



**DERİN ÖĞRENME YÖNTEMLERİ İLE PERİFERİK
YAYMA GÖRÜNTÜLERİNİN ANALİZİ VE
SINIFLANDIRILMASI**

Yusuf Yargı BAYDİLLİ

**2020
DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ**

**Tez Danışmanı
Dr. Öğr. Üyesi Ümit ATILA**

**DERİN ÖĞRENME YÖNTEMLERİ İLE PERİFERİK YAYMA
GÖRÜNTÜLERİNİN ANALİZİ VE SINIFLANDIRILMASI**

Yusuf Yargı BAYDİLLİ

**Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Doktora Tezi
Olarak Hazırlanmıştır**

**Tez Danışmanı
Dr. Öğr. Üyesi Ümit ATİLA**

**KARABÜK
Temmuz 2020**

Yusuf Yargı BAYDİLLİ tarafından hazırlanan “DERİN ÖĞRENME YÖNTEMLERİ İLE PERİFERİK YAYMA GÖRÜNTÜLERİNİN ANALİZİ VE SINIFLANDIRILMASI” başlıklı bu tezin Doktora Tezi olarak uygun olduğunu onaylarım.

Dr. Öğr. Üyesi Ümit ATİLA

Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Doktora tezi olarak kabul edilmiştir. 23/07/2020

Ünvanı, Adı SOYADI (Kurumu)

İmzası

Başkan : Dr. Öğr. Üyesi Kemal AKYOL (KÜ)

Üye : Prof. Dr. İsmail Rakıp KARAŞ (KBÜ)

Üye : Dr. Öğr. Üyesi Abdülkadir KARACI (KÜ)

Üye : Dr. Öğr. Üyesi M. Kamil TURAN (KBÜ)

Üye : Dr. Öğr. Üyesi Ümit ATİLA (KBÜ)

...../...../2020

KBÜ Lisansüstü Eğitim Enstitüsü Yönetim Kurulu, bu tez ile, Doktora derecesini onamıştır.

Prof. Dr. Hasan SOLMAZ

Lisansüstü Eğitim Enstitüsü Müdürü

“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”

Yusuf Yargı BAYDİLLİ

ÖZET

Doktora Tezi

DERİN ÖĞRENME YÖNTEMLERİ İLE PERİFERİK YAYMA GÖRÜNTÜLERİNİN ANALİZİ VE SINIFLANDIRILMASI

Yusuf Yargı BAYDİLLİ

Karabük Üniversitesi

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:

Dr. Öğr. Üyesi Ümit ATİLA

Temmuz 2020, 142 sayfa

Beyaz kan hücrelerinin insan bağışıklık sisteminden sorumlu olması ve alt-tiplerinin kan hastalıklarının belirlenmesinde önemli bir indikatör olarak görev yapması nedeniyle, beyaz kan hücrelerinin doğru bir şekilde tanımlanması büyük bir önem arz etmektedir. Fakat, hücrelerin hematolojist tarafından manüel olarak incelenmesi süreci uzatmakta, aynı zamanda hataya yatkın bir hale getirmektedir. Dolayısıyla hem bu süreci hızlandırabilecek hem de yüksek başarı oranlarına ulaşabilecek bir makine öğrenmesi modeli gereksinimi oluşmaktadır.

Derin öğrenme, ham verilerden herhangi bir ön-işleme sürecine ihtiyaç duymadan çıkarım yapabilmesi ve bu çıkarımları iyi bir şekilde özetleyebilmesi sayesinde, son on yılda oldukça fazla tercih edilen makine öğrenmesi yöntemlerinden biri olmuştur. Fakat, bu yöntemlerin ulaşacağı başarı değerleri, eğitim aşamasında kullanılan örnek sayısı ile doğru orantılıdır. Ne yazık ki, medikal veri toplama sürecinin gerek etik

gerekse maliyet açısından meşakkatli olması, bu yöntemler ile kararlı bir sunum ortaya koymanın önündeki en büyük engeli oluşturmaktadır. Bu yüzden derin öğrenme yöntemlerini kullanarak medikal veri analizi gerçekleştirmek isteyen araştırmacılar, veri arttırma tekniklerinden faydalanarak veya farklı koşullarda elde edilen verileri birleştirerek veri seti boyutunu yüksek noktalara taşımaya çalışmaktadır. Fakat, birinci seçenek ön-işleme süreci anlamına gelirken, ikinci seçenek ise alan kayması probleminin ortaya çıkmasına neden olmaktadır.

Bu çalışmada medikal verilerin analizi sürecinde sıklıkla karşılaşılan problemlere beyaz kan hücrelerinin alt-tiplerinin sınıflandırılması özelinde çözümler aranmıştır. Bu amaçla çalışmanın ilk bölümünde, az veri ile baştan-sona bir eğitim imkânı sağlayacak derin öğrenme modeli ihtiyacı göz önünde bulundurularak, literatürde sıklıkla kullanılan yöntemler detaylı bir şekilde incelenmiştir. Gerçekleştirilen testler sonunda, Kapsül Ağlarının bu ihtiyacı giderebileceği ulaştığı yüksek doğruluk değerleriyle gösterilmiştir (%96,86). Çalışmanın ikinci aşamasında ise, beyaz kan hücreleri veri setinde gözlemlenen alan kayması probleminin çözümüne yönelik yeni bir model önerilmiştir. Model sayesinde yalnızca bir veri setinden çıkarılan özelliklerin, diğer veri setlerinde de efektif bir şekilde kullanılması sağlanmıştır. Dokuz farklı veri seti üzerinde yapılan testlerde başarılı sonuçlar elde edilmiş, modelin tüm veri setleri üzerinde ulaştığı toplam doğruluk değeri ise %98,09 olarak ölçülmüştür.

Anahtar Sözcükler : Medikal veri analizi, beyaz kan hücreleri, derin öğrenme, alan adaptasyonu, sınıflandırma.

Bilim Kodu : 92431

ABSTRACT

Ph. D. Thesis

ANALYSIS AND CLASSIFICATION OF PERIPHERAL BLOOD SMEAR IMAGES USING DEEP LEARNING METHODS

Yusuf Yargı BAYDİLLİ

**Karabük University
Institute of Graduate Programs
Department of Computer Engineering**

Thesis Advisor:

Assist. Prof. Dr. Ümit ATILA

Jul 2020, 142 pages

Due to the fact that white blood cells are responsible for the human immune system and sub-types of them serve as an important indicator to the detection of blood diseases, it is so important that white blood cells are accurately identified. However, the manual examination of these cells by the hematologist prolongs the process and at the same time makes it prone to error. Therefore, there is a need for a machine learning model that able to accelerate this process and achieve high success rates.

Deep learning has become one of the most preferred methods in machine learning in the last decade, since it can extract features from raw data without any pre-process stage and summarize these inferences well. However, the success of these methods is directly proportional to the number of samples used in the training phase. Unfortunately, the fact that the medical data collection process is difficult due to both ethical and cost reasons constitutes the biggest obstacle to making a stable presentation

with these methods. Therefore, researchers who want to perform medical data analysis using deep learning methods try to increase the data set size by taking advantage of data augmentation techniques or by combining the data obtained under different conditions. However, while the first option means to pre-process, the second option causes the domain shift problem.

In this study, solutions to the problems frequently encountered during the medical data analysis were searched in the basis of classification of white blood cells sub-types. For this purpose, in the first part of the study, by considering the need for a deep learning model that will provide an end-to-end training with few samples, the methods frequently used in the literature are examined in detail. At the end of the tests carried out, it was shown that the Capsule Networks could meet this need with high accuracy values (96.86%). In the second stage of the study, a new model was proposed to solve the domain shift problem observed in the white blood cell data set. Thanks to the model, the features extracted from only one data set was able to be used effectively in other data sets. Tests performed on nine different data sets yielded successful results and the total accuracy of the model on all data sets was measured as 98.09%.

Keywords : Medical data analysis, white blood cells, deep learning, domain adaptation, classification.

Science Code : 92431

TEŞEKKÜR

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam *Dr. Öğr. Üyesi Ümit ATILA*'ya;

Doktora sürecinde bilimsel ve akademik anlamda katkı sağlayan değerli hocalarım *Prof. Dr. İsmail Rakıp KARAŞ*'a, *Dr. Öğr. Üyesi M. Kâmil TURAN*'a ve *Dr. Öğr. Üyesi Abdullah ELEN*'e;

Meslek ve öğrenim hayatım süresince hiçbir yardımı esirgmeden yanımda olan, başta amirlerim olmak üzere tüm *Karabük Üniversitesi Fen Bilimleri Enstitüsü* ve *Bilgisayar Mühendisliği Anabilim Dalı akademisyenlerine* ve *çalışanlarına*;

Bu günlere kadar gelmemde en büyük emeğe sahip olan *aileme*;

Ömrümü özel yapan *eşime* ve *oğluma*;

Ve son olarak, her şeyin gerçekleşmesini mümkün kılan *Yüce Yaradan*'a;

Sonsuz teşekkürlerimi sunarım...

İÇİNDEKİLER

	<u>Sayfa</u>
KABUL.....	ii
ÖZET.....	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ.....	xii
ÇİZELGELER DİZİNİ	xv
SİMGELER VE KISALTMALAR DİZİNİ	xvi
BÖLÜM 1	1
GİRİŞ	1
BÖLÜM 2	5
BEYAZ KAN HÜCRELERİ	5
2.1. GRANÜLOSİTLER.....	5
2.1.1. Nötrofiller	5
2.1.2. Eozinofiller	5
2.1.3. Bazofiller	6
2.2. MONOSİTLER	6
2.3. LENFOSİTLER.....	6
2.4. ÖZET.....	6
BÖLÜM 3	8
DERİN ÖĞRENME.....	8
3.1. MAKİNE ÖĞRENMESİ VE MEDİKAL GÖRÜNTÜ İŞLEME	8
3.2. DERİN ÖĞRENME NEDİR?	9
3.3. DERİN ÖĞRENME YÖNTEMLERİ	12
3.4. DERİN ÖĞRENME VE BİLGİSAYARLI GÖRÜ	15
3.4.1. Konvolüsyonel Sinir Ağları (CNN).....	15

Sayfa

3.4.1.1. Konvolüsyonel Sinir Ağlarının Elemanları	16
3.4.1.2. Konvolüsyonel Ağların Eğitim Süreci.....	23
3.4.1.3. Konvolüsyon Ağ Katmanlarının Görselleştirilmesi.....	28
3.4.2. Diğer Yöntemler	30
BÖLÜM 4	31
BEYAZ KAN HÜCRELERİNİN KAPSÜL AĞLARI İLE SINIFLANDIRILMASI.....	31
4.1. BEYAZ KAN HÜCRELERİNİN SINIFLANDIRILMASI ALANINDA YAPILMIŞ ÇALIŞMALAR.....	31
4.1.1. Klasik Makine Öğrenmesi Yöntemleri Kullanılarak Gerçekleştirilen Çalışmalar	31
4.1.2. Derin Öğrenme Yöntemleri Kullanılarak Gerçekleştirilen Çalışmalar	35
4.2. DERİN ÖĞRENMEDEKİ PROBLEMLER	39
4.2.1. CNN Yönteminin Sahip Olduğu Problemler.....	40
4.3. KAPSÜL AĞLARI	42
4.3.1. Hiyerarşik Sunum	44
4.3.2. Bakış Açısı Değişmezliği (Invariance).....	45
4.3.3. Bakış Açısı Eşitsizliği (Equivarance)	46
4.3.4. Ters Grafik (Inverse Graphic)	46
4.3.5. Poz Matrisi.....	47
4.3.6. Poz Matrisi ile Tahmin Gerçekleştirmek	49
4.3.7. CNN Nerede Başarısız Oluyor?.....	50
4.3.8. Vektörler ile Objelerin Sunumu.....	52
4.3.9. Kapsül Ağlarının Yapısı	53
4.3.9.1. Kodlayıcı.....	53
4.3.9.2. Çözücü	62
4.3.10. Kapsül Ağlarının Avantajları ve Dezavantajları.....	65
4.4. VERİ SETİ	65
4.5. ÖNERİLEN MODEL (WBCaps).....	67
4.6. DENEYSEL SONUÇLAR.....	69
4.6.1. Sonuçların Diğer Yöntemler ile Kıyaslanması.....	72
4.7. TARTIŞMA.....	76

