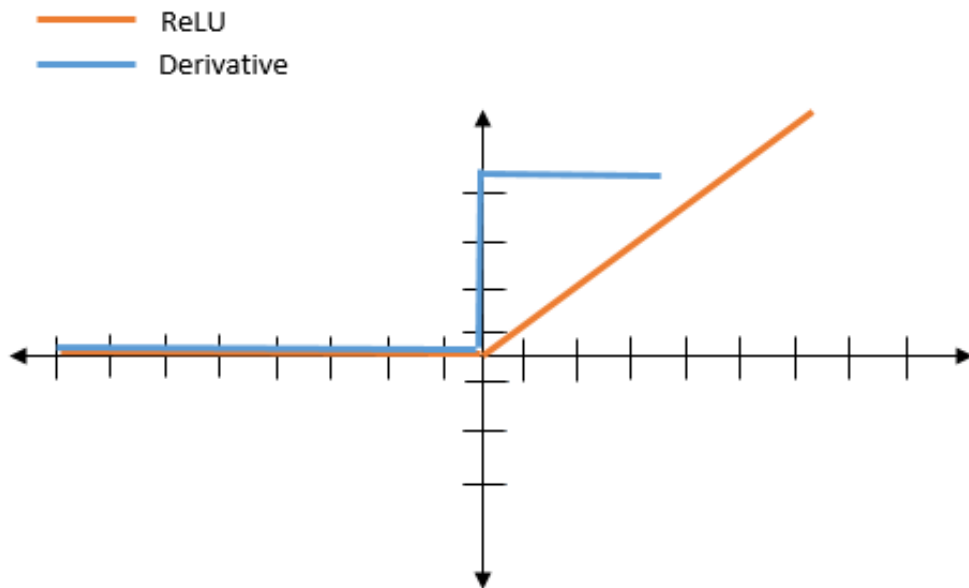
Şekil 2.10. Tanh Aktivasyon Fonksiyonu Grafiği [25].

2.2.3 ReLU (Rectified Linear Unit) Aktivasyon Fonksiyonu

Doğrultulmuş doğrusal ünite (Rectified Linear Unit -ReLU) aktivasyon fonksiyonu, türev işlemini gerçekleştirebilen doğrusal olmayan bir aktivasyon fonksiyonudur. Sigmoid ve hiperbolik tanjant aktivasyon fonksiyonlarında belirli bir seviyeden sonra türev alınamadığından dolayı kaybolan gradyan problemleri meydana gelmektedir. Bu sebeple kaybolan gradyan problemine çözüm bulmak için ReLU aktivasyon fonksiyonu geliştirilmiştir. Denklem 2.3’de aktivasyon fonksiyonu gösterilmiştir.

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (2.3)$$

ReLU, kaybolan gradyan problemini çözdüğü için genel olarak tercih edilmektedir. Ancak, ReLU aktivasyon fonksiyonunun negatif bölge sorunu olduğundan dolayı negatif değerleri sıfır olur ve türevi alınamaz. Bundan dolayı eğitim süreci yavaşlar. En büyük avantajları ise hesaplama yükünün diğer aktivasyon fonksiyonlarına göre düşük olmasıdır. Bundan dolayı da çok katmanlı mimarilerde en yaygın kullanılan aktivasyon fonksiyonlarından biridir. ReLU aktivasyon fonksiyonunun grafiği Şekil 2.11’de verilmiştir [25].

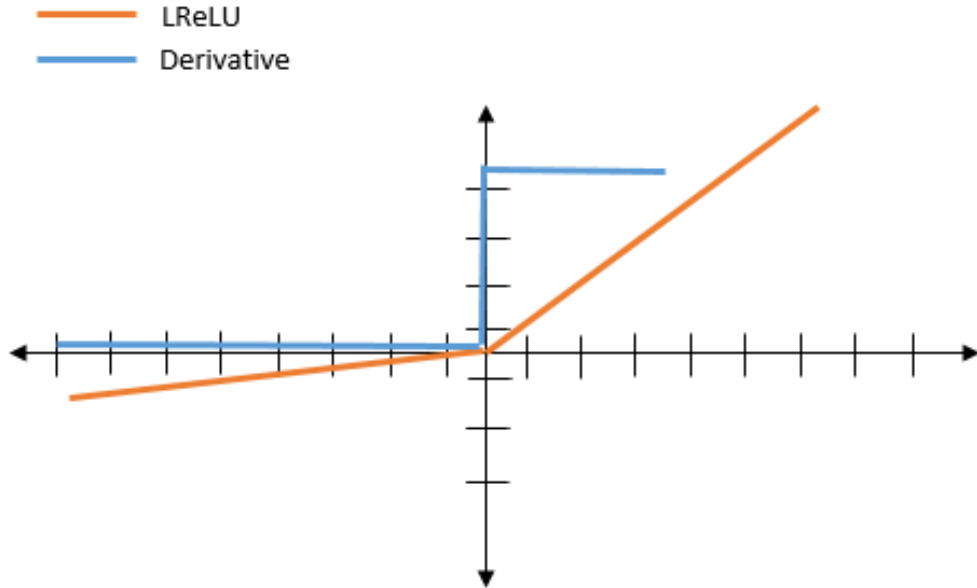


Şekil 2.11. ReLU Aktivasyon Fonksiyonu Grafiği [25].

2.2.4. Leaky ReLU Aktivasyon Fonksiyonu

Sızan doğrultulmuş doğrusal ünite (Leaky Rectified Linear Unit- LReLU) aktivasyon fonksiyonu, ReLU aktivasyon fonksiyonunun negatif bölge sorununun üstesinden gelmek amacıyla geliştirilmiştir. ReLU fonksiyonunun gelişmiş bir türüdür. Sıfırdan küçük giriş değeri olduğunda aktivasyon fonksiyonunu sıfır olarak görmek yerine Denklem 2.4'teki gibi değeri görülmektedir. Leaky ReLU'nün grafiği, Şekil 2.12' de verilmiştir [24].

$$f(x) = \begin{cases} \alpha x, & x < 0 \text{ ve } \alpha = 0.01 \\ x, & x \geq 0 \end{cases} \quad (2.4)$$

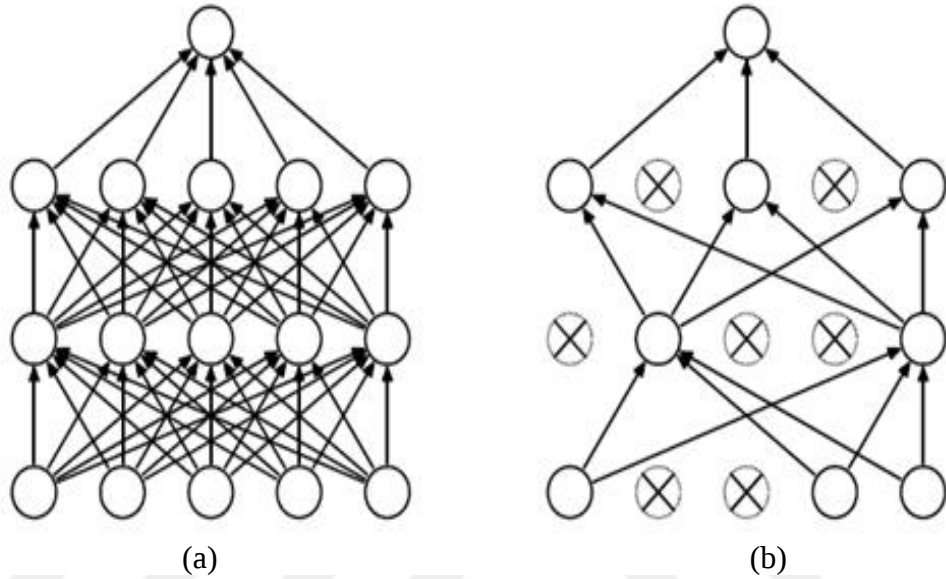


Şekil 2.12. Leaky ReLU Aktivasyon Fonksiyonun Grafiği [25].

2.3. Düşüm Seyreltme (Dropout)

Çok sayıda parametreye sahip derin öğrenme modelleri, çok güçlü derin öğrenme modelleridir. Fakat bu tür modellerde aşırı öğrenme (overfitting) önemli bir sorun teşkil etmektedir. Aşırı öğrenme eğitim sırasında veri seti üzerinde gereğinden fazla çalışıp modelin ezber yapmasıdır. Düşüm seyreltme bu sorunu ele alan bir yöntemdir. Düşüm seyreltme işlemi tam bağlı katmanlardan sonra kullanılır. Eğitim sırasında gizli birimlerin arasındaki bağların rasgele olarak koparılması işlemidir. Sinir ağının incelenmesi yani daha az bilgi iletilmesi aşırı öğrenmenin engellenmesini sağlar. Sabit bir p olasılık değeri belirlenir ve bu p olasılık değeri her bir katman için ne oranda seyreltme işleminin

yapılacağını belirler. Test sırasında kullanmak mümkün değildir. Çünkü test zamanında kullanılırsa bu yalnızca tahminlere gürültü ekler. Şekil 2.13’de düğüm seyreltme işleminin sinir ağı örneği gösterilmiştir [26].



Şekil 2.13. Düğüm Seyreltme Sinir Ağı Modeli (a) Standart Sinir Ağı Modeli (b) Düğüm Seyreltme Uygulanmış Sinir Ağı Modeli [26].

2.4. Optimizasyon Algoritmaları

Gradyan iniş (Gradient descent), sinir ağlarını optimize etmek için kullanılan en popüler ve yaygın algoritmalarından biridir. Sinir ağının tahmininde düşük hata elde edebilmesi için optimum parametre değerlerini hesaplar. Gradyan inişin 3 farklı varyantı mevcuttur. Bunlar toplu gradyan iniş (Batch Gradient Descent), stokastik gradyan iniş (Stochastic gradient descent, SGD) ve mini toplu gradyan iniştir (Mini-batch gradient descent) [27].

Toplu gradyan iniş, her parametre güncellemesinde tüm veri kümesinin gradyanlarını hesaplayarak işlem yapmaktadır. Toplu gradyan inişin dezavantajı veri kümesinin tümünü alması nedeniyle çok yavaş olabilmesi ve belleğe sığmayan veri kümeleri için eğitimin zor olmasıdır. Denklem 2.5’te toplu gradyan inişin formülü gösterilmiştir [27].

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta) \quad (2.5)$$

Stokastik gradyan iniş, her bir giriş örneği $x^{(i)}$ ve çıkış etiketi $y^{(i)}$ için ayrı ayrı parametre güncellemesi yapmaktadır. Toplu gradyan iniş her parametre güncellemesinden önce gradyanları yeniden hesapladığından büyük veri kümelerinde gerek olmayan hesaplamalar yapar. SGD ise her seferinde bir güncelleme yaptığı için bu gereksiz hesapları ortadan kaldırır. Bu nedenle genellikle çok hızlıdır. Denklem 2.6' da formülü verilmiştir [28].

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)}) \quad (2.6)$$

Mini toplu gradyan iniş, hem toplu gradyan iniş hem de SGD'nin özelliklerini kullanarak her bir mini grup için bir güncelleme gerçekleştirir. Büyük veri setleri üzerinde eğitim yapılacağı zaman mini gruplara bölünerek parça parça eğitim tamamlanır. Her bir turda yeni bir mini parti eğitime alınır. Veri seti eğitim sırasında genel olarak 50-256 olacak şekilde mini parti gruplarına ayrılmaktadır. Yapılacak çalışmaya göre bu değişkenlik gösterebilir. Denklem 2.7'de formülü gösterilmiştir [28].

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i:i+n)}; y^{(i:i+n)}) \quad (2.7)$$

2.4.1. RMSprop

Karelerin ortalamasının karekökü yayılımı (Root Mean Square Propagation- RMSprop), Geoff Hinton tarafından Coursera' da önerilen yayınlanmış, uyarlanabilir bir öğrenme hızı yöntemidir. Adagrad optimizasyon algoritmasının modifiye edilmesiyle elde edilmiştir. Adagrad optimizasyon algoritmasında her parametrenin kendi öğrenme hızı bulunmaktadır. Algoritmanın özelliklerine bağlı olarak öğrenme oranı giderek azalmaktadır ve bir zaman sonra sistem öğrenmeyi bırakmaktadır. Bu Adagrad'ın en büyük dezavantajıdır. RMSprop bu sorunu çözerek hızlı düşüşü önler. Gradyan düşümü esnasındaki salınımları sönmülemde etkili olmakla beraber daha büyük bir öğrenme oranı (learning rate) η kullanımı sağlar. Bu da öğrenme oranı süresini hızlandırmaktadır. Hinton η öğrenme oranını 0.001 varsayılan değer olarak ayarlamıştır. RMSprop formülleri, Denklem 2.8 ve 2.9'daki gibidir [27,28].

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2 \quad (2.8)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \varepsilon}} g_t \quad (2.9)$$

2.4.2. ADAM

Uyarlanabilir moment tahmini (Adaptive Moment Estimation, ADAM), derin öğrenmede en yaygın kullanılan optimizasyon algoritmalarından biridir. ADAM, temelde momentum ve RmsPROP optimizasyon algoritmalarını baz almaktadır. Denklem 2.10 ve 2.11’ de verilen m_t ve v_t gradyanların sırasıyla birinci momentinin (ortalama) ve ikinci momentinin (merkezlenmemiş varyans) tahminleridir. Özellikle bozulma oranları küçük olduğunda yani β_1 ve β_2 1’e yakinken, moment değerinin sıfıra doğru eğilimli olduğu gözlemlenmiştir. Kingma D.P. ve Ba J.L., β_1 değerini 0.9, β_2 değerini 0.999 ve ε değerini ise 10^{-8} varsayılan olarak ayarlamışlardır [28-29].

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2.10)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2.11)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.12)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.13)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \varepsilon}} \hat{m}_t \quad (2.14)$$

2.5. Hata Matrisi (Confusion Matrix)

Model performansının ölçülmesi için çeşitli metotlar mevcuttur. Sınıflandırma modellerinin performanslarının değerlendirilmesi için genellikle hata matrisi tercih edilmektedir. Gerçek değerlerin tahmin edilen değerler ile karşılaştırılması yapılmaktadır. Tahmin değerleri için dört ayrı çıktı bulunmaktadır [30]. Bu değerler:

- Doğruya doğru demek (True Positive- TP)
- Yanlışla yanlış demek (True Negative – TN)

- Doğruya yanlış demek (False Negative – FN)
- Yanlışla doğru demek (False Positive – FP)

Doğru pozitif (TP), çıktının gerçek değerinin sonucu doğru olup, modelin çıktısının da doğru olarak tahmin edilmesidir. Doğru negatif (TN), gerçekte değer yanlış olduğu ve çıktının doğru olarak tahmin ettiği sınıftır. Yanlış pozitif (FP), çıktının gerçek değeri yanlış olduğu halde çıktının doğru sonucunu verip yanlış tahmin yaptığı sınıftır. Buna tip I hatası da denmektedir. Yanlış negatif (FN), gerçekte çıktının doğru olduğu halde negatif sonucunu verip yanlış tahmin ettiği sınıftır. Buna tip II hatası da denmektedir [30,31]. Bu değerlerin birleşiminden oluşan matris, Şekil 2.14’te gösterilmiştir.

		Var olan Durum	
		Pozitif Durumlar	Negatif Durumlar
Tahmin	Pozitif	TP	FP
	Negatif	FN	TN

Şekil 2.14. Hata Matrisi [30].

Sınıflandırmanın değerlendirmesinde doğruluk (accuracy), geri çağırma (recall), kesinlik (precision), özgüllük (specificity), F1-skoru ve Matthew korelasyon katsayısı (MCC) kullanılan metriklerdir [30,31].

Doğruluk değeri modelde doğru tahmin edilen alanların toplamının toplam veri kümesine oranı ile hesaplanır. Formülü Denklem 2.15’ te verilmiştir [31].

$$\text{Doğruluk(Accuracy)} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.15)$$

Geri çağırma değeri tüm pozitif sınıflardan kaç tanesinin doğru tahmin edildiğini göstermektedir. Diğer bir adı true positive rate (TPR)’tir ve yüksek olması beklenir. Formülü, Denklem 2.16’da verilmiştir [31].

$$\text{Geri çağırma(Recall)} = \frac{TP}{TP+FN} \quad (2.16)$$

Kesinlik pozitif olarak tahmin ettiğimiz tüm sınıflardan kaçının gerçekten pozitif olduğunu göstermektedir. Positive predictive value (PPR) olarak da adlandırılan kesinliğin de oldukça yüksek olması beklenir. Denklem 2.17’de gösterilmiştir [31].

$$\text{Kesinlik(Precision)} = \frac{TP}{TP+FP} \quad (2.17)$$

Özgüllük olumsuz sonuçların ne kadar doğru tahmin edildiğinin ölçüsüdür. Diğer bir adı true negative rate (TNR)’tir ve yüksek olması beklenir. Formülü Denklem 2.18’de verilmiştir [30].

$$\text{Özgüllük(Specificity)} = \frac{TN}{TN+FP} \quad (2.18)$$

Denklem 2.19’da verilen F1 skoru ise geri çağırma ve kesinliğin değerlerinin harmonik ortalaması ile hesaplanır. F1 skoru değerinin kullanılmasındaki en temel sebep eşit dağılmamış veri kümelerinde hatalı model seçimi yapmamaktır. F1 skorunun ulaşabileceği en büyük değer 1’dir [30,31].

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (2.19)$$

Matthew korelasyon katsayısı (MCC) ikili ve çok sınıflı sınıflandırmaların kalitesinin bir ölçüsü olarak kullanılmaktadır. MCC, -1 ile +1 arasında bir korelasyon katsayısı değeridir. +1 katsayısı mükemmel tahmini, 0 ortalama rastgele tahmini ve -1 ters tahmini temsil etmektedir. Formülü Denklem 2.20’de verilmiştir [32].

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP+FP) * (TP+FN) * (TN+FP) * (TN+FN)}} \quad (2.20)$$

2.6. Model Geliřtirmede Kullanılan Kütüphaneler ve Yazılım Dili

Bu tez çalışmasında tercih edilen yazılım dili Python programlama dilidir. Bu dilin tercih edilmesinin sebebi açık kaynak kodlu olması ve yapay zekâ uygulamalarında popüler olmasıdır. Yine açık kaynak olarak sunulan Keras kütüphanesi ve Google tarafından geliştirilmiş olan Keras alt yapısını kullanan Tensorflow kütüphanesi tercih edilmiştir. Ayrıca modellerin eğitimi ve sonuçların kontrolü için destekleyici kütüphaneler olan Numpy, Matplotlib, Scikit-Learn ve PIL kütüphaneleri de kullanılmıştır [33,34].

Numpy kütüphanesi, Python programlama dili için geliştirilmiş, üst düzey matematiksel işlemlerin yapıldığı bir Python kütüphanesidir. Cebirsel işlemler, temel istatistik işlemleri gibi matematiksel işlemler gerçekleştirilebilmektedir [35].

Scikit- Learn, Python programlama dili için geliştirilmiş ücretsiz bir makine öğrenim kitaplığıdır. PIL (Python imaging library) ise Python programlama dili için geliştirilen grafik işleme kütüphanesidir. Açık kaynaklı bir kütüphane olup, görüntü işlemede araçtır. Matplotlib kütüphanesi, Python programlama dili ve Numpy kütüphanesi için bir grafik çizim kütüphanesidir [36-38].

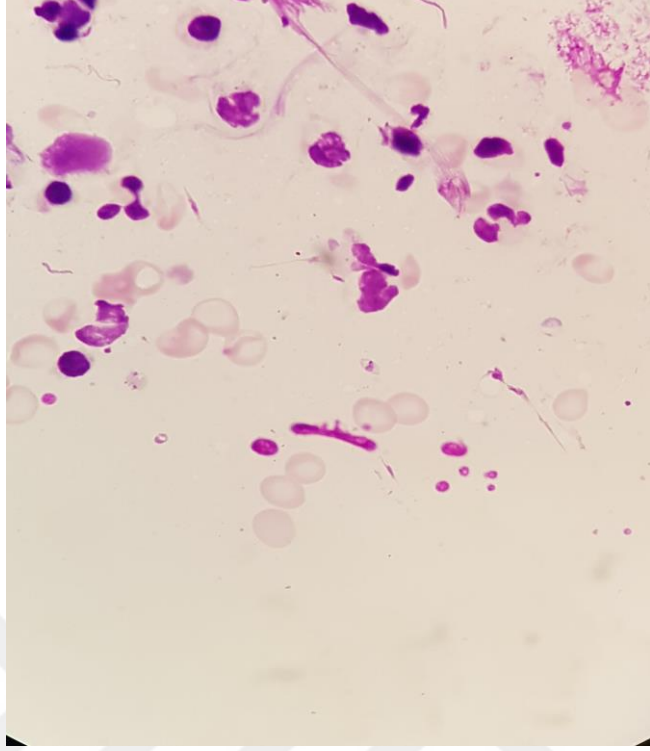
3. BÖLÜM

MODELİN OLUŞTURULMASI ve PERFORMANSININ ÖLÇÜLMESİ

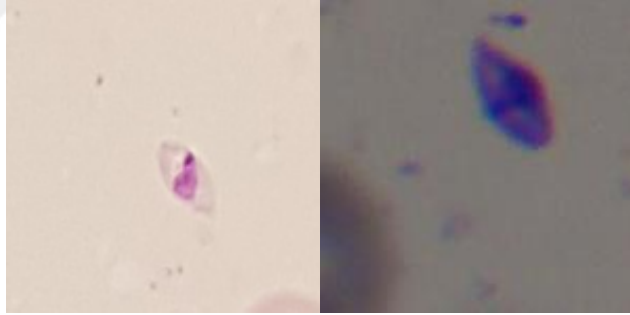
Bu bölümde tez çalışmasında kullanılan derin öğrenme modelinin yapısı, veri setinin hazırlanması, modeli oluşturma esnasında kullanılan hiperparametrelerden bahsedilecektir.