	<u>Sayfa</u>
BÖLÜM 5	79
ALAN KAYMASI (DOMAIN SHIFT).....	79
5.1. DOMAIN VE GÖREV	81
5.1.1. Domain ve Görev Eşitsizliği.....	82
5.2. TRANSFER ÖĞRENME.....	83
5.2.1. Derin Transfer Öğrenme.....	85
5.2.2. Negatif Transfer.....	88
5.2.3. Domain Adaptasyonu	89
BÖLÜM 6	91
BEYAZ KAN HÜCRELERİ VERİ SETLERİNDE GÖZLEMLENEN ALAN KAYMASI PROBLEMİNİN ÇÖZÜMÜNE YÖNELİK BİR ÇOKLU- HEDEF ALAN ADAPTASYONU MODELİNİN GELİŞTİRİLMESİ.....	91
6.1. VERİ SETLERİ.....	92
6.2. ÖNERİLEN MODEL.....	97
6.2.1. Veri Arttırma (Data Augmentation)	98
6.2.2. Üretici Çekişmeli Ağlar (GAN).....	99
6.2.3. İnce-ayar (Fine-tuning).....	101
6.2.4. Özet.....	103
6.3. DENEYSEL SONUÇLAR.....	104
6.4. TARTIŞMA.....	121
BÖLÜM 7	125
GENEL SONUÇLAR	125
KAYNAKLAR	129
ÖZGEÇMİŞ	142

ŞEKİLLER DİZİNİ

Sayfa

Şekil 3.1. Veri boyutuna göre ML kategorileri ve yöntem seçimi: sınıflandırma, regresyon, kümeleme ve boyut azaltma.....	9
Şekil 3.2. Makine öğrenmesi ve derin öğrenme.....	11
Şekil 3.3. Konvolüsyonel ağların üç önemli mekanizması (yerel bağlantı, ağırlık paylaşımı ve boyut azaltma)	16
Şekil 3.4. Sekiz rakamının matris formu	17
Şekil 3.5. Konvolüsyon operasyonu ve sonrasında elde edilen özellik haritası.....	18
Şekil 3.6. Bir deve görüntüsünün çeşitli filtrelerle özellik çıkarımı	18
Şekil 3.7. İki farklı filtre ile özellik çıkarımı.....	19
Şekil 3.8. Stride ve zero-paddding	20
Şekil 3.9. ReLU operasyon.....	20
Şekil 3.10. Alt-örnekleme operasyonu	21
Şekil 3.11. Alt-örnekleme işlemi sonrası özellik haritası	21
Şekil 3.12. Bir CNN örneği.....	22
Şekil 3.13. Konvolüsyon katmanının detaylı gösterimi	22
Şekil 3.14. Örnek bir sınıflandırıcı.....	23
Şekil 3.15. MLP de geri yayılım	24
Şekil 3.16. Standart bir MLP ve CNN	25
Şekil 3.17. CNN de ileri besleme süreci	25
Şekil 3.18. Bir konvolüsyon ve alt-örnekleme katmanının MLP olarak sunumu	26
Şekil 3.19. CNN de hatanın geriye yayılması süreci	27
Şekil 3.20. Çok boyutlu CNN katmanı	27
Şekil 3.21. Bir aracın CNN ile özellik çıkarımı	28
Şekil 3.22. Bir insan yüzünün CNN ile özellik çıkarımı.....	28
Şekil 3.23. MNIST veri setinin CNN katmanlarında görselleştirilmesi	29
Şekil 4.1. CNN'deki yer değiştirme değişmezliği problemi	41
Şekil 4.2. CNN'deki bakış açısı değişmezliği problemi	41
Şekil 4.3. Maksimum havuzlama işlemi	42
Şekil 4.4. HMAX modeli ve görsel korteksin hiyerarşik modeli.....	43

Sayfa

Şekil 4.5. Ağacın hiyerarşik sunumu.....	44
Şekil 4.6. Nöronların özelliklere göre aktivasyonu.....	45
Şekil 4.7. Kapsüllerin objenin uzaysal değişikliklerine göre adaptasyonu	46
Şekil 4.8. Rendering ve ters grafik	47
Şekil 4.9. Poz matrisleri aracılığıyla özelden genele gitmek	48
Şekil 4.10. Ağız ve gözün pozisyonuna göre yüzün pozisyon tahmini	50
Şekil 4.11. CNN ve Kapsül Ağlarının giriş özelliklerini yorumlaması	51
Şekil 4.12. Objelerin vektörel olarak ifadesi.....	52
Şekil 4.13. Kapsül Ağlarının kodlayıcı bölümü.....	53
Şekil 4.14. Kapsül Ağlarındaki birincil kapsüller bölümü.....	54
Şekil 4.15. Bir kapsülün sahip olduğu örnekleme parametreleri	56
Şekil 4.16. Ağırlık matrisi ile vektörel özelliklerin kazandırılması.....	57
Şekil 4.17. Dinamik yönlendirme algoritması	58
Şekil 4.18. Düşük-yüksek seviyeli kapsüller arasındaki bağıntı.....	59
Şekil 4.19. Kapsül Ağlarında eşleştirme etkisi	60
Şekil 4.20. Eşleştirme sabitlerinin güncellenmesi.....	61
Şekil 4.21. Doğru ve yanlış tahmin edilen için yüksek seviye kapsüller için çizilen kayıp değerleri	62
Şekil 4.22. Çözücü bölümü ile görüntülerin yeniden inşa edilmesi.....	63
Şekil 4.23. Kapsüller sayesinde objelerin farklı özelliklerinin algılanması ve yeniden oluşturulması	64
Şekil 4.24. Kapsül Ağlarının çözücü bölümü	64
Şekil 4.25. Önerilen mimari.	68
Şekil 4.26. Ağın eğitim ve doğrulama değerleri.	69
Şekil 4.27. Orijinal test verileri ve yeniden inşa edilmiş halleri, görüntülerin gerçek ve tahmin edilen sınıfları.	71
Şekil 4.28. Veri setinin benzerliğine ve boyutuna göre yöntem seçimi.....	74
Şekil 4.29. Baştan eğitilen CNN ve transfer öğrenme modellerinin eğitim ve doğruluk değerleri.	75
Şekil 5.1. Geçmiş görevlerden öğrenilen bilgilerin yeni görevlere aktarılması.	79
Şekil 5.2. Araba sınıflandırması senaryoları	80
Şekil 5.3. Kaynak ve hedef domainler arasındaki dağılım farklılığına göre çizilen karar sınırları.	84
Şekil 5.4. Örnek-tabanlı derin TL	86

	<u>Sayfa</u>
Şekil 5.5. Haritalama-tabanlı derin TL.....	86
Şekil 5.6. Ağ-tabanlı derin TL.....	87
Şekil 5.7. Çekişme-tabanlı derin TL	88
Şekil 6.1. Veri setlerinde yer alan lenfosit örneklerinin piksel şiddet dağılım grafiği.	95
Şekil 6.2. Veri setlerinin t-SNE sunumları.....	96
Şekil 6.3. GAN mimarisi.....	99
Şekil 6.4. Eğitim ve ince-ayar sürecinde kullanılan CNN modeli.	102
Şekil 6.5. Önerilen modelin diyagramı.	104
Şekil 6.6. Veri çoğaltma optimizasyon süreci.	105
Şekil 6.7. Veri çoğaltma sonrası elde edilen görüntü örnekleri (her bir sınıfa ait).	108
Şekil 6.8. WGAN-GP modeli.....	111
Şekil 6.9. Üretici modelin üretici ve ayırıcı bölümleri için her bir devirde hesaplanan kayıp değerleri.....	112
Şekil 6.10. Üretici model tarafından üretilen görüntüler.	113
Şekil 6.11. Kaynak ve hedef domainleri içeren t-SNE sunumu.....	114
Şekil 6.12. CNN modelinin eğitim sonunda elde ettiği eğitim-doğrulama eğrileri.	116
Şekil 6.13. Sınıflandırıcı modelin hedef domainler üzerinde ulaştığı başarıları gösteren karmaşıklık matrisleri.	118
Şekil 6.14. Tüm veri setleri birleştirilerek yapılan sınıflandırma sonuçlarını içeren karmaşıklık matrisi (çoklu-hedef DA).	120

ÇİZELGELER DİZİNİ

	<u>Sayfa</u>
Çizelge 2.1. WBC alt-tipleri.....	7
Çizelge 4.1. Literatür özeti.....	38
Çizelge 4.2. Kullanılan veri setinin özellikleri.	66
Çizelge 4.3. Karmaşıklık matrisi ve yanlış tahmin edilen WBC alt-tipleri.	71
Çizelge 4.4. Sınıflandırma raporu.	72
Çizelge 5.1. TL türleri.....	84
Çizelge 6.1. Kaynak ve hedef domainler.	94
Çizelge 6.2. Veri arttırımı sırasında sınıflandırmaya en fazla katkı sağlayan 20 poliçe.....	106
Çizelge 6.3. Eğitim ve test setleri.....	117
Çizelge 6.4. Oluşturulan genel veri seti.	120
Çizelge 6.5. Tüm verilerle yapılan deneyin sınıflandırma raporu.	120

SİMGELER VE KISALTMALAR DİZİNİ

KISALTMALAR

- 2B : 2 Boyutlu
- 3B : 3 Boyutlu
- ANN : Artificial Neural Networks (Yapay Sinir Ağları)
- BP : Back Propagation (Geri-yayılım Algoritması)
- CAD : Computer-aided Diagnosis (Bilgisayar-destekli Tanı)
- CNN : Convolutional Neural Networks (Konvolüsyonel Sinir Ağları)
- DBN : Deep Belief Networks (Derin İnanç Ağları)
- DA : Domain Adaptation (Alan Adaptasyonu)
- DL : Deep Learning (Derin Öğrenme)
- DNN : Deep Neural Networks (Derin Sinir Ağları)
- ELM : Extreme Learning Machines (Aşırı Öğrenme Makineleri)
- GAN : Generative Adversarial Networks (Üretici Çekişmeli Ağlar)
- GMM : Gaussian Mixture Models (Gauss Karışım Modelleri)
- GPU : Graphical Processing Unit (Grafik İşleme Birimi)
- KNN : K - Nearest Neighbour (K - En Yakın Komşu)
- LSTM : Long Short-Term Memory (Uzun Kısa-Vadeli Hafıza)
- ML : Machine Learning (Makine Öğrenmesi)
- MLP : Multi Layer Perceptron (Çok Katmanlı Algılayıcı)
- RBM : Restricted Boltzmann Machines (Kısıtlı Boltzmann Makineleri)
- RNN : Recurrent Neural Networks (Yinelemeli Sinir Ağları)
- RF : Random Forest (Rastgele Orman)
- SGD : Stochastic Gradient Descend (Stokastik Gradyan İniş)
- SVM : Support Vector Machines (Destek Vektör Makineleri)
- TL : Transfer Learning (Transfer Öğrenme)
- WBC : White Blood Cells (Beyaz Kan Hücreleri)

BÖLÜM 1

GİRİŞ

Kan, vücudun yaşam fonksiyonlarını devam ettirebilmesi için gereken temel görevleri üstlendiğinden dolayı, insanoğlu için hayati bir öneme sahiptir. Kanın yaşamımızda önemli bir yer tutmasının bir diğer nedeni de insan sağlığı ile ilgili önemli bilgiler barındırmasıdır. Bu özelliği sayesinde, yapılan kan testleri ile kişinin geniş bir spektrumda analizi hastalık analizi gerçekleştirilebilmektedir [1].

Preferik kan yayma kan üzerinde gerçekleştirilen bu testlerden biridir. Bu testte hastadan alınan kan örneği lamel üzerine yayılır ve bir boyama prosedüründen geçirilir. Kandaki bileşenlerin sahip oldukları spesifik özelliklerin boyama sürecine farklı tepkiler vermesi sayesinde, hücrelerin kendi içinde ve diğer hücrelerle ayrıştırılması sağlanır [2].

Vücudumuzda bulunan yaklaşık 5 litre kanın bir damlasında dahi, binlerce bileşen yer alır. Bu bileşenlerden beyaz kan hücreleri (WBC), insanoğlunun bağışıklık sisteminin önemli bir kısmından sorumludur. Dolayısıyla, WBC'lerin temel niceliklerinin doğru tanımlanması oldukça önemlidir. Bir diğer önemli ayrıntı da WBC alt-tiplerinin doğru bir şekilde sınıflandırılmasını sağlamaktır. Çünkü, WBC'lerin sayısı, oranı, şekli gibi nicelikler bazı hastalıkların tespit edilmesinde önemli bir indikatör olarak görev yapabildiği gibi (ör: lösemi), bu hücrelerin alt-sınıflarında gözlemlenecek anomalilerin de kimi hastalıkların habercisi olduğu bilinen bir gerçektir (ör: nötrofil lökositöz) [3].

WBC'lerin tespiti çoğu zaman bir hematolojist tarafından, preferik yayma veya kemik iliği yayma örneklerinin ışık mikroskobu altında gözle incelenmesi ile sağlanır. Fakat bu süreç hem zaman alıcıdır hem de incelemeyi yapan uzmanın yeteneklerine bağlıdır. Bu nedenle, WBC'lerin hızlı ve doğru bir şekilde sınıflandırılması için kullanılacak bilgisayar destekli tanı sistemleri (CAD) gereksinimi oluşur. Bu

ihtiyacın farkında olan arařtırmacılar da gemiřten gnmze bu problemi en hızlı ve en doęru řekilde zözmeye yönelik alıřmalar gerekleřtirmiřtir. Gerekleřtirilen bu alıřmaları klasik makine öęrenmesi (ML) ve derin öęrenme (DL) olarak iki bařlık altında incelemek mümkündür.

Klasik ML yöntemlerini kullanan arařtırmacılar WBC tanımlanmasını genellikle üç ařamada gerekleřtirmektedir: *segmentasyon (segmentation)*, *özellik ıkarımı (feature extraction)* ve *sınıflandırma (classification)*. Segmentasyon ařamasında hücrelerin arkaplandan ve dięer hücrelerden ayrıřtırılması saęlanır, özellik ıkarımında ise hücrelerin renk, řekil gibi özellikleri ıkartılarak hücreleri tanımlayacak özellik vektörleri oluřturulur [4,5]. Daha sonra ise bu vektörler ML yöntemlerine giriş deęeri olarak verilerek öęrenme ve sınıflandırma gerekleřtirilir [6].

Gemiřten gnmze oldukça bařarılı segmentasyon algoritmaları önerilmiř olsa da karmařık yapıdaki nesnelere ve görüntüler için segmentasyon gerekleřtirmek hala daha zor bir görevdir. Dięer yandan segmentasyon ve özellik ıkarımı ařamasında yapılacak hataların da sınıflandırma performansını olumsuz etkilemesi muhtemeldir. Bu bakımdan arařtırmacılar, dięer bilgisayarlı görü problemlerinde de olduęu gibi Derin Öęrenme (DL) yöntemlerine yönelmiřlerdir [7].

DL özellikle son on yılda oldukça popülerleřmiř bir ML alanıdır. Bu yöntemlerin ML yöntemlerine nazaran saęladıęı en büyük avantaj, ML yöntemlerinde görülen segmentasyon ve özellik ıkarımı sürecine ihtiyaç duymadan verinin özelliklerini ıkarabilme ve özetleyebilme yeteneęine sahip olmasıdır. Özellikle mikroskopik verilerin makroskopik verilere nazaran daha karmařık veriler olması DL yöntemlerini medikal veri iřleme alanında özel kılan nedenlerden biri olmaktadır [7].

Ne yazık ki, ML yöntemleri gibi DL yöntemlerinin de handikapları bulunmaktadır. Bu handikaplardan en büyüęü ise bu yöntemlerin yüksek sayıda eğitim verisine ihtiyaç duymasıdır [8]. Fakat medikal veri toplama sürecinin etik ve maliyet gibi nedenlerden oldukça sıkıntılı olması, DL sistemleri ile kararlı bir model ortaya koyabilecek yeterlilikte veriye sahip olmayı daha da zorlařtırmaktadır [9]. Bu nedenden dolayı da medikal veri analizi gerekleřtirmek isteyen arařtırmacılar problemi özecek modeli

oluşturmadan önce veri seti boyutunu arttırmaya yönelik işlemler gerçekleştirilmektedirler.

Veri setlerinin boyutunu *veri arttırımı (data augmentation)* teknikleri uygulayarak arttıran araştırmacılar için hem sürecin uzaması hem de sınıflandırmaya olumsuz katkı yapabilecek örneklerin de üretilmesi söz konusu olmaktadır [10]. Diğer yandan, birden fazla veri seti birleştirilerek sahip olunan veri sayısı arttırılmaya çalışılırsa, o zaman da *alan kayması (domain shift)* adı verilen problemin meydana gelmesi muhtemeldir [11]. Sonuç olarak medikal veri analizinde kullanılabilir, hem az veri ile yüksek başarı oranları yakalayabilecek hem de DL'nin sunduğu avantajlardan da faydalanmayı sağlayacak bir model ihtiyacı ortaya çıkmaktadır.

Alan kayması sadece medikal verilerde değil tüm bilgisayarlı görü veri setlerinde gözlemlenebilen bir fenomendir. Verinin elde edilmiş sürecindeki faktörlerden de ortaya çıkabilen bu problem nedeniyle, tüm veri setlerinde yüksek doğruluk ile çalışacak bir model oluşturulması mümkün olamamaktadır. Bu nedenle araştırmacılar, *transfer öğrenme (transfer learning)* ve *alan adaptasyonu (domain adaptation)* tekniklerinden faydalanarak, bir veri setinden öğrenilen bilgilerin diğer veri setlerinde de başarılı bir şekilde kullanılmasını amaçlamaktadırlar [11].

Medikal verileri bu açıdan bakıldığında özel kılan bir diğer konu ise, medikal verilerin çoğunlukla birçok denekten alınan örnekler ile oluşturulmasıdır. Bu nedenle sadece görüntüleme aracının değil, deneklerin sahip olduğu biyolojik varyasyonlarında doğru bir karar mekanizması oluşturmayı olumsuz etkilemesi beklenir [12]. Aynı zamanda transfer ve adaptasyon süreci çoğunlukla bir veri setinden diğerine doğru gerçekleştirilmesi, her yeni veri seti için bu sürecin tekrarlanması anlamına gelir.

Bu nedenle, bu tez çalışmasında, medikal veri analizinde sıklıkla karşılaşılan bu iki probleme, bir medikal veri türü olan periferik yayma görüntüleri kullanılarak bir çözüm üretilmesi amaçlanmıştır. Bu amaçla tezin ilk aşamasında bilgisayarlı görü alanında sıklıkla kullanılan DL yöntemleri derinlemesine analiz edilmiş, oldukça az sayıda örnek içeren bir WBC veri seti kullanılarak bu yöntemler üzerinde testler

gerçekleştirilmiştir. Testler sonucunda ise en yüksek başarı oranlarının Kapsül Ağları (CapsNet) ile elde edilebildiği gösterilmiştir.

Tezin ikinci aşamasında ise WBC'lerin sınıflandırılmasında kullanılacak bir *çoklu-hedef alan adaptasyonu (multi-target domain adaptation)* modeli önerilmiştir. Bu doğrultuda, yalnızca bir veri seti *kaynak alan (source domain)* olarak seçilmiş, bu veri setinden çıkarılan özelliklerin de *hedef alan (target domain)* veri setlerinin sınıflandırılması amacıyla efektif olarak kullanılması sağlanmıştır. Önerilen modelin, *dokuz* farklı veri seti üzerinde gerçekleştirilen deneylerde yüksek doğrulukla çalıştığı gözlemlenmiştir.

Bu tez *sekiz* bölümden meydana gelmektedir. *BÖLÜM 2*'de WBC ve alt-tipleri tanıtarak, okuyucunun problem hakkında fikir sahibi olması amaçlanmaktadır. *BÖLÜM 3*'te ise "DL nedir?", "Neden DL?" ve "Bilgisayarlı görüde sıklıkla kullanılan DL yöntemleri nelerdir?" sorularının cevabı verilecektir. *BÖLÜM 4*, WBC'lerin sınıflandırılmasında daha önce gerçekleştirilen çalışmaları, bu tez çalışmasında kullanılan yöntemi, elde edilen sonuçları ve diğer yöntemlerle kıyaslamaları içermektedir. *BÖLÜM 5*'te ise, alan kayması problemi tanıtılmış, literatürde bu problemin ortadan kaldırılmasına yönelik yapılan varsayımlardan ve kullanılan yöntemlerden bahsedilmiştir. *BÖLÜM 6*'da WBC veri setlerinde gözlemlenen alan kayması problemine yol açan sebepler, önerilen çoklu-hedef alan adaptasyonu modelinin oluşturulması sürecinde izlenen adımlar ve model ile elde edilen sonuçlar görülebilir. Tez çalışmasının sonuçlarının genel olarak değerlendirilmesi, sonuçlardan elde edilen genel çıkarımlar, çalışmanın gelecekte ve klinikteki etkileri ise *BÖLÜM 7*'de sunulmuştur.