3.1. Veri Seti Hazırlanması

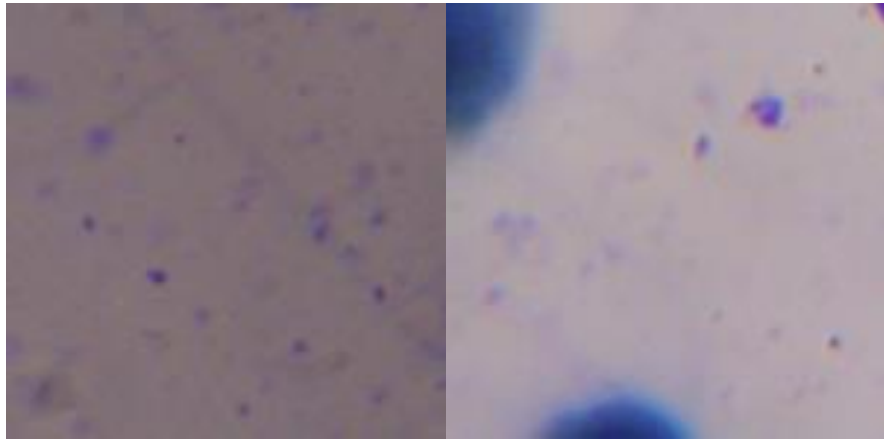
Veri seti, Erciyes Üniversitesi Halil Bayraktar M.Y.O.'da görev yapan Doç. Dr. Ülfet Çetinkaya'dan ve açık kaynak olarak verilen internet sitesinden elde edilmiştir [5]. Görüntülerin hazırlanmasında PIL kütüphanesinden faydalanılmıştır. 50 adet RGB görüntüden oluşan veri seti, toplamda 115 parazit içeren, 115 parazit içermeyen olmak üzere 230 görüntü haline getirilmiştir. 150 x 150 piksel boyutuna getirilen görüntüler 184 adet eğitim (Train) ve model performansının test edilmesi için 46 adet test verisine rastgele olarak bölünmüştür. Görüntülerin rastgele olarak ayrılmasında Scikit-learn kütüphanesinden faydalanılmıştır. Veri setinden örnek görüntü, Şekil 3.1' de ve 150 x 150 boyutuna getirilen görseller, Şekil 3.2 ve Şekil 3.3' de gösterilmiştir.



Şekil 3.1. Veri Setinden Örnek Görüntü.



Şekil 3.2. Veri setinde parazit içeren görüntüler.



Şekil 3.3. Veri setinde parazit içermeyen görüntüler.

3.2. Derin Öğrenme Modelinin Oluşturulması

Keras kütüphanesi ile Tensorflow alt yapısı kullanılarak 5 farklı derin öğrenme modeli oluşturulmuştur. Bu modeller oluşturulurken farklı hiperparametre değerleri ayarlanmıştır. Evrişimli katmanların sayıları, filtre sayıları, gizli katmanlardaki nöron sayıları ve toplam eğitilebilir parametre sayıları her model için ayrı ayrı belirlenmiştir. Her bir evrişim katmanından sonra kullanılan MaxPool katmanında ise filtre boyutu 2x2 olarak belirlenmiştir.

Tam bağlı katmanlarda uygulanan düğüm seyreltme işlemi, her bir model için 2 ayrı değer olarak belirlenmiş olup, her modelde dropout değerleri 0 ve 0.5 olarak atanmıştır. Evrişim katmanlarından ve tam bağlı katmanlardan sonra aktivasyon fonksiyonu olarak ReLU tercih edilmiştir. Sadece çıkış katmanında sigmoid fonksiyonu kullanılmıştır. Bunun sebebi ise sınıflandırma sonucunda 1 ve 0 değerlerini elde edilmesinin istenmesidir. Tablo 3.1’ de oluşturulan CNN tabanlı 5 farklı derin öğrenme modelinin hiperparametre değerleri gösterilmiştir.

Tablo 3.1. CNN tabanlı oluşturulan derin öğrenme modellerinin hiperparametre değerleri.

	Evrişimli katman sayısı	Her bir Evrişimli katmanın filtre sayısı	Dense katman sayısı	Her bir dense katmandaki nöron sayısı	Toplam parametre sayısı
Model 1	3	32-32-64	2	64-1	1.212.513
Model 2	4	32-32-64-64	2	64-1	266.401
Model 3	4	32-32-64-64	3	64-32-1	268.449
Model 4	3	32-32-64	3	32-32-1	621.633
Model 5	4	32-32-64-128	3	64-32-1	506.081

3.3. Derin Öğrenme Modellerinin Eğitilmesi

Jupyter Notebook, kod, denklemler, görselleştirmeler ve metin içeren belgeler oluşturmak ve paylaşmak için kullanılan, veri bilimini ve bilimsel hesaplamaları desteklemek için geliştirilen, 2014 yılında ortaya çıkan açık kaynaklı bir web uygulamasıdır [39].

Jupyter Notebook ortamında oluşturulan evrişimli sinir ağı tabanlı modeller GPU desteğiyle eğitilmiştir. Eğitimler sırasında batch boyutu, kayıp fonksiyonu, optimize edici gibi farklı hiperparametreler kullanılmıştır. Batch boyutu 16, kayıp fonksiyonu binary_crossentropy olarak tercih edilmiştir. Optimize edici olarak iki farklı seçenek tercih edilip birincisi ADAM ve ikincisi RMSprop olarak kullanılmıştır. Eğitimler NVIDIA GeForce GTX 950M desteği ile 25 turda (epoch) tamamlanmıştır.

3.4. Modellerin performansları

Eğitimleri tamamlanan modellerin hata matrisleri elde edilmiş ve bu değerlerden modellerin doğruluk, geri çağırma, kesinlik, özgüllük, F1 skoru ve Matthew korelasyon katsayıları hesaplanmıştır. Tablo 3.2’de görüleceği üzere evrişimli sinir ağı tabanlı modeller arasında en iyi sonucu optimize edici olarak ADAM, düğüm seyreltme olarak 0.5 hiperparametre değerleri ile Model 5 göstermiştir. Doğruluk, geri çağırma, kesinlik, özgüllük, F1 skoru ve Matthew korelasyon katsayısı değerine baktığımızda sırasıyla 0.9130, 0.85, 0.9444, 0.9615, 0.8947 ve 0.8243 değerleri elde edilmiştir.

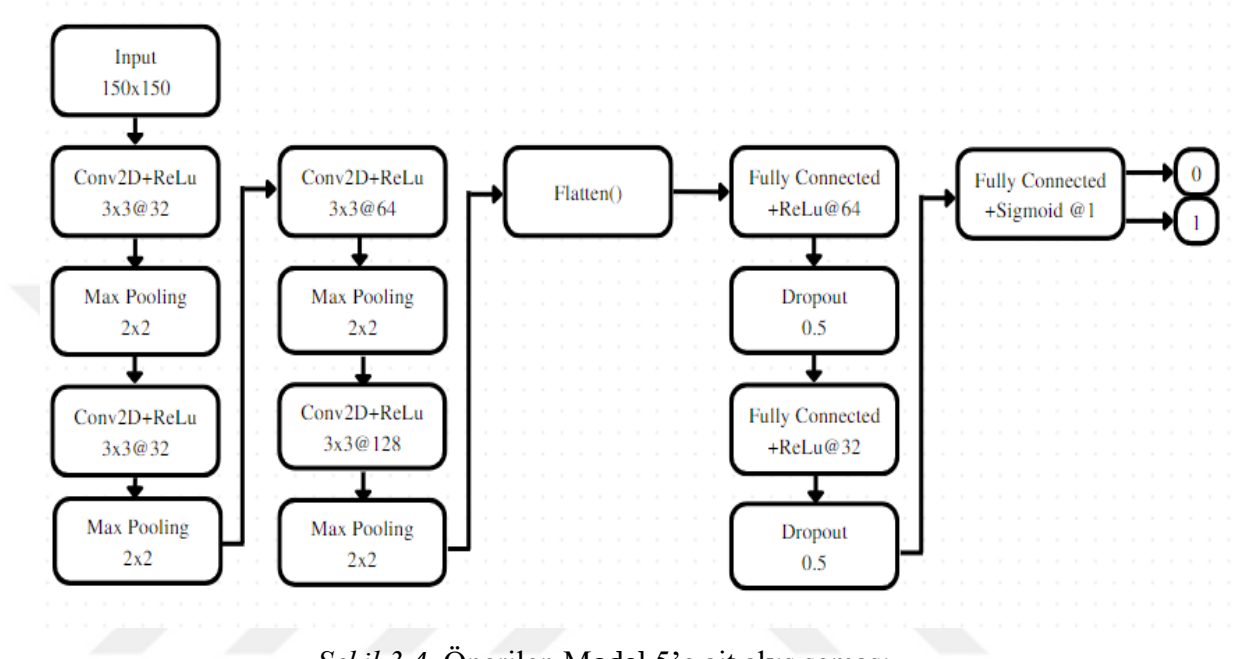
3.5. Önerilen Model Sonuçları

Derin öğrenme modellerinin eğitimlerinin sonucunda Tablo 3.2’de görülen performanslar kıyaslandığında en iyi sonuçları veren derin öğrenme modeli, Model 5 olarak görülmektedir. Önerilen modelde toplam 506.081 eğitilebilir parametre yani başka bir deyişle ağırlık (W) mevcuttur. Bu modele ait akış şeması, Şekil 3.4’de verilmiştir.