BÖLÜM 2

BEYAZ KAN HÜCRELERİ

Sağlıklı bir insanın damarlarında vücut ağırlığının yaklaşık %7'si kadar kan bulunur. Eğer bir kan örneği mikroskop altında incelenirse, plazma adı verilen sıvıdan ve çeşitli hücrelerden meydana geldiği görülecektir. Bu hücrelerin hacimce %45'i kırmızı kan hücreleri, %1'i ise beyaz kan hücreleridir. Kırmızı kan hücreleri (eritrositler) O₂ ve CO₂ taşımaktan sorumlu iken, beyaz kan hücreleri (lökositler) enfeksiyonlarla mücadele ederler [1]. WBC'ler genellikle üç grup altında incelenirler: *granülositler*, *monositler* ve *lenfositler*.

2.1. GRANÜLOSİTLER

Granülositler yapısında sayısız lizozom ve salgı vezikülleri barındırır. Bu organelleri de morfolojilerine ve boyama özelliklerine üç sınıf altında incelemek mümkündür [1].

2.1.1. Nötrofiller

Çoklu-çekirdek yapısından dolayı polimorfonükleer lökositler olarak da bilinen *nötrofiller*, kanda en yaygın olarak bulunan granülosit tipidir. Özellikle bakteri gibi mikro organizmaları yok ettiğinden, bakteriyel enfeksiyona karşı doğal bağışıklıkta kilit rol oynar. Nötrofil sayısının azalması nötropeni, artması ise nötrofil olarak adlandırılır [1,13].

2.1.2. Eozinofiller

Eozinofiller, genellikle iki nükleer lobdan oluşur ve sitoplazması turuncumsu granüller ile doludur. Granüllerin altında bulunan sitoplazması ise soluk mavi renklidir [14].

Kandaki deęerlerinin artmasına eozinofili adı verilir. Alerjik rahatsızlıklara oldukça duyarlıdırlar [13].

2.1.3. Bazofiller

Lökositler arasında en nadir bulunan türdür. Histamin, serotonin ve heparin bakımından oldukça zengindirler. Çekirdekleri düzensizdir. Sitoplazması büyük ve mor granüller içerdiğinden, çekirdeklerini görebilmek çoęu zaman imkansızdır. Özellikle kronik miyelojenöz lösemide iyi bir indikatör olarak görev yaparlar [13,14].

2.2. MONOSİTLER

Monositler dolaşımda yer alan en büyük boyutlu WBC türüdür. Sitoplazması gri-mavi renktedir. Çekirdeęi geniş ve kıvrımlı yapıdadır. Bazı kronik enfeksiyonlarda, tüberküloz, kronik miyeloid ve akut lösemide sayılarında artış görülür [14].

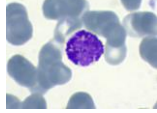
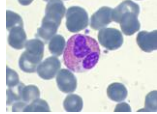
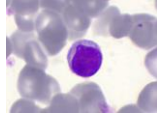
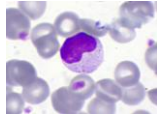
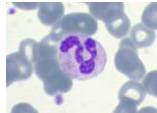
2.3. LENFOSİTLER

B lenfositleri ve T lenfositleri olarak iki türü bulunur. Bağışıklık sisteminde önemli bir yer tutarlar. T lenfositleri sayesinde birçok yabancı işgalcinin yok edilmesi sağlanır. B lenfositleri ise dięer WBC türlerinin aksine humoral bağışıklıktan sorumludur. Bu özellikleri sayesinde bir enfeksiyonu hatırlayan antikorları üretirler. B lenfositleri mevcut aşuların çoęunda etkin rol oynamaktadırlar [15].

2.4. ÖZET

Görüldüğü üzere WBC'ler beş temel hücre tipinden oluşmaktadır; *bazofil*, *eozinofil*, *lenfosit*, *monosit* ve *nötrofil*. Bu çalışmada da literatürde yer alan çalışmalara paralel olarak bu beş temel hücrenin sınıflandırılması sağlanacaktır. Tezin geri kalanında bu beş hücre tipi *WBC alt-tipleri* olarak adlandırılacaktır. Her bir hücre tipinin spesifik özelliklerini içeren Çizelge 2.1 aşağıda görülebilir.

Çizelge 2.1. WBC alt-tipleri [1,13,14].

WBC alt-tipi	Görünüş	Çap (μm)	Sitoplazma	Çekirdek	Granüller
Bazofil		12 – 15	Soluk mavi, çoğunlukla görmek mümkün değil	Tekli, ikili, üçlü görünümde	Büyük morumsu ve siyaha yakın
Eozinofil		10 – 12	Ufak ufak granüller	Çoğunlukla iki loblu	Turuncu ve kırmızı yoğunluklu
Lenfosit		7 – 8 Veya 12 – 15	Berrak ve soluk mavi	Tek parçalı ve girintili	Mevcut değil
Monosit		15 – 30	Gri-mavi	Tek parçalı, U veya fasulye şeklinde	Mevcut değil
Nötrofil		10 – 12	Çok soluk, şeffafa yakın veya çok açık pembe	Çok parçalı	Fark edilmeyecek şekilde ufak ve açık renkli

BÖLÜM 3

DERİN ÖĞRENME

Günümüzde oldukça fazla alanda karşımıza çıkan ML'nin sıklıkla kullanıldığı alanlardan biri de medikal görüntü analizidir. Medikal görüntülerde ML'nin popüler olmasının sebebi, lezyon ve organların kompleks yapılarda bulunmasıdır. Dolayısıyla, bu yapıların çözümünde de yine bu denli komplekslikte yapılara ihtiyaç duyulmaktadır. Kompleks yapıların beraberinde getirdiği yüksek sayıda parametre de el analiz edilemeyeceğinden bilgisayar ve yapay zeka hesaplamaları kullanılmaktadır [7,9].

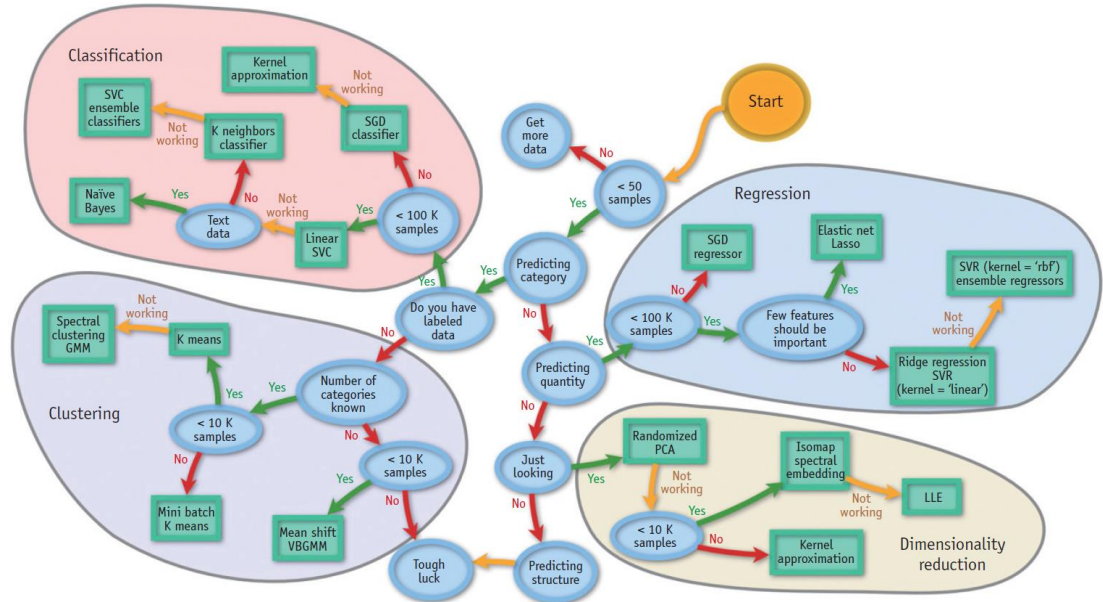
3.1. MAKİNE ÖĞRENMESİ VE MEDİKAL GÖRÜNTÜ İŞLEME

ML metotlarını danışmanlı ve danışmansız öğrenme olarak iki kategori altında incelemek mümkündür. Danışmanlı yöntemler, eğitilen verilerden bir çıkarım yaparak fonksiyon üretirler. Diğer bir deyişle, eğitilecek veriden nümerik ya da sembolik vektörler oluşturularak, ilgili çıktıyı tanımlayacak karakteristikte verinin ifade edilmesini sağlarlar. Eğer çıkış değerleri sürekli ise, problem *regresyon* olarak, fakat çıkış verileri kategorize edilmiş ise, *sınıflandırma* olarak tanımlanır. Danışmanlı öğrenmenin aksine, danışmansız öğrenme çıkış verisi ile ilgilenmez. Bu yöntemin amacı yapıdaki önemli ve gizli olan niteliklerin ortaya çıkarılmasını sağlamaktır. Bu anlamda, istatistikteki kümelenme analizine oldukça benzer bir yöntemdir (Şekil 3.1).

ML ve medikal görüntü analizi ile ilgili yapılan çalışmalar konusunda literatür incelemesi yapıldığında, en sık kullanılan yöntemlerden bir tanesinin Bayesian model olduğu görülmektedir. Bu model giriş verilerinin olasılık dağılımına odaklanan, oldukça basit bir yöntem olsa da örnek olarak rRNA dizilerinin incelenmesinde oldukça başarılıdır. Sık kullanılan diğer yöntemlere Destek Vektör Makineleri (SVM) ve Yapay Sinir Ağları (ANN) da örnek olarak verilebilir. Bu yöntemler kullanılırken,

araştırmacıların yaptıkları belki de ilk şey, eldeki verilerin nümerik veya sembolik vektörler haline getirilmesidir. Çeşitli metotlar aracılığı ile ön-işleme aşamasından geçirilen görüntüler, ML yöntemlerine giriş değeri olarak verilerek regresyon, sınıflandırma ve danışmansız öğrenmede kullanılabilir hale getirilmektedir [9].

Her ne kadar bu yöntemler dikkate değer performanslar sağlasa da *sıfırlanan türevler* (sigmoid fonksiyonunun türevlerinin sıfır değerine hızlı yaklaşması dolayısıyla bir süre sonra öğrenmenin durması) ve *ezberleme* gibi problemler araştırmacıları farklı çözümler aramaya itmiştir [9].



GMM: Gaussian Mixture Model, LLE: Locally-linear Embedding, PCA: Principal Component Analysis, SGD: Stochastic Gradient Descent, SVC: Support Vector Classification, SVR: Support Vector Regression, VBGMM: Variational Bayesian Gaussian Mixture Model.

Şekil 3.1. Veri boyutuna göre ML kategorileri ve yöntem seçimi: sınıflandırma, regresyon, kümeleme ve boyut azaltma [16].

3.2. DERİN ÖĞRENME NEDİR?

Deep Learning, 2006 yılından itibaren ML araştırmacıları arasında çalışılmaya başlanmış, yeni ve dikkat çekici bir alan olmuştur. Geçtiğimiz son on yıl itibari ile, sinyal ve bilgi işlemede geniş bir alanda kullanılmış, kullanılmaya da devam edilmektedir. Deep Learning kavramı için çeşitli tanımlamalar yapılmış olsa da, bu tanımlar genel olarak iki gerçeği ortaya koymuştur. Bunlar;

- Birden fazla katman veya aşama içeren non-lineer bilgi işleme modellerine sahiptirler.
- Ardışık ve karmaşık katmanlar arasından, danışmanlı ve danışmansız öğrenmeyi özellik çıkarımında başarılı bir şekilde kullanabilen sistemlerdir [17].

DL'nin araştırmacılar arasında yaygın olarak kullanımının ise üç önemli sebebi vardır:

- Bilgisayar çiplerinin işleme kabiliyetlerinin artması,
- Donanımların işleme kabiliyetinin getirdiği maliyetin azalması,
- ML ve sinyal/bilgi işlemedeki son gelişmelerdir [18].

Bu gelişmeler DL metodlarının, kompleks ve non-lineer fonksiyonların üstesinden başarılı bir şekilde gelmesini sağlamıştır [19,20].

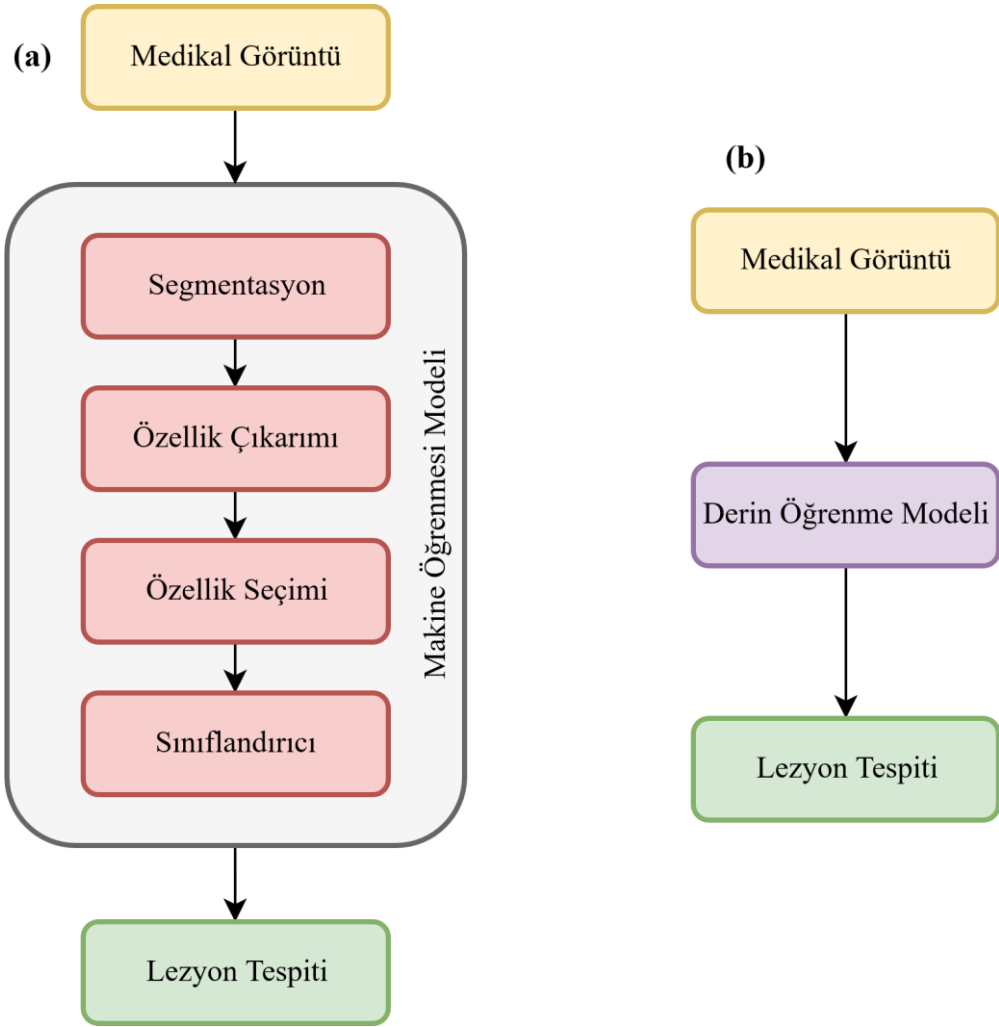
Günümüze dek, çoğu ML ve sinyal işleme tekniği, sıg-yapılı mimariler aracılığıyla birçok problemi başarılı bir şekilde çözebilmişlerdir. Bu yapıdaki mimariler genellikle bir ya da iki katmandan oluşan non-linear bir yapıdadır. Örnek olarak:

- Gauss Karışım Modelleri (GMM),
- Destek Vektör Makineleri (SVM),
- Lojistik Regresyon (Logistic Regression),
- Çekirdek Regresyon (Kernel Regression),
- Aşırı Öğrenme Makineleri (ELM),
- Yapay Sinir Ağları (ANN) verilebilir [21].

Sıg-yapılı mimariler birçok probleme başarılı bir şekilde cevap verse de, daha karmaşık veya gerçek-dünya problemleriyle karşılaştığında (insan konuşması, doğal ses ve dil işleme, doğal görüntü ve görsel vaka vb.), limitli modelleme ve limitli sunum gibi zorluklara maruz kalmaktadır. Araştırmacılar da bu tip problemlerin üstesinden gelebilmek, karmaşık sistemleri daha hızlı ve başarılı bir şekilde çözümlenebilmek adına DL sistemlerine yönelmişlerdir [22,23].

Diğer yandan, DL yöntemlerinin belki de en etkileyici kısmı, öznitelik çıkarımını kendi kendine gerçekleştirebilmesidir.

Standart bir ML algoritmasında (Şekil 3.2-a) objeler (ör: lezyonlar) öncelikle eşikleme, segmentasyon vb. teknikler kullanılarak bir ön-işlem sürecinden geçirilirler. Daha sonra, kontrast, ovalik ve boyut gibi öznitelik çıkarımları sağlanarak ML modeline (ör: Çok Katmanlı Algılayıcı (MLP) veya SVM) giriş olarak verilirler. Giriş verileriyle ve bilinen sınıflarla eğitilen ML modelleri, eğitim sonrasında kendisine verilecek yeni bir vektörde aranılan özelliğin olup olmadığını bulmaya çalışırlar (ör: görüntüde kanser olup olmadığı gibi).



Şekil 3.2. a) Makine öğrenmesi ve b) derin öğrenme.

DL de ise, tüm bu işlemler atlanarak piksel tabanlı bir sınıflandırma gerçekleştirilir (Şekil 3.2-b). Yani, herhangi bir öznitelik çıkarımı yapılmadan sadece görüntünün kendisini giriş verisi olarak alan sistem, eğitim sonunda kendisine gösterilen yeni bir görüntü ile ilgili kestirim yapabilme özelliğini kazanır [7].

3.3. DERİN ÖĞRENME YÖNTEMLERİ

Tarihsel olarak DL yöntemi incelendiğinde, Yapay Sinir Ağları'ndan devşirilmişir denilebilir. Literatürde, *yeni-jenerasyon sinir ağları* olarak adlandırıldığı da görülmüştür. İleri beslemeli sinir ağları ya da birçok gizli katmana sahip MLP de, çoğunlukla Derin Sinir Ağları (DNN) olarak tanımlanmaktadır ve derin-yapılara iyi birer örneklerdir [17].

80'li yıllarda popülerleşen Geri-yayılım Algoritması (BP) bu ağlar için iyi bir öğrenme metodudur. Fakat, tek başına BP algoritması, az sayıdaki gizli katmandan sonrası için yeterince iyi değildir. Çünkü, konvex olmayan problemlerde yaygın yerel optimum noktaları ve diğer optimizasyon problemlerini öğrenmede oldukça zorlanmaktadır. BP algoritmasının lokal minimum noktalarda kalması ve ağın derinliği arttıkça bu problemin iyice hissedilmesi, araştırmacıları kaybın daha az olduğu ve global optimumun verimli bir şekilde bulunduğu, fakat daha zayıf modellenmiş, ML modellerine (SVM vb.) yönelmeye itmiştir [19,22].

2006 yılında gerçekleştirilen konferanslarda, yeni bir danışmansız öğrenme algoritmasının [24] gösterilmesiyle beraber optimizasyon probleminin üstesinden gelinebilmeye başlanmıştır. Kısıtlı Boltzmann Makineleri (RBM)'nin yığını kullanan Derin İnanç Ağları (DBN) sayesinde ağırlıklar, zaman karmaşıklığı ağın genişliğiyle ve derinliğiyle lineer olacak şekilde optimize edilebilmiştir. Böylece; danışmansız bir şekilde eğitilebilen ve BP algoritmasına ince-ayar (fine-tuning) yapılmış DBN sayesinde, DNN literatüre kazandırılmış oldu. Bu modellerin geneline literatürde DNN veya DBN olarak rastlansada, DBN kısmı DNN'nin eğitim süreci olduğundan, DBN-DNN olarak adlandırıldığı da görülmüştür [17].

Diğer yandan, DL'nin gelişim sürecinde, çeşitli eğitim süreçleri (oto-kodlayıcılar, Stokastik Gradyan İniş (SGD vb.) ortaya çıktı ve DNN geliştirilerek oldukça fazla alanda başarılı bir şekilde uygulanmaya başlanmıştır [17].

Bir DL uygulaması yaparken belirli bir problem için hangi modelin seçileceğinin belirlenmesinde, bir kaç soru gözönünde bulundurulmalıdır;

- Üretken (generative) mi yoksa ayırt eden (discriminative) bir model mi kullanılmalı?
- Verinin özellikleri nelerdir?
- Giriş değerlerinin büyüklüğü ne kadardır?

Eğer girdi verisi yeniden yapılandırılmak isteniyorsa üretken bir model tercih edilmelidir. Amaç sınıflandırma ise ayırt edici bir model kullanılması uygun olacaktır. İki yöntemden birini seçerken aynı zamanda simülasyon süresi de göz önünde bulundurulmalıdır. Bu durumda Grafik İşleme Birimi (GPU) ile paralelleştirme, ayrıca, optimizasyon amaçlı olarak kullanılan L-BFGS (Limited-memory BFGS) ve Conjugate Gradient Descent yerine SGD kullanımı önerilebilir. Ayrıca verilerin geçici bir yapıya sahip olması veya resim gibi yüksek boyutlu olması da çalışılacak modelin seçiminde önemli etkenlerdir. Özetle, kullanılacak yöntem eldeki verinin ve problemin tipine göre şekillenmektedir [25].

ML'deki öğrenme yöntemleri gibi, DL yöntemleri de üç başlık altında incelenebilir. Bunlar:

- Danışmansız Derin Ağlar: Bu tiğ ağlar hedefle ilgili bilgi verilmediğinde, elde edilen ya da gözlemlenen veriden en yüksek korelasyonun çıkartılması amacıyla kullanılır.
- Danışmanlı Derin Ağlar: Bu kategorideki yöntemler ise, sınıflandırma amaçlı ve hedef bilgileri gözetilerek ayırım yapabilen algoritmaları içerir.
- Hibrit Derin Ağlar: Bu türdeki algoritmalar ise genellikle danışmansız öğrenme yöntemlerinden elde edilen sonuçların asiste edilmesiyle ortaya çıkmıştır [17].