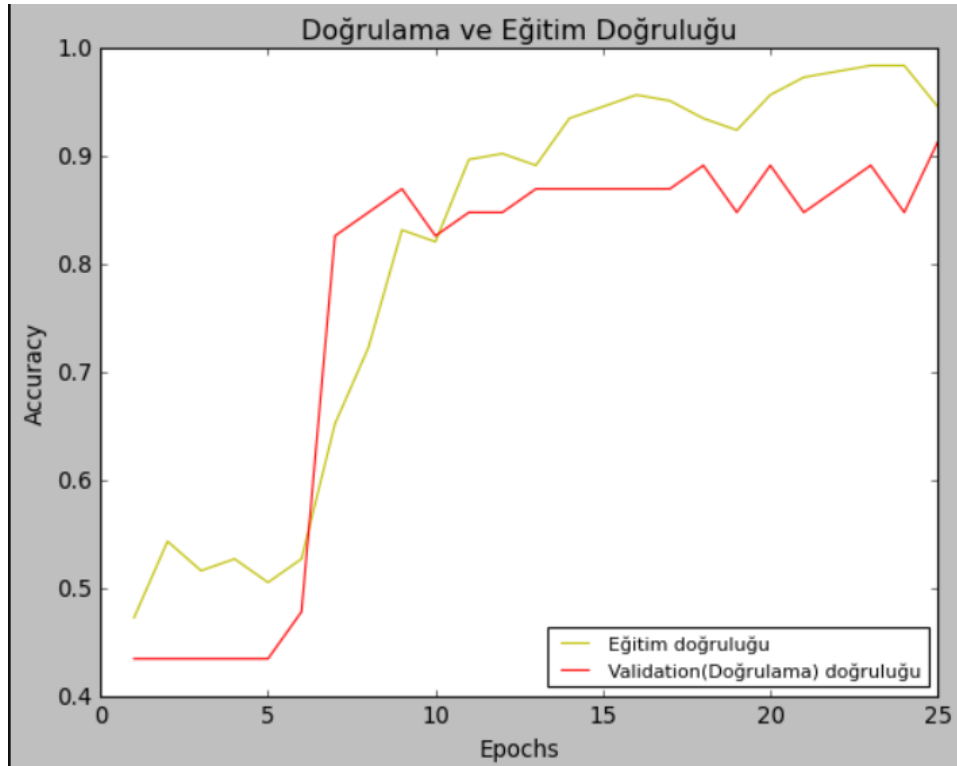
Tablo 3.2. Farklı hiperparametre değerlerine göre oluşturulan modellerin eğitimlerinin sonucunda elde edilen hata matrisi ile hesaplanan performans değerleri.

	Optimize edici	Dropout	Doğruluk (Accuracy)	Geri Çağırma (Recall) TPR	Kesinlik (Precision) PPV	Özgüllük (Specificity) TNR	F1 Skoru	MCC
Model-1	ADAM	0	0.7174	0.45	0.8181	0.9230	0.5806	0.4335
Model-2	ADAM	0	0.8260	0.65	0.9285	0.9615	0.7647	0.6588
Model-3	ADAM	0	0.8260	0.65	0.9285	0.9615	0.7647	0.6588
Model-4	ADAM	0	0.7608	0.50	0.9091	0.9615	0.6451	0.5363
Model-5	ADAM	0	0.7826	0.55	0.9166	0.9615	0.6875	0.5775
Model-1	ADAM	0,5	0.7608	0.55	0.8461	0.9230	0.667	0.5208
Model-2	ADAM	0,5	0.8261	0.70	0.875	0.9230	0.777	0.6485
Model-3	ADAM	0,5	0.8261	0.70	0.875	0.9230	0.7777	0.6485
Model-4	ADAM	0,5	0.8695	0.75	0.9375	0.9615	0.8334	0.7405
Model-5	ADAM	0,5	0.9130	0.85	0.9444	0.9615	0.8947	0.8243
Model-1	RMSProp	0	0.7826	0.55	0.9166	0.9615	0.6875	0.5774
Model-2	RMSProp	0	0.8478	0.65	1.0	1.0	0.7878	0.7156
Model-3	RMSProp	0	0.8043	0.65	0.8667	0.9230	0.7428	0.6060
Model-4	RMSProp	0	0.8043	0.55	1.0	1.0	0.7096	0.6392
Model-5	RMSProp	0	0.8696	0.75	0.9375	0.9615	0.8333	0.7405
Model-1	RMSProp	0,5	0.7391	0.45	0.90	0.9615	0.60	0.4946
Model-2	RMSProp	0,5	0.804	0.55	1.0	1.0	0.7096	0.6392
Model-3	RMSProp	0,5	0.8043	0.60	0.9231	0.9615	0.7272	0.6182
Model-4	RMSProp	0,5	0.7826	0.55	0.9166	0.9615	0.6875	0.5775
Model-5	RMSProp	0,5	0.8261	0.70	0.8750	0.9230	0.7777	0.6485

Model 5'e ait eğitim başarımları ve kayıp grafikleri Şekil 3.5 ve Şekil 3.6'da gösterilmektedir. Şekil 3.5'te eğitim ve test başarımları verilmiştir. Grafikte sarı ile belirtilen eğri, eğitim başarımlarını (training accuracy), kırmızı renk ile gösterilen eğri ise test başarımlarını (val accuracy) göstermektedir.

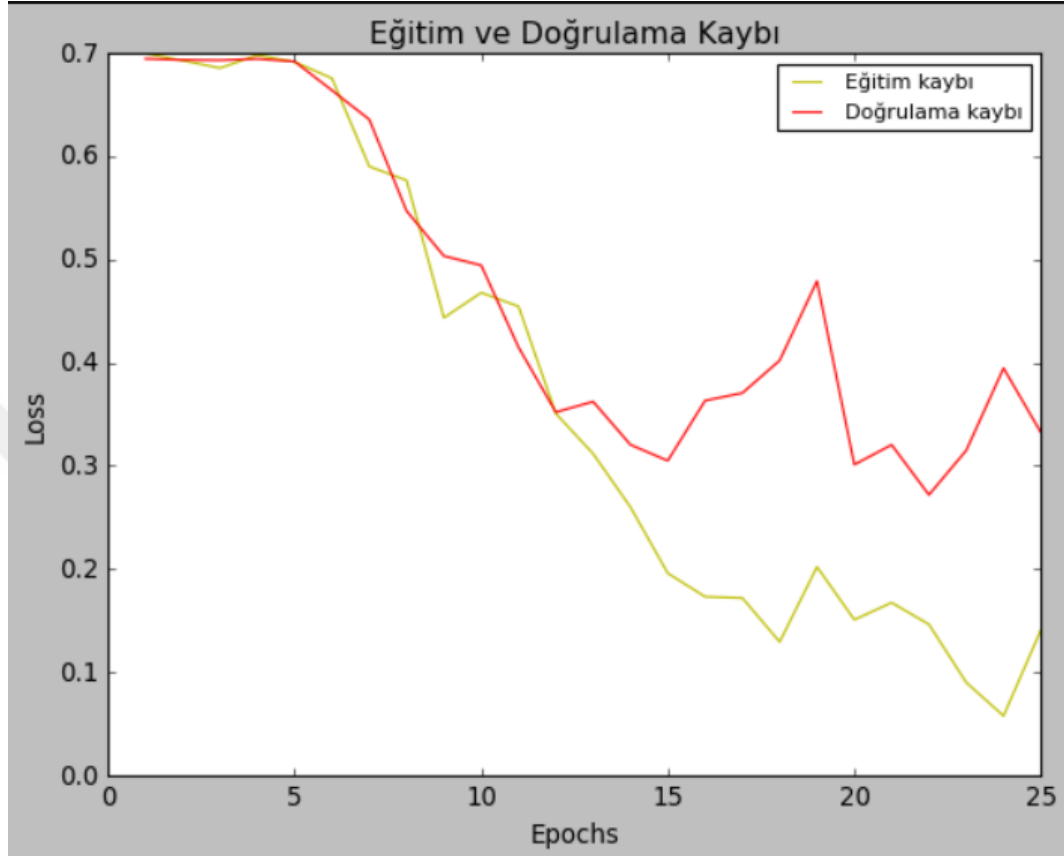


Şekil 3.4. Önerilen Model 5'e ait akış şeması.



Şekil 3.5. Model 5'e ait eğitim doğruluk grafiği.

Şekil 3.6’da sarı ile belirtilen eğri, eğitim kayıplarını (training loss), kırmızı renk ile gösterilen eğri ise test kayıplarını (val loss) göstermektedir.



Şekil 3.6. Model 5’e ait eğitim kayıp grafiği.

Model 5’e ait hata matrisi de Şekil 3.7’de gösterilmektedir. Hata matrisindeki değerler incelendiğinde 25 TN, 17 TP, 3 FP ve 1 FN olmak üzere toplam 42 doğru tahmin ve 4 hatalı tahmin yapıldığı görülmektedir.

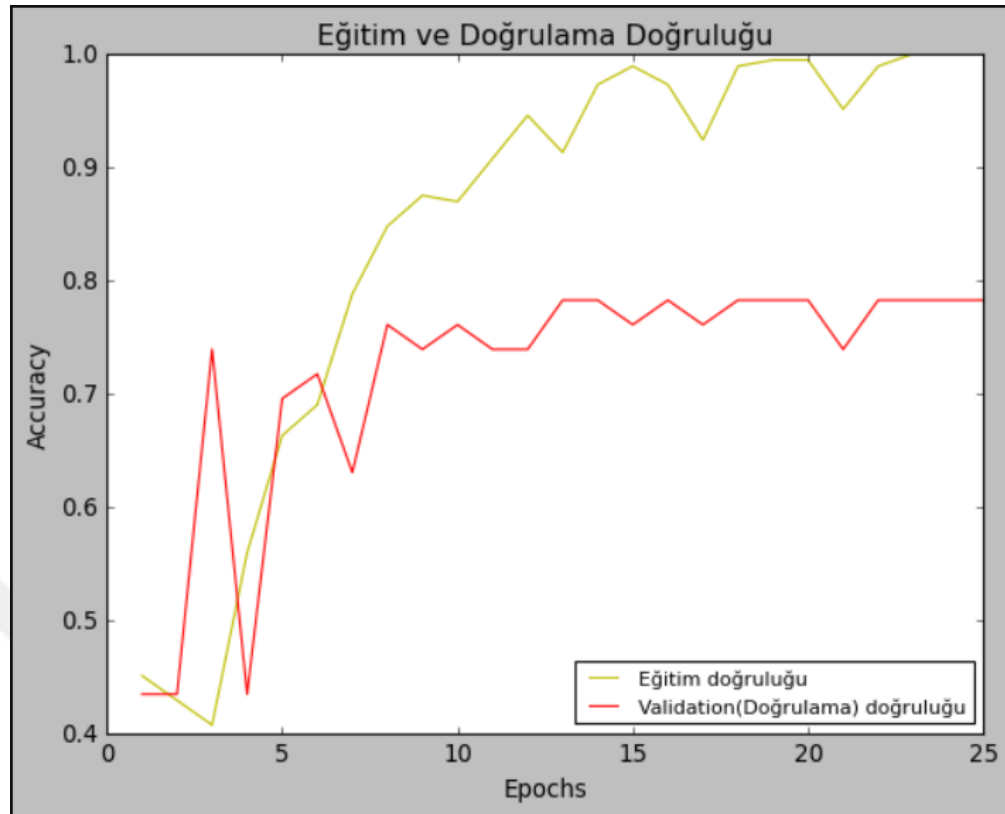
MCC değerine ve diğer performans ölçüm değerlerini incelediğimizde önerilen modelin performansının diğer modellere göre farkı görülmektedir. Model eğitim doğruluk oranı yüzde olarak % 91.30 ve 0.8243 MCC değerinin +1’e yakın olması tahmin edilen değerlerin doğruluğunun yüksek olduğunu göstermektedir.

Diğer modellerin sonuçlarına bakacak olursak Model 1’in 4 ayrı eğitim sonucu ayrı ayrı incelendiğinde en iyi performansını RMSProp optimize edici ve 0 dropout’ta sergilemiştir. Model 1’e ait doğruluk ve kayıp grafikleri Şekil 3.8 ve Şekil 3.9’da gösterilmiştir. Model 2’de ise eğitim sonuçları arasında en başarılısı RMSProp optimize edici ve 0 dropout’ta

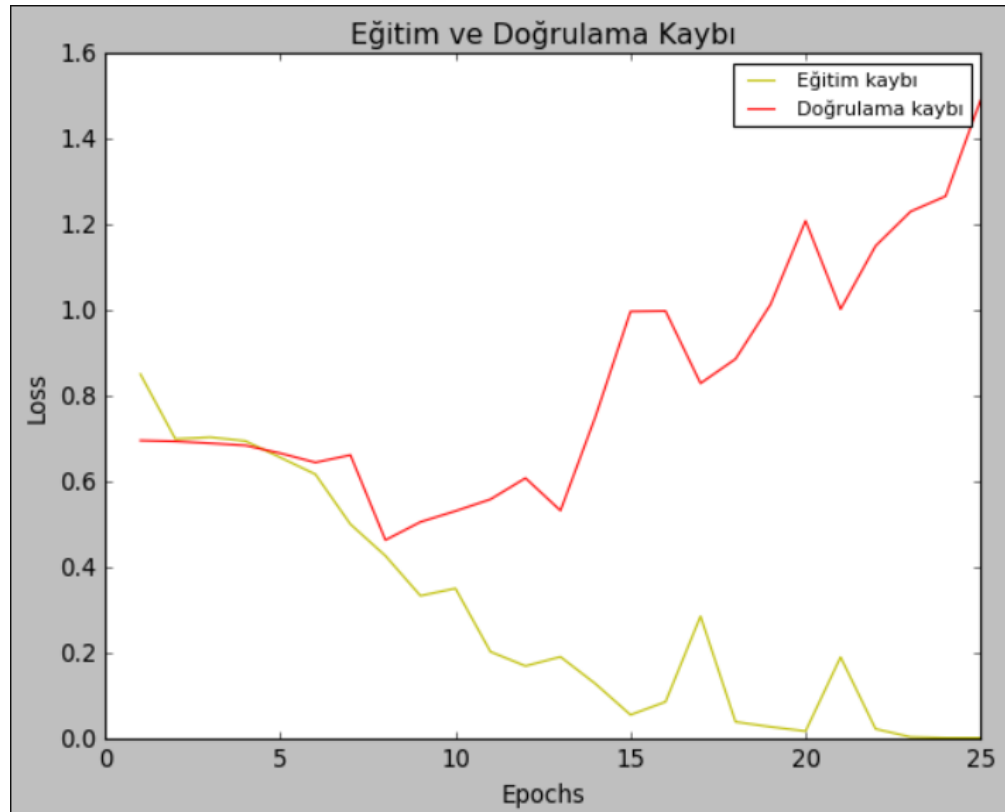
sergilenmiştir. Model 2'nin doğruluk ve kayıp grafikleri de Şekil 3.10 ve Şekil 3.11' de gösterilmiştir. Model 1 ve 2'nin hata matrisleri ise şekil 3.12 ve şekil 3.13'te verilmiştir. 4 ayrı eğitimin ardından, Model 3'lerin arasında en iyi performans gösteren sonucu Adam optimize edici ve 0.5 dropouttur. Model 3'ün doğruluk ve kayıp grafikleri Şekil 3.14 ve 3.15'te yer almaktadır. Son olarak Model 4 incelendiğinde Model 4'ler arasında en iyi performansı Adam optimize edici ve 0.5 dropout'ta göstermiştir. Model 4'e ait grafikler Şekil 3.16 ve Şekil 3.17'te gösterilmiştir. Model 3 ve 4'ün hata matrisleri şekil 3.18 ve şekil 3.19'da verilmiştir.

		Hata Matrisi	
		0	1
Gerçek Değerler	0	25 TN	1 FN
	1	3 FP	17 TP
		Tahmin Edilen Değerler	

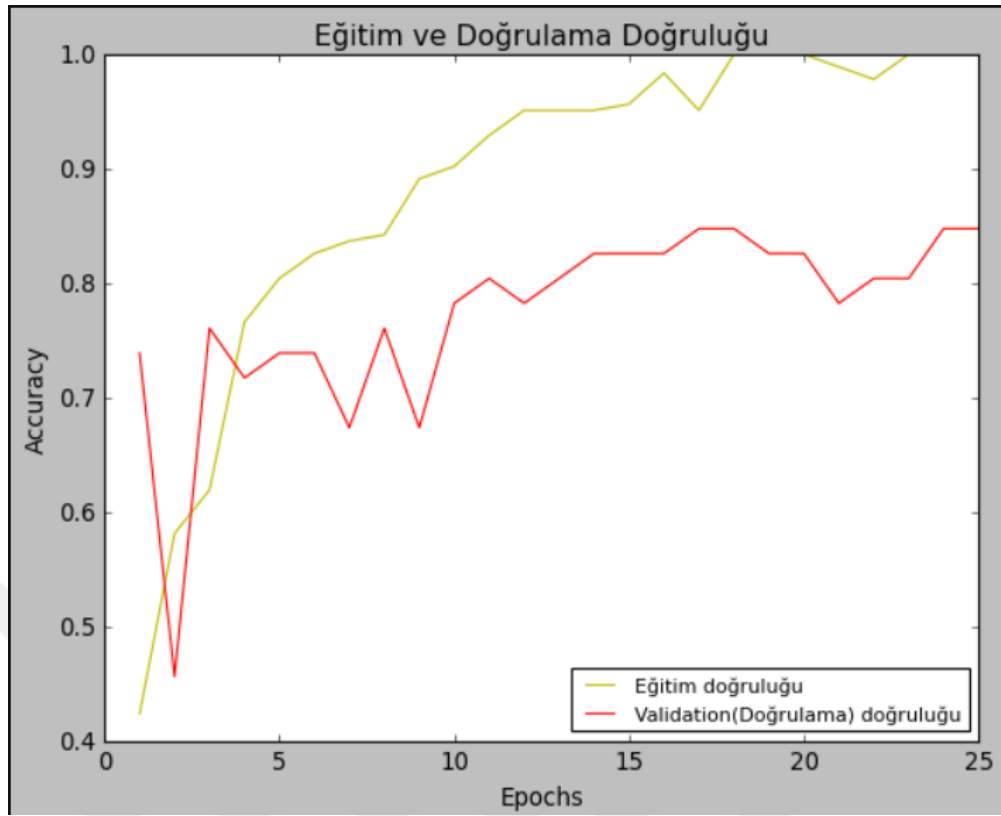
Şekil 3.7. Model 5'e ait hata matrisi.



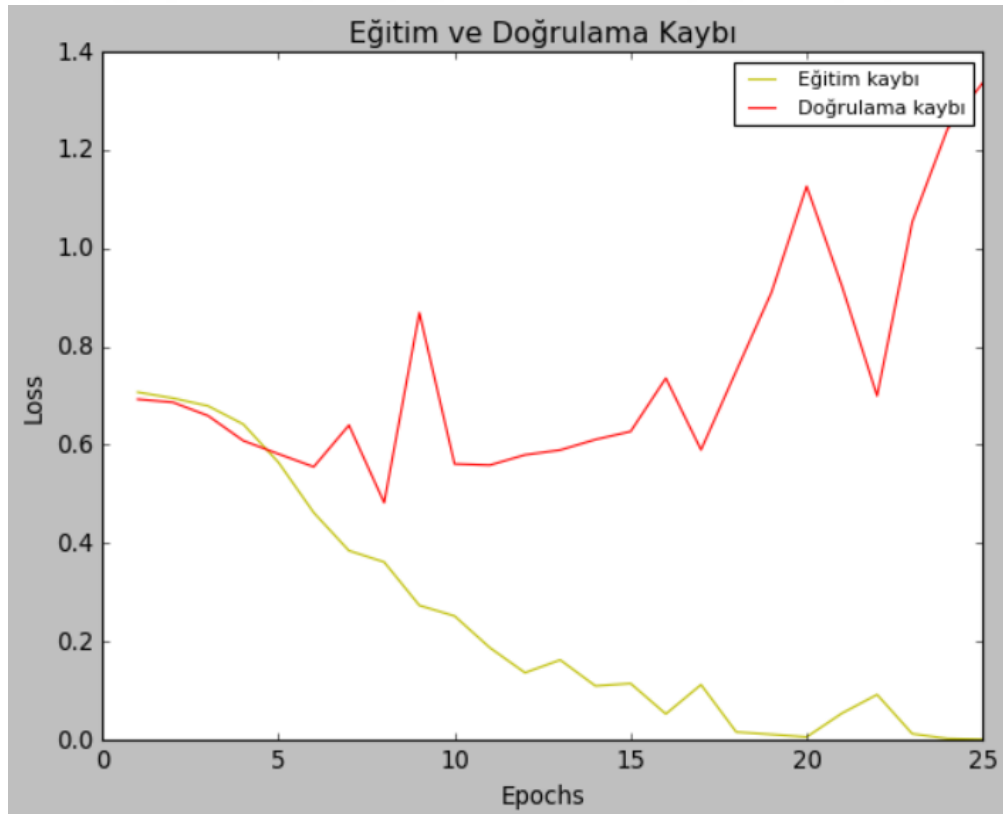
Şekil 3.8. Model 1'e ait eğitim doğruluk grafiği.



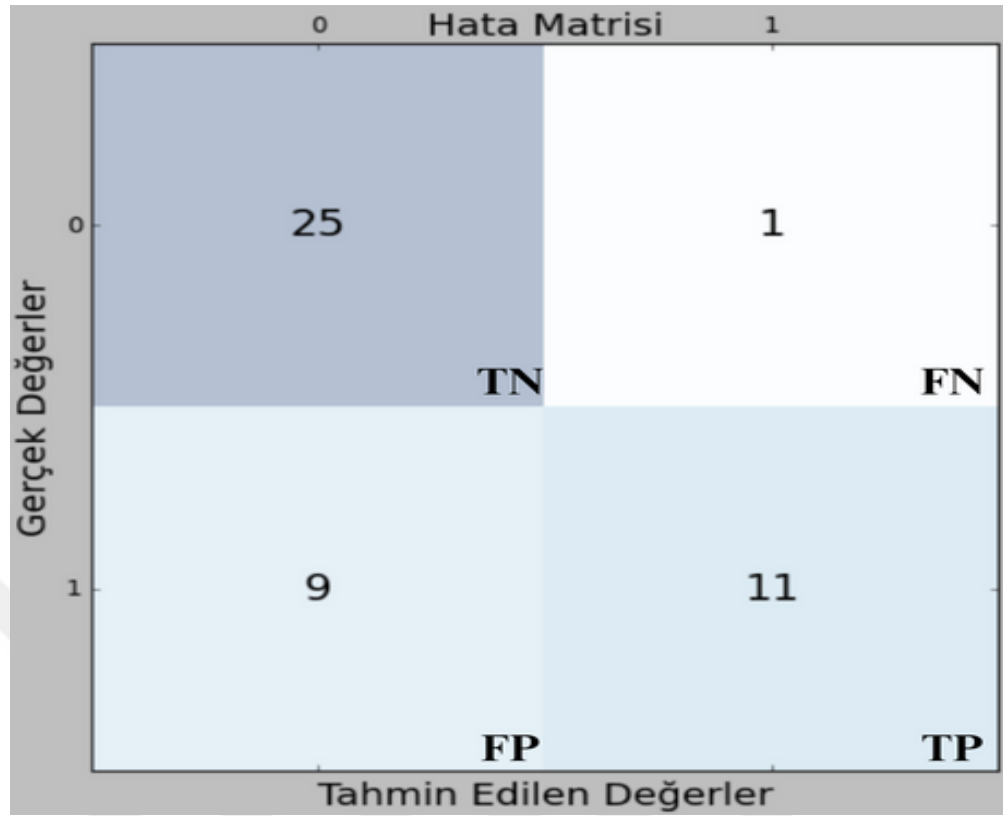
Şekil 3.9. Model 1'e ait eğitim kayıp grafiği.