Danışmansız DL yöntemleri arasında en popülerleri;

- Oto-Kodlayıcılar (Auto-Encoders),
- Derin Boltzmann Makineleri (Deep Boltzmann Machines),
- Kısıtlı Boltzmann Makineleri (Restricted Boltzmann Machines),
- Yinelemeli Sinir Ağları (Recurrent Neural Networks),
- Gizli Markov Modeli (Hidden Markov Model)'dir.

Bu yöntemler özellikle yazı ve konuşma tanımada sıklıkla kullanılır, problemin ve verilerin tipine göre ağların değişikliklere uğramasıyla ortaya çıkmıştır [17].

Danışmanlı öğrenmede ise;

- Koşullu Rastgele Alanlar (Conditional Random Fields),
- Derin İstifleme Ağları (Deep Stacking Network),
- Konvolüsyonel Sinir Ağları (Convolutional Neural Network),
- Yinelemeli Sinir Ağları (Recurrent Neural Networks),
- Gizli Markov Modeli (Hidden Markov Model) gibi yöntemler karşımıza çıkmaktadır.

Yine, bu yöntemler de konuşma, yazı ve fonetik tanıma gibi problemlerde başarılı bir şekilde uygulanmıştır [17,20].

Bu yöntemler *ön-öğrenme* aşamasında kullanılarak ve diğer yöntemlerle birleştirilerek hibrit modeller de elde etmekte mümkündür [21].

Daha önce de bahsedildiği gibi DL bir çok alanda başarılı bir şekilde uygulanmıştır.

Genel olarak uygulama alanlarını;

- Hisse senedi tahmini,
- Görüntü analizi,
- Dil işleme,
- Ses tanıma,
- Hareket tanıma,
- Psikolojik veri analizi olarak sınıflandırmak mümkündür [17,26].

Sonraki bölümde DL’de sıklıkla ve başarılı bir şekilde kullanılan Konvolüsyonel Sinir Ağları (CNN) metotundan detaylıca bahsedilerek bilgisayarlı görü ve DL kavramına giriş yapılacaktır.

3.4. DERİN ÖĞRENME VE BİLGİSAYARLI GÖRÜ

DL ve bilgisayarlı görü konusunu incelediğimizde karşımıza ilk çıkacak model *CNN* olacaktır. Bu tip ağları diğer DL yöntemlerinden ayıran en büyük özelliği, giriş olarak görüntünün kendisini almasıdır. Diğer DL modellerinde (ör: Yığılılı Oto-Kodlayıcı (Stacked Auto-Encoder), DBN, DNN) de giriş vektör formundadır. Fakat, medikal görüntülerde tüm pikseller ve vokseller yapı hakkında önemli bilgiler taşıyabilmektedir. Bu nedenle vektörleştirme bu bilgilerin kaybına neden olacaktır [27]. CNN ise 2B ve 3B görüntüleri direkt giriş verisi olarak kullanır. Klasik bir ileri beslemeli sinir ağının yapacaklarından önce, öznitelik çoğaltma ve özetleme katmanlarını da barındırarak, öznitelik çıkarmayı da içine almaktadır. Bu sayede, görüntünün yapısını daha iyi analiz etmeyi sağlamaktadır.

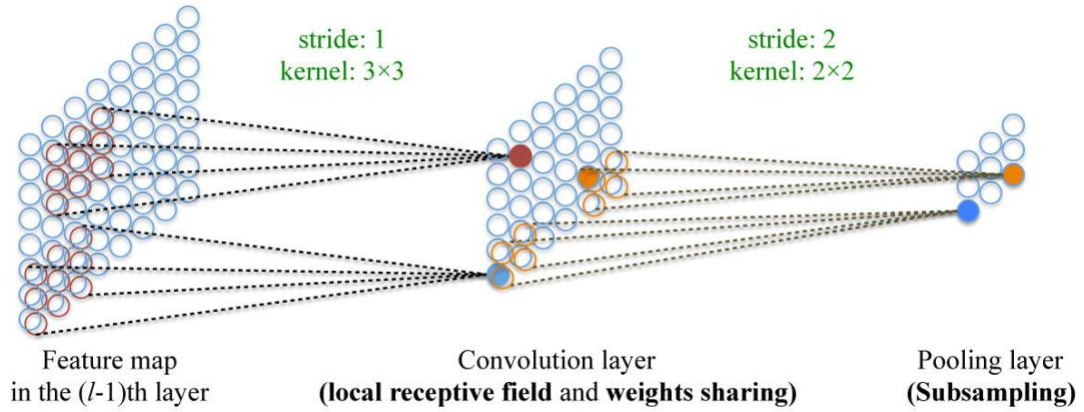
3.4.1. Konvolüsyonel Sinir Ağları (CNN)

Standart bir CNN mimarisi *konvolüsyon (convolution)*, *alt-örnekleme (pooling)* ve *tam bağlı katman (fully-connected layer)* adı verilen bir MLP'den oluşmaktadır. Her bir konvolüsyon filtre eğitim sırasında öğrenilen parametreleri içermektedir. Literatürde kernel, weight matrix gibi tanımlamalarda görülebilir. Bir konvolüsyon katmanının en önemli amacı motif benzeri kenarları, çizgileri vb. görsel elemanları tespit etmektir. Daha sonra alt-örnekleme katmanları ile de görüntünün boyutu azaltılarak, olabilecek en az sayıda parametreyle görüntü temsil edilmeye çalışılır [9,27,28].

CNN'in sağladığı en önemli avantajları üç madde halinde verirsek;

- Örneğin, elimizde $200 \times 200 \times 3$ boyutlarında bir görüntü olduğunu farz edelim. Eğer, bu görüntüyü vektör haline getirirsek, elde edilen vektör tam *120 bin* parametreden oluşacaktır. Eğer ağımızın ilk gizli katmanında *bin* düğüm varsa sadece ilk katmanda eğitilmesi gereken parametre sayısı *120 milyon* olur. Gizli katman sayısı arttıkça, bu sayıda katlanarak artacaktır. Fakat bunun yerine konvolüsyon filtreleme ve alt-örnekleme katmanları kullanarak hem görüntüden daha fazla bilgi çıkarımı sağlayabilir, hem de eğitilecek ağ parametresini ciddi oranda azaltabiliriz.

- CNN'in bir diğere önemli özelliđi de *ađırlık paylaşımıdır (weight sharing)*. Bir filtre bir görüntüdeki her yeri tarayacağından (ör: dikey kenar detektörü), aynı ađırlık deđerlerinin farklı nöronlara bađlanması söz konusu olacaktır (kısmi bađımlılık).
- Son olarak CNN'nin *yerele bađlantı (local connectivity)* özelliđinden bahsedebiliriz. Bir filtreden geçirilen her bir bölüm sonraki katmanda sadece bir nörona bađlanmaktadır. Yani her bir nöron sadece görüntünün bir bölümünden sorumlu olacaktır. Bu da görüntünün diğere piksellerinin bu çıkışı etkilemeyeceđi anlamına gelmektedir (kısmi bađımsızlık) (Şekil 3.3) [27].



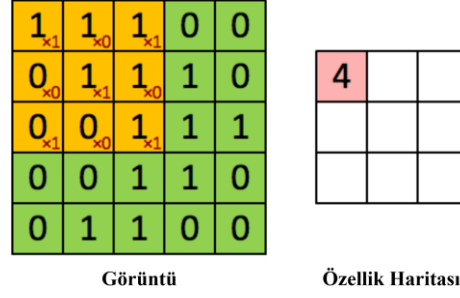
Şekil 3.3. Konvolüsyonel ađların üç önemli mekanizması (yerele bađlantı, ađırlık paylaşımı ve boyut azaltma) [8].

3.4.1.1. Konvolüsyonel Sinir Ađlarının Elemanları

CNN günümüzde görüntü tanıma ve sınıflandırma gibi görevlerde oldukça fazla kullanılan bir yöntemdir. Fakat bu ađların tam olarak nasıl çalıştığını anlamak kimi zaman zor olabilir. Bu bölümde CNN mimarileri daha detaylı açıklamaya çalışılacaktır.


Bir CNN mimarisi veriyi dört aşamada işler. Bunlar sırasıyla;



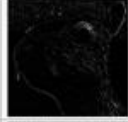




- Konvolüsyon (Convolution)
- Doğrusal-Olmama (Non-Linear)
- Alt-Örnekleme (Pooling)
- Tam Bađlı Katman (Fully Connected Layer)



Şekil 3.5. Konvolüsyon operasyonu ve sonrasında elde edilen özellik haritası [30].

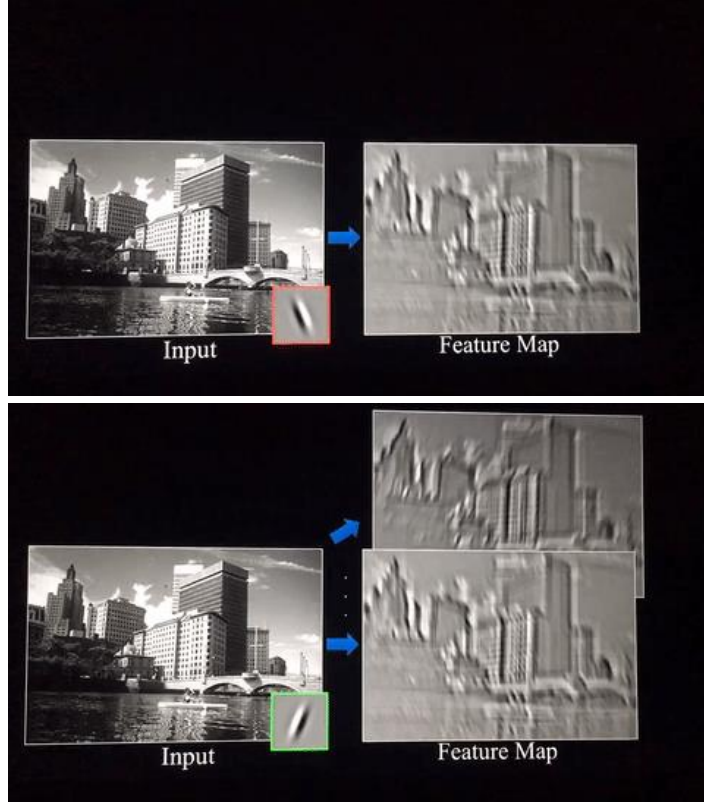
Gerçek dünyadan bir örnek verirse bir deve görüntüsünün, farklı filtrelerle nasıl özellik çıkarımı yapıldığı aşağıdaki şekilde görülebilir (Şekil 3.6).



Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Şekil 3.6. Bir deve görüntüsünün çeşitli filtrelerle özellik çıkarımı [30].

Konvolüsyonel operasyonun nasıl işlediğine dair bir başka örnekte Şekil 3.7’de görülebilir.