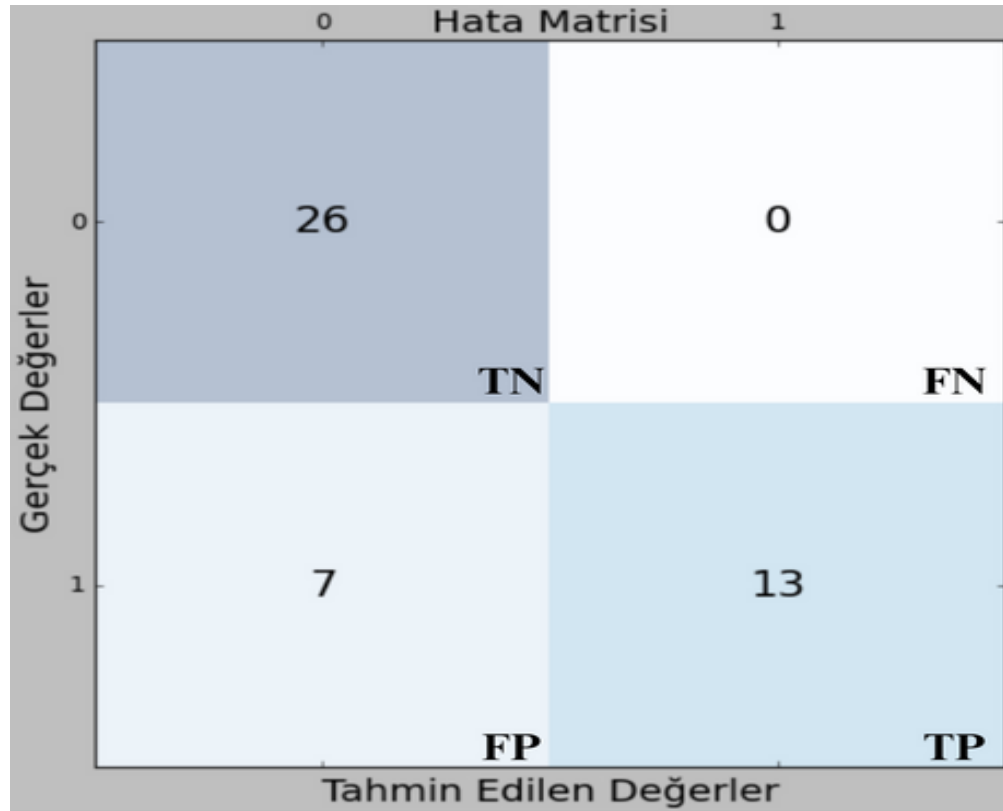
Şekil 3.10. Model 2'ye ait eğitim doğruluk grafiği.



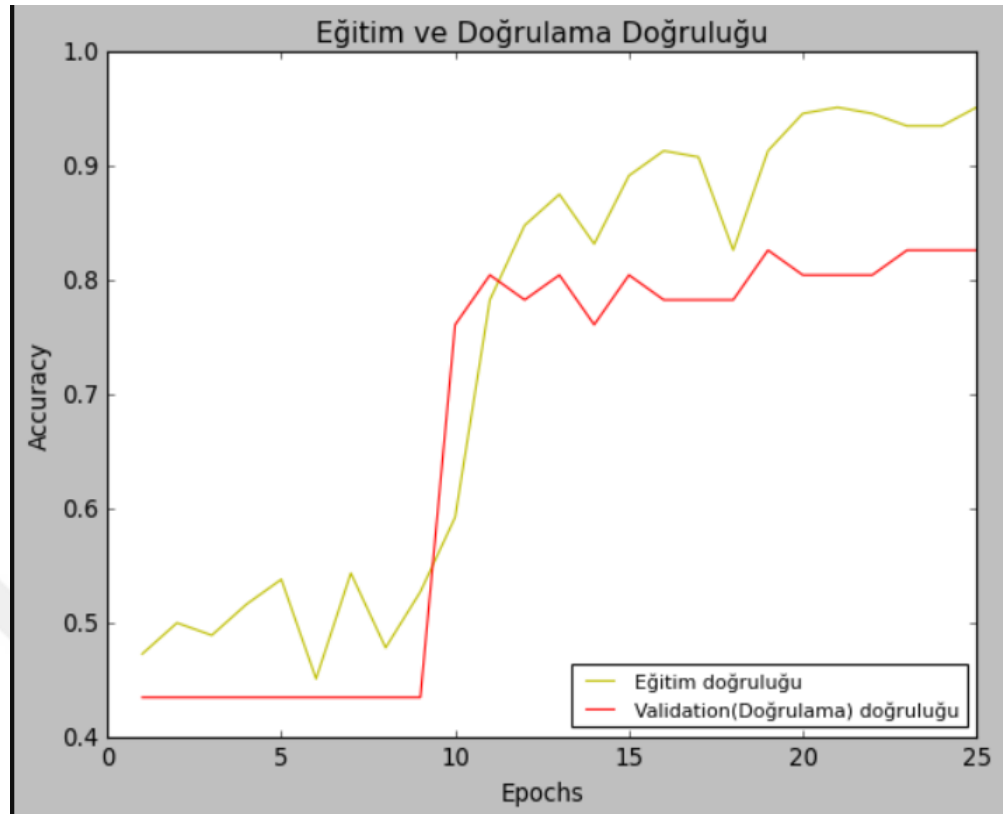
Şekil 3.11. Model 2'ye ait eğitim kayıp grafiği.



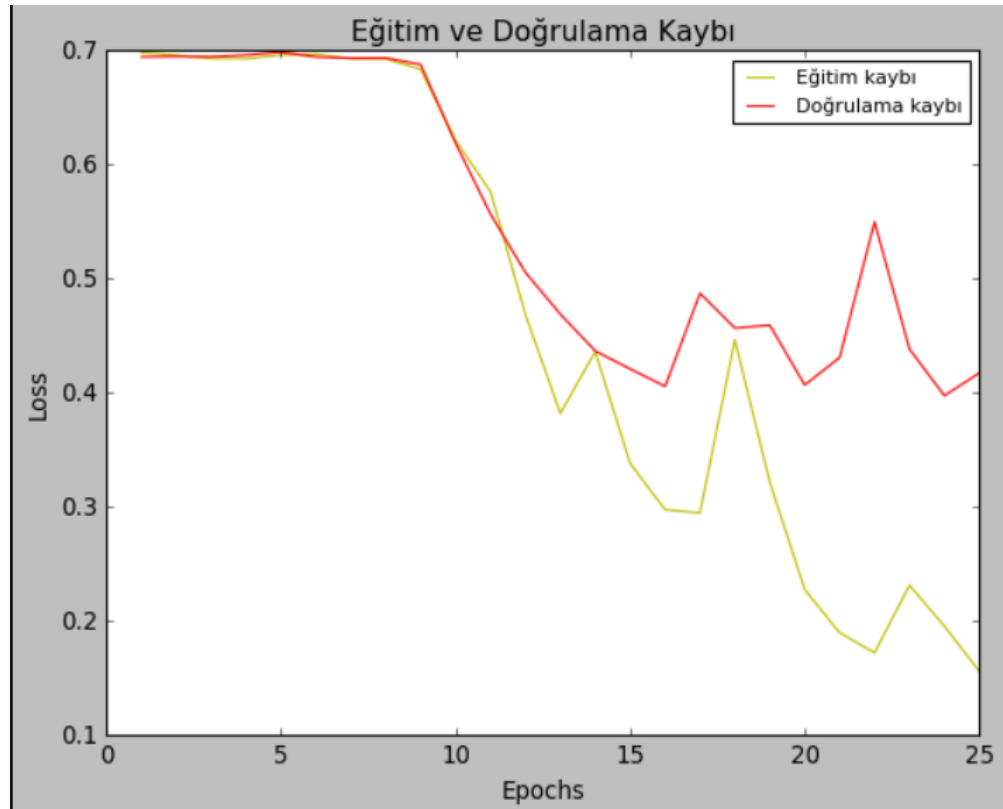
Şekil 3.12. Model 1'e ait hata matrisi.



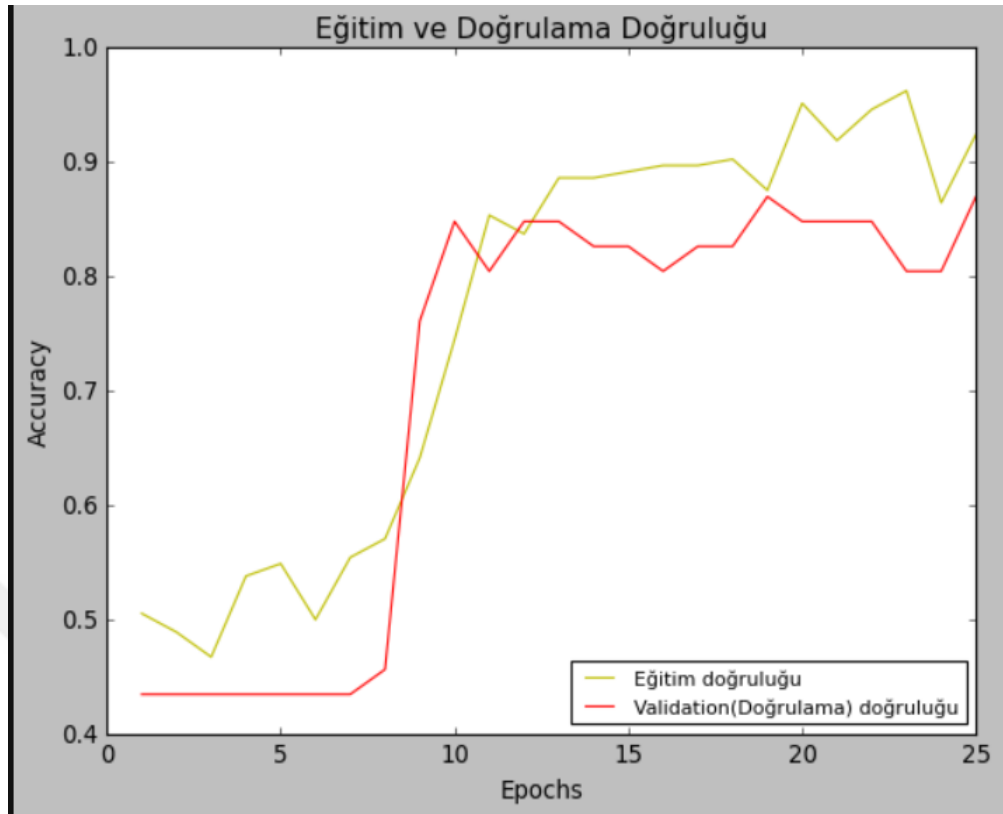
Şekil 3.13. Model 2'ye ait hata matrisi.



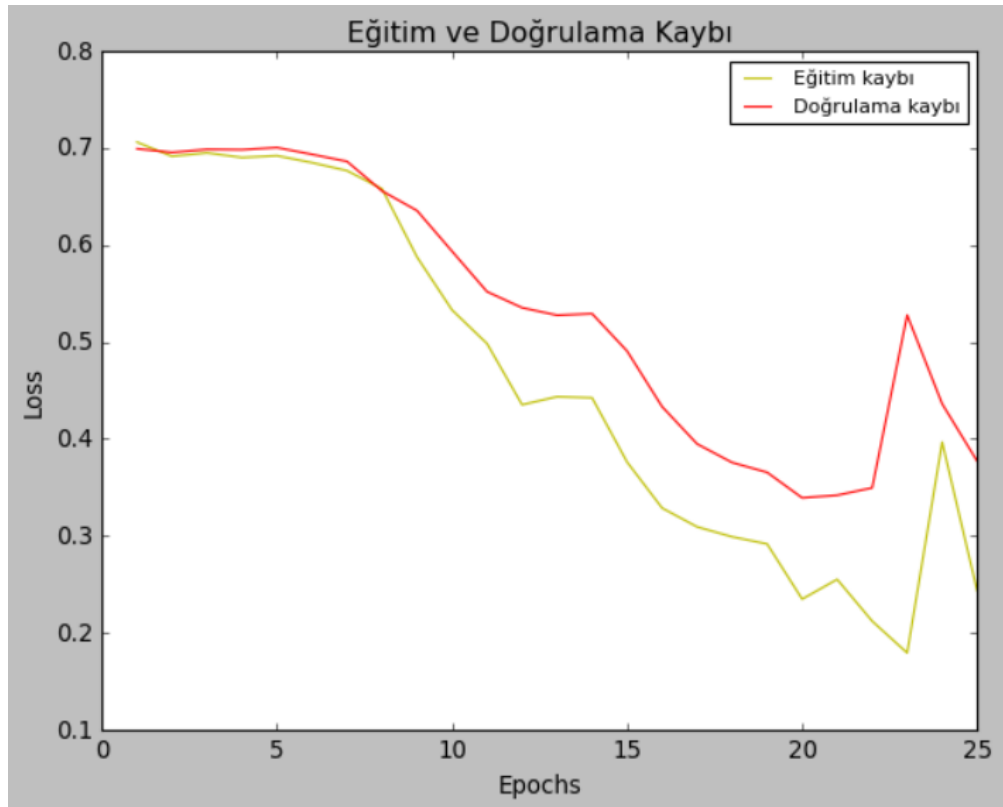
Şekil 3.14. Model 3'e ait eğitim doğruluk grafiği.



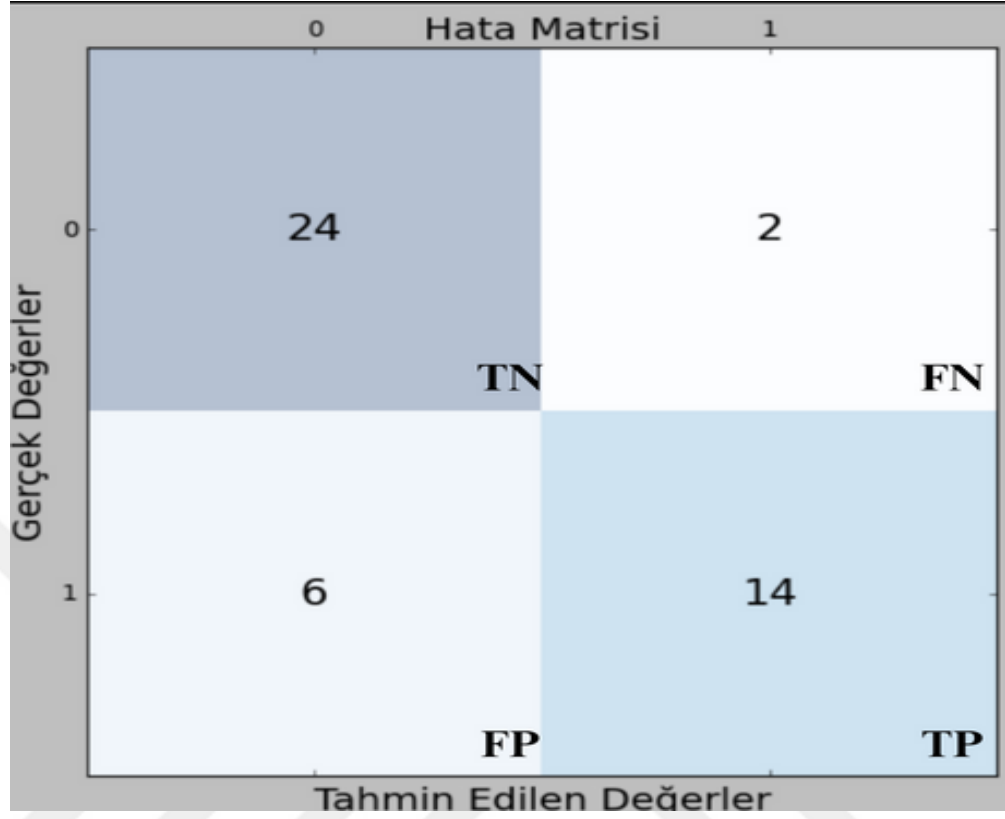
Şekil 3.15. Model 3'e ait eğitim kayıp grafiği.



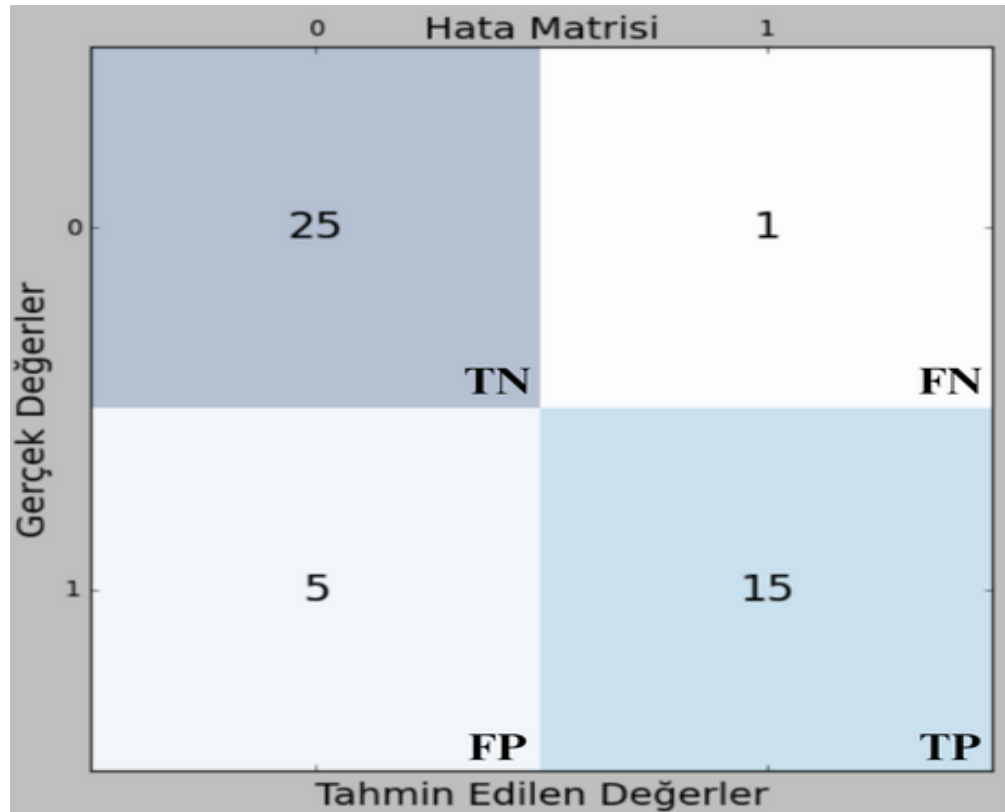
Şekil 3.16. Model 4'e ait eğitim doğruluk grafiği



Şekil 3.17. Model 4'e ait eğitim kayıp grafiği



Şekil 3.18. Model 3'e ait hata matrisi.



Şekil 3.19. Model 4'e ait hata matrisi.

4. BÖLÜM

TARTIŞMA, SONUÇ ve ÖNERİLER

4.1.Tartışma

Bu çalışmada öncelikle literatüre girmiş derin öğrenme ve makine öğrenmesi yöntemleri ile parazit tespiti ve sınıflandırması çalışmaları incelenmiş, kullanılan yöntemler, modeller ve bu modellerin sonuçlarının nasıl değerlendirildiği gözlemlenmiştir. Kullanılan modeller ışığında literatür taraması yapılmış ve CNN tabanlı modelin bu çalışmada kullanılmasına karar verilmiştir. Modeller Python yazılım dili ile Jupyter Notebook ortamında oluşturulmuştur.

Daha sonra toplanan mikroskop görüntülerinden veri seti oluşturulmuş ve modeller ayrı ayrı eğitime tabi tutulmuştur. Her bir model için ayrı ayrı hiperparametre değerleri belirlenmiş, eğitim esnasında ayrı optimizasyon algoritmaları kullanılmıştır.

Modellerin grafik değerleri, hata matrisleri ve Tablo 3.2 incelendiğinde Model 5'in diğer modellere göre daha başarılı sonuç verdiği görülmüştür. Literatüre baktığımızda yapılan parazit tespit ve sınıflandırma çalışmalarının sonuçlarına yakın ve hatta daha iyi sonuçlar gözlemlenmiştir.

Elimizde bulunan veri seti ile elde edilen sonuçlar tatmin edici seviyede olsa da veri seti miktarının artırılması, oluşturulan modeller için daha da iyi performans gözlemlenmesi anlamına gelir. Derin öğrenme modelleri eğilirken kullanılan veri seti miktarı ne kadar fazla ise eğitilen model o kadar fazla başarı sergilemektedir. Eğitilen model, klinik çalışmalarda ve laboratuvarda kullanılabilecek olup, böylece erken teşhis yapıp, tedavi süreci kısalabilecektir.