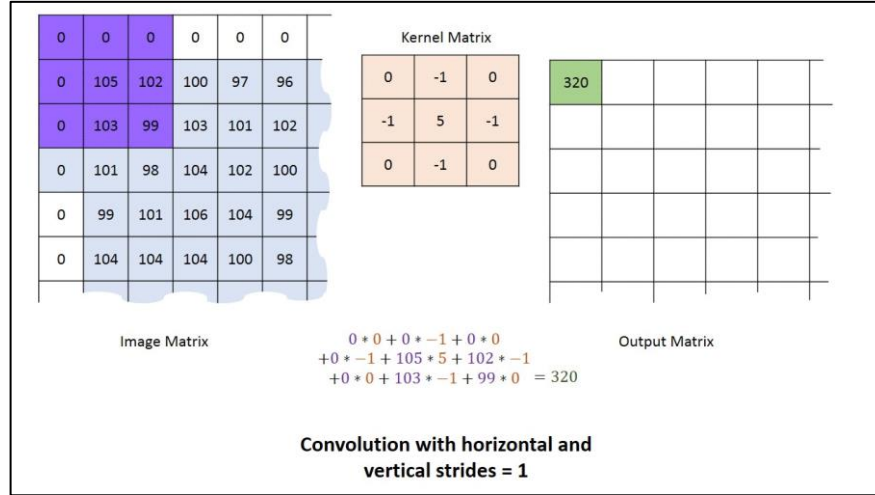


Şekil 3.7. İki farklı filtre ile özellik çıkarımı [31].

Görüldüğü üzere iki farklı filtre görüntü üzerinde gezerken, iki farklı özellik haritası ortaya çıkarmıştır. CNN’nin en güzel yanlarından bir tanesi de bu filtrelerin sahip olduğu değerlerin, eğitim sürecinde ağ tarafından öğrenilmesidir. Yani; ağ en iyi sonuca ulaşabileceği filtreleri kendisi belirleyecektir.

Bir özellik haritası üç parametre tarafından kontrol edilmektedir. Bunlar;

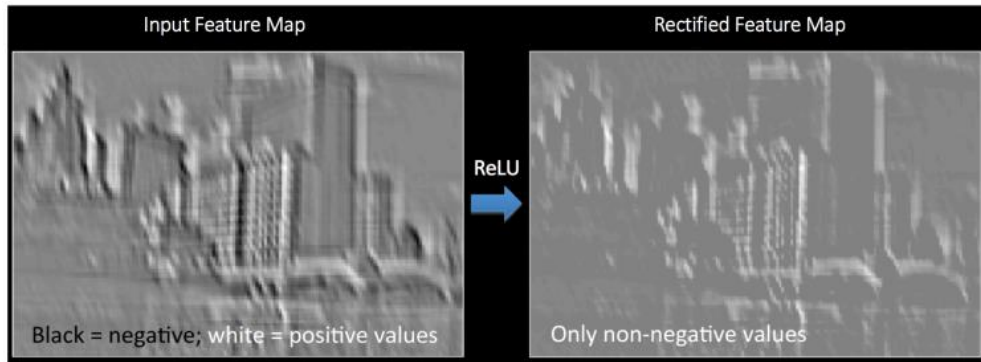
- Derinlik: Bir katmanın kaç filtre içerdiğini belirtir.
- Adım (stride): Görüntü üzerinde ilerleyen filtrenin kaç piksel atlayarak tarama yapacağını gösterir.
- Sıfır-dolgu (zero-padding): Konvolüsyon işlemi boyut azaltmaya neden olduğundan, bunu engellemek için ‘0’ lardan oluşan bir kenar eklenir. Böylece kenarlardaki önemli niceliklerin de ağ tarafından öğrenilebilmesi sağlanır (Şekil 3.8).



Şekil 3.8. Stride ve zero-paddding [32].

Doğrusal-Olmama (Non-Linear)

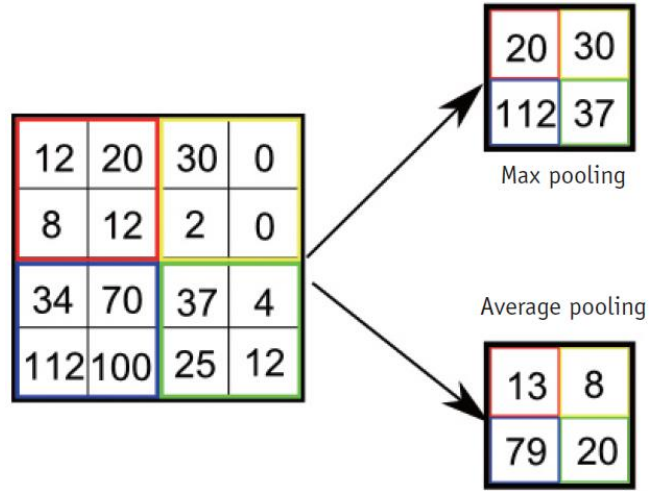
Bilindiği üzere aktivasyon fonksiyonları giriş değerleri ve ağırlıkların çarpımıyla gelen değerlerin toplamını, belirli bir aralıkta ifade etmeye yarar. Örneğin; sigmoid fonksiyonu sonuç ne olursa olsun ilgili değeri $[0,1]$ aralığında normalize eder. Böylece bir nöronun ürettiği çıkış değeri elde edilmiş olur. Fakat, sigmoid fonksiyonu türevi alınıp güncellemeler yapıldıkça, 0'a doğru çok hızlı bir şekilde yaklaşır (*vanishing gradient problem*). Bu da ağıın ürettiği sonuçların giderek kötüleşmesi anlamına gelir. DL modellerinde bu problemin üstesinden gelebilmek adına *ReLU* aktivasyon fonksiyonu üretilmiştir [33]. Bu fonksiyon sadece '0' veya pozitif bir değer üretir $g(z) = \max(0, x)$. Eğer sonuç negatif ise değerleri '0' olarak, pozitif ise olduğu gibi geçirir. ReLU işleminden geçirilmiş bir özellik haritası aşağıdaki gibidir (Şekil 3.9).



Şekil 3.9. ReLU operasyon [34].

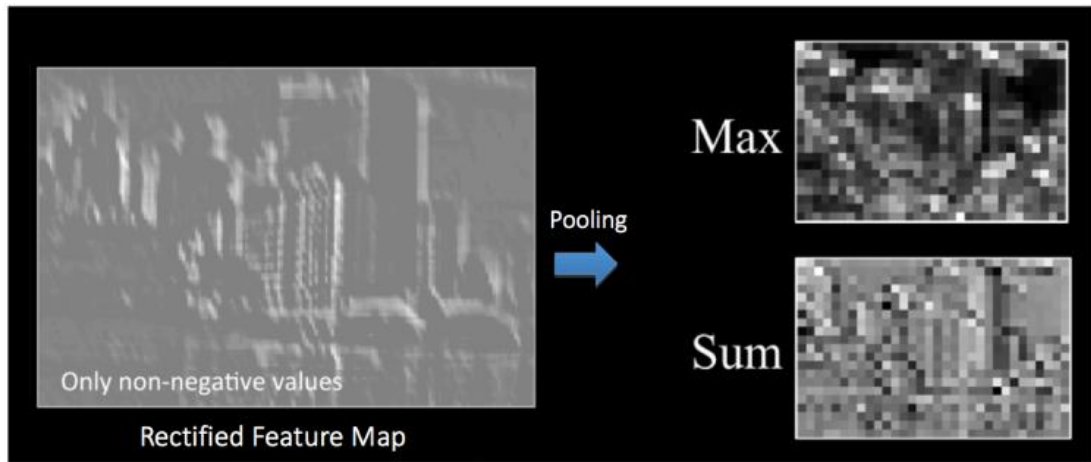
Alt-Örnekleme (Pooling)

Alt-örnekleme katmanı her bir aktivasyon haritasının (özellik haritası) boyutunu azaltmak için kullanılır. Böylece en değerli veriler alınarak, nispeten daha az parametre ile eğitim yapılmasını mümkün kılar, bu da eğitim verilerinin ezberlenmesini engeller (Şekil 3.10).



Şekil 3.10. Alt-örnekleme operasyonu [9].

Alt-örnekleme işlemi uygulandıktan sonra aktivasyon haritası şu şekilde gözükecektir (Şekil 3.11).

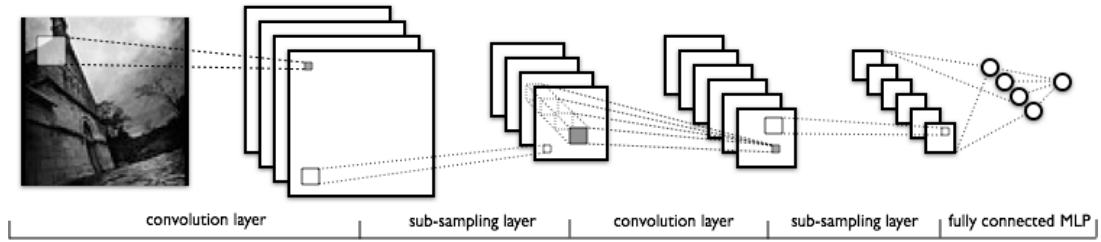


Şekil 3.11. Alt-örnekleme işlemi sonrası özellik haritası [34].

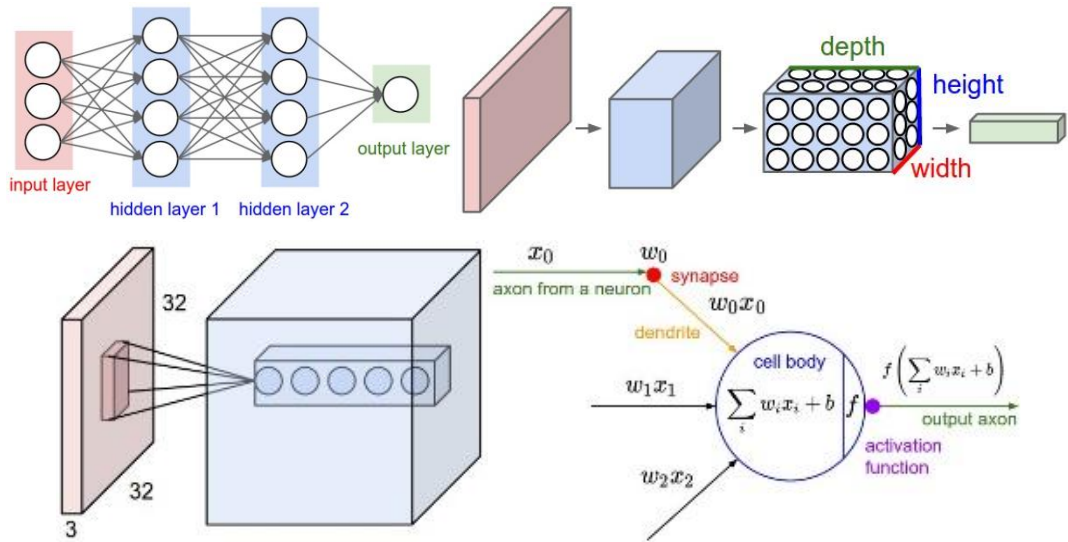
Tam Bağlı Katman

Bu katman ise tam anlamıyla bir MLP olarak çalışmaktadır. Son konvolüsyonel katmanın sahip olduğu değerler vektörleştirilerek, istenilen sayıda gizli katmana ve nörona bağlanır. Konvolüsyonel ağın en sonunda ise sınıflandırma işleminin üretmesi gereken sınıf sayısı kadar nörondan oluşan bir *softmax* katmanı bulunur bu katman, giriş değeri olarak verilen görüntü için ilgili olasılık değerlerini hesaplar.