4.2.Sonuç ve Öneriler

CNN tabanlı oluşturulan 5 modelin toplam 230 görüntü ile 25 turda eğitimleri tamamlanmıştır. Eğitim desteği olarak NVIDIA GeForce GTX 950M kullanılmış olup, eğitim süresinin kısaltılmasına yardımcı olmuştur. Eğitim sonuçlarına bakıldığında ADAM optimize edicisinin RMSProp optimize edicisine oranla daha iyi sonuçlar verdiği görülmektedir. Ayrıca dropout kullanılan modellerde aşırı öğrenmenin önüne geçilerek, modellerin ezber yapmaması sağlanmış ve başarı oranının yükseldiği gözlemlenmiştir. ADAM optimize edici ile birlikte 0.5 dropout kullanılan Model 5'in sonuçlarına bakıldığında diğer modellerin başarı seviyesine göre değerleri daha iyi seviyededir.

Optimizasyonda doğruluk, kesinlik, geri çağırma ve özgüllük değerlerinin yüksek olması beklenir. Ayrıca F1 skoru değerinin de yüksek olması beklenir ve ulaşacağı en yüksek değer 1'dir. MCC değerinin +1'e yakın değerde olması, sınıflandırma modellerinde başarının yüksek olduğu anlamına gelir. Tablo 3.2 incelendiğinde Model 5'in doğruluk, geri çağırma, kesinlik, özgüllük, F1 skoru ve Matthew korelasyon katsayısı değeri sırasıyla % 91.30, % 85, % 94.44, % 96.15, % 89.47 ve %82.43 olarak görülmektedir. Kesinliğin %94.44 olması demek modelin parazitli bir görüntü algıladığında %94 oranında doğru olduğunu gösterir. Geri çağırma oranı % 85 olup bu oran ise modelin gerçekten parazitli görüntülerin %85'ini doğru bir şekilde tespit ettiğini gösterir.

Gelişen teknoloji ile birlikte tıbbi görüntülerin analizi ve sınıflandırılması için derin öğrenme yöntemleri yaygın olarak kullanılmaktadır. Dolayısıyla bu tez çalışmasında önerilen model kullanılarak, leishmania parazitinin teşhisinde daha az uzmanlık bilgisi, teşhisin daha hızlı yapılabilmesi ve daha etkili sonuçların elde edilmesi beklenmektedir.

Veri setinde bulunan görüntülerin artırılması ile oluşturulan modellerin performanslarının daha iyi sonuçlar vereceği ve bu çalışmanın ileride yapılacak olan sınıflandırma ve tespit çalışmalarına kaynak niteliğinde olması beklenmektedir.

KAYNAKÇA

1. T.C. Sağlık Bakanlığı, 2023. Leishmaniasis. (Web sayfası: <https://www.seyahatsagligi.gov.tr/site/HastalikDetay/Leishmaniasis>), (Erişim tarihi: Mayıs 2023).
2. Wikipedia, 2023. Giemsa boyası. (Web sayfası: https://tr.wikipedia.org/wiki/Giemsa_boyas%C4%B1), (Erişim tarihi: Mayıs 2023).
3. Harman, M., 2015. Kutanöz leishmaniasis. **Turk J Dermatol**, **9**(4):168-176.
4. Medikalpark, 2023. PCR nedir?. (Web sayfası: <https://www.medicalpark.com.tr/pcr-testi-nedir/hg-2393>), (Erişim Tarihi: Mayıs 2023)
5. M. Farahi, H. Rabbani, A. Mehri, 2014. Automatic boundary extraction of leishman bodies in bone marrow samples from patients with visceral leishmaniasis, **Journal of Isfahan Medical School**, **32**(286):726-739
6. Zhang, C. et al., 2022. Deep learning for microscopic examination of protozoan parasites. **Computational and Structural Biotechnology Journal**, **20**: 1036–1043.
7. WHO, 2022. Leishmaniasis. (Web sayfası: <https://www.who.int/news-room/factsheets/detail/leishmaniasis>), (Erişim tarihi: Mayıs 2023)
8. Direkel, Ş., 2016. Hidrazon yapısındaki on adet bileşiğin antileishmanial aktivitesinin araştırılması (Web sayfası: <https://docplayer.biz.tr/13280492-Hidrazon-yapisindaki-on-adet-bilesigin-antileishmanial-aktivitesinin-arastirilmesi.html>), (Erişim Tarihi: Haziran 2023).
9. Jacques, S.M.L., 2017. Image Analysis and Classification Techniques for Leishmaniasis Detection. Polytechnic University of Catalonia, Degree Thesis, Barcelona, 44.
10. Zare, M., Akbarialiabad, H., Parsaei, H. et al., 2022. A machine learning-based system for detecting leishmaniasis in microscopic images. **BMC Infect Diseases** **22**(1):48.
11. Qanbar, M.M., 2019. Derin Öğrenme Yaklaşımı ile Malaria (Sıtma) Hastalığı Görüntülerinin Sınıflandırılması. Selçuk Üniversitesi, Yüksek Lisans Tezi, Konya, 96.

12. Razin, R.W.M., Gunawan, T.S., Kardiwi, M., Yusoff, N.M., 2022. Malaria parasite detection and classification using CNN and YOLOv5 architectures. pp. 277-281 *IEEE 8th International Conference on Smart Instrumentation, Measurement and Applications*, 26-28 September 2022, Meleka, Malaysia.
13. Butploy, N., Kanarkard, W., & Maleewong Intapan, P., 2021. Deep learning approach for *Ascaris lumbricoides* parasite egg classification. **Journal of parasitology research**, 2021(1):6648038.
14. Çakır Ö., 2019. Derin öğrenme nedir?. (Web sayfası: <https://www.yapayzekatr.com/2019/12/16/derin-ogrenme-nedir/>), (Erişim tarihi: Mayıs 2023).
15. Oracle., 2022. Derin öğrenme nedir?. (Web sayfası: <https://www.oracle.com/tr/artificial-intelligence/machine-learning/what-is-deep-learning/>), (Erişim tarihi: Mayıs 2023).
16. Akca M.F., 2020. RNN nedir? Nasıl çalışır?. (Web sayfası: <https://medium.com/deep-learning-turkiye/rnn-nedirnas%C4%B1%C3%A7al%C4%B1%C5%9F%C4%B1r9e5d572689e1>), (Erişim tarihi: Mayıs 2023).
17. Cho K., Merrienboer B., Schwenk H., Bougares F., Bengio Y., Bahdanau D., 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation, **ArXiv:1406.1078v3**.
18. Amidi S., Amidi A. (Web sayfası: <https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-recurrent-neural-networks#architecture>), (Erişim tarihi: Mayıs 2023).
19. Hochreiter, S. Schmidhuber, J., 1997. Long short-term memory, **Neural computation**, 9 (8): 1735-1780.
20. Olah, C., 2015. Understanding LSTM networks. (Web sayfası: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>), (Erişim tarihi: Mayıs 2023).
21. Şeker A., Diri B., Balık H.H., 2017. Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. **Gazi Mühendislik Bilimleri Dergisi**, 3(3): 47-64.
22. Sayracı B., 2021. Derin öğrenme ve evrimsel sinir ağları (Web sayfası: <https://medium.com/deep-learning-from-deepest/deri%C3%B6%C4%B1nme-ve-evri%C5%9Fi%C3%B6msel->

- [si%CC%87ni%CC%87r-a%C4%9Fflari-446dc8a8d2f](https://www.geogebra.org/m/si%CC%87ni%CC%87r-a%C4%9Fflari-446dc8a8d2f)), (Erişim tarihi: Haziran 2023).
23. Hosla S., 2023. Introduction to pooling layer. (Web sayfası: <https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>), (Erişim tarihi: Haziran 2023).
24. Amidi, S., Amidi, A. 2021. Evrişimli sinir ağları (Web sayfası: <https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-convolutional-neural-networks>), (Erişim tarihi: Mayıs 2023).
25. Kılıçarslan S., Adem K., Çelik M., 2021. An overview of the activation functions used in deep learning algorithms. **Journal of New Results in Science**. 10 (3): 75-88.
26. Srivastava, N., Hinton, G., Krizhevsky A., Sutskever I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, 15 (1): 1929-1958
27. Soydaner D., 2020. A comparison of optimization algorithms for deep learning. **ArXiv:2007.14166v1**.
28. Ruder S., 2017. An overview of gradient descent optimization algorithms. **ArXiv:1609.04747v2**.
29. Kingma, D.P., Ba, J.L., 2015. ADAM: A method for stochastic optimization. **ArXiv:1412.6980v9**.
30. Narkhede, S., 2018. Understanding confusion matrix. (Web sayfası: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>), (Erişim tarihi: Mayıs 2023).
31. Wikipedia, 2023. Confusion matrix. (Web sayfası: https://en.wikipedia.org/wiki/Confusion_matrix), (Erişim tarihi: Mayıs 2023).
32. Wikipedia, 2023. Phi coefficient (Web sayfası: https://en.wikipedia.org/wiki/Phi_coefficient), (Erişim tarihi: Mayıs 2023).
33. TensorFlow. Hakkında. (Web sayfası: <https://www.tensorflow.org/about>), (Erişim tarihi: Haziran 2023).
34. Keras, 2023. Hakkında. (Web sayfası: <https://keras.io/about/>), (Erişim tarihi: Mayıs 2023).
35. Numpy, 2023. Hakkında. (Web sayfası: <https://numpy.org/about/>), (Erişim tarihi: Haziran 2023).

36. Scikit-learn, 2023. Machine learning in python. (Web sayfası: [**https://scikit-learn.org/stable/**](https://scikit-learn.org/stable/)), (Erişim tarihi: Haziran 2023).
37. Pillow, 2023. Pillow. (Web sayfası: [**https://pillow.readthedocs.io/en/stable/**](https://pillow.readthedocs.io/en/stable/)), (Erişim tarihi: Haziran 2023).
38. Matplotlib, 2023. Matplotlib pyplot tutorial. (Web Sayfası: [**https://matplotlib.org/stable/tutorials/introductory/pyplot.html**](https://matplotlib.org/stable/tutorials/introductory/pyplot.html)), (Erişim tarihi: Haziran 2023).
39. Jupyter, 2023. Hakkında. (Web sayfası: [**https://jupyter.org/about**](https://jupyter.org/about)), (Erişim tarihi: Mayıs 2023).



ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Enes ŞEKER
Uyruğu: Türkiye (T.C.)

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi, Elektrik Elektronik Müh.	-
Lisans	Erciyes Üniversitesi, Elektrik Elektronik Müh.	2019
Lise	Mustafa Eminoglu Anadolu Lisesi, Kayseri	2014

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2023- Halen	Unilever	Üretim&Proses

YABANCI DİL

İngilizce