Tüm bu nicelikleri bir arada bulandıran örnek bir ağ şeklindeki gibidir (Şekil 3.12). Bir konvolüsyonel ağ giriş olarak üç boyutlu olarak aldığı görüntüyü, tüm nöronları üç boyutlu olarak sıralayarak (genişlik, yükseklik, derinlik) bir katmanda yer alacak hale getirir (Şekil 3.13).



Şekil 3.12. Bir CNN örneği [35].

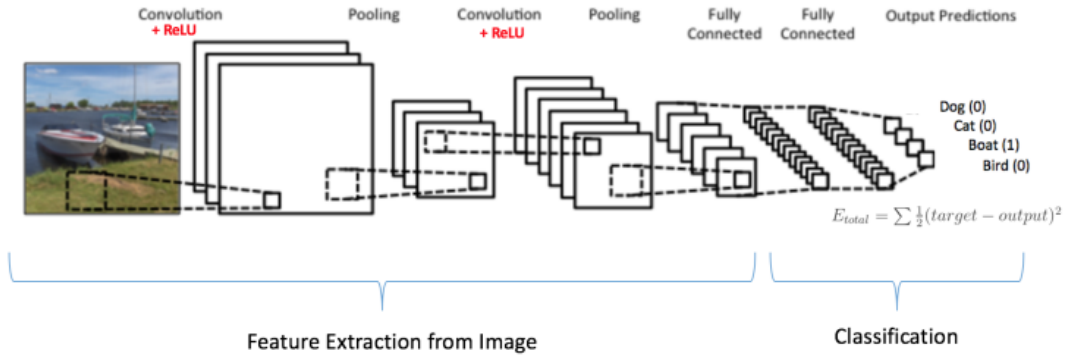


Şekil 3.13. Konvolüsyon katmanının detaylı gösterimi [36].

3.4.1.2. Konvolüsyonel Ağların Eğitim Süreci

Daha önce bahsedildiği üzere konvolüsyon ve alt-örnekleme katmanları görüntüden özellik çıkarımı amacıyla, tam bağlı katmanlar ise sınıflandırma amacıyla kullanılmaktadır. Örneğin, bir CNN mimarisine giriş olarak verilen bir *bot* görüntüsü için ağdan bir tahmin değeri üretmesini istediğimizde, ağımızın sınıflandırma katmanının aşağıdaki gibi bir tahminde bulunduğunu varsayalım (Şekil 3.14).

- Giriş görüntüsü: *Bot*
- Hedef Vektör: [0, 0, 1, 0]



Şekil 3.14. Örnek bir sınıflandırıcı [30].

O zaman ağın gerçekleştireceği eğitim süreci aşağıdaki gibi olacaktır.

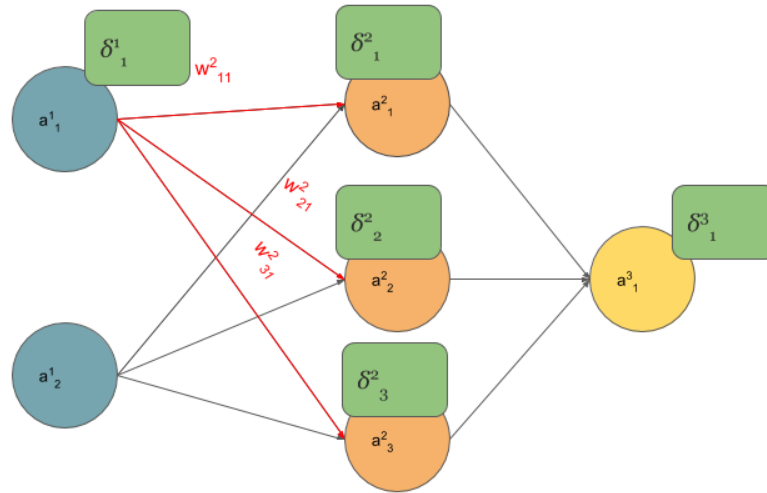
- Tüm filtrelerin sahip olduğu parametrelere (ağırlıklar), rastgele ilk değerleri verilir.
- Ağ görüntüyü işleyerek ileri beslemeyi gerçekleştirir ve sonucunda her sınıf için bir olasılık değeri üretir.
 - Örneğin, bot için [0.2, 0.4, 0.1, 0.3] değerlerini üretmiş olsun.
 - Ağırlıklar rastgele olduğundan ilk görüntü için oluşturulacak tahmin değerleri de rastgele olacaktır.
- Toplam hata hesaplanır (tüm sınıflar için üretilen değerlerin toplanması ile elde edilir).

$$E = \frac{1}{2} \sum_p (t^p - y^p)^2 \quad (3.1)$$

- Geri yayılım algoritması ile hatanın ağırlıklara göre türevleri hesaplanır ve kullanılan optimizasyon algoritmasına göre (ör; SGD) ağırlıklar ve diğer parametreler güncellenir.
 - Ağırlıklar toplam hataya yaptığı katkı oranında güncellenmektedir.
 - Örneğin yeni sınıflandırma vektörümüz $[0.1, 0.1, 0.7, 0.1]$ olursa bunun hedef vektöre $[0, 0, 1, 0]$ oldukça yakın olduğu yani ağıın sınıflandırmayı başardığı kabul edilir.
 - Filtre sayısı, filtre boyutu, ağıın mimarisi ağıın eğitiminin başında belirlenir ve eğitim sırasında değişmez. Güncellenen nicelikler filtre matrisi ve bağlantı ağırlıklarıdır.
- Tüm görüntüler için 2 ve 4 arasındaki işlemler tekrar edilir [30].

Gerı Yayılım Süreci

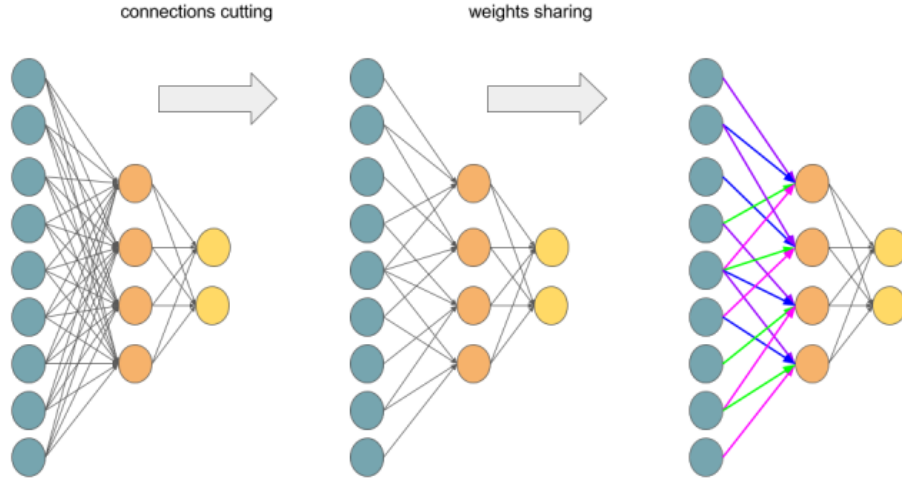
Konvolüsyonel sinir ağılarında gerçekleşen geri yayılım işlemi MLP'lerdeki sürecin neredeyse aynıdır.



Şekil 3.15. MLP de geri yayılım [37].

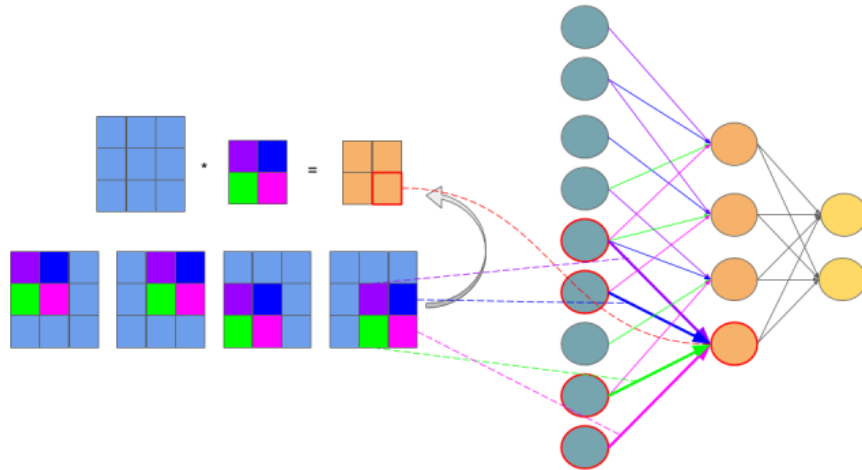
Bilindiği üzere, bir MLP'de ağırlıklar güncellenirken aktivasyon fonksiyonu olarak seçilen fonksiyonun türevi kullanılarak, genel hatanın çıkış değerine göre türevi alınır. Daha sonra bu *deltalar* (δ) kullanılarak ağıın giriş katmanına kadar tüm ağırlıkların güncellenmesi sağlanır (Şekil 3.15).

Konvolüsyonel ağlarda yapı olarak MLP'lere benzerler, fakat CNN'de tüm nöronlar bir sonraki katmana tam bağlı değildir. Kullanılan filtrelerden dolayı, belli nöronlara bağlı ağırlıkların, belli nöronlara bağlanması söz konusudur (weight sharing, local connectivity) (Şekil 3.16).



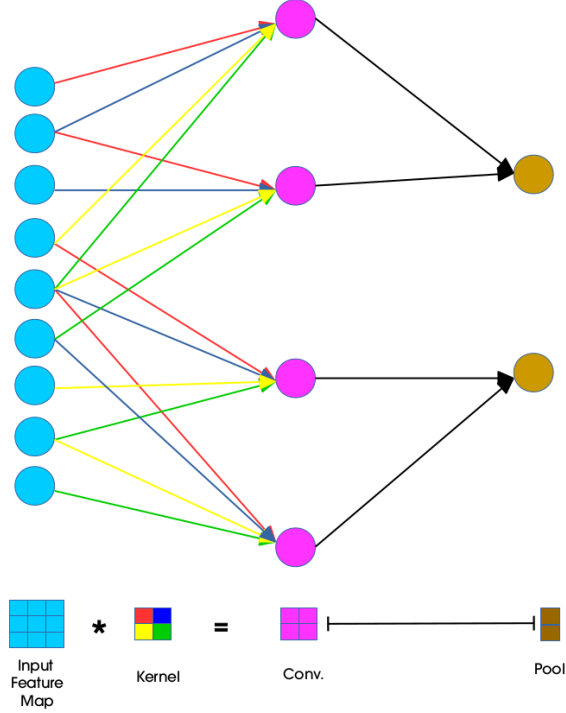
Şekil 3.16. Standart bir MLP ve CNN [37].

Şekilde görüldüğü üzere CNN ağlarında bulunan *parametre paylaşımı ve yerel bağlantı* özelliklerinden dolayı, *dokuz* nörondan oluşan katmandan, bir sonraki katmanda bulunan nöronlara *dokuz* değil *dört* ağırlık bağlanmıştır. Burada ağırlık olarak tanımladığımız nicelikler filtre değerleridir. Filtre değerlerine göre ileri besleme sürecinde ağırlık bağlantılarının oluşumu bir sonraki şekilde gibidir (Şekil 3.17).



Şekil 3.17. CNN de ileri besleme süreci [37].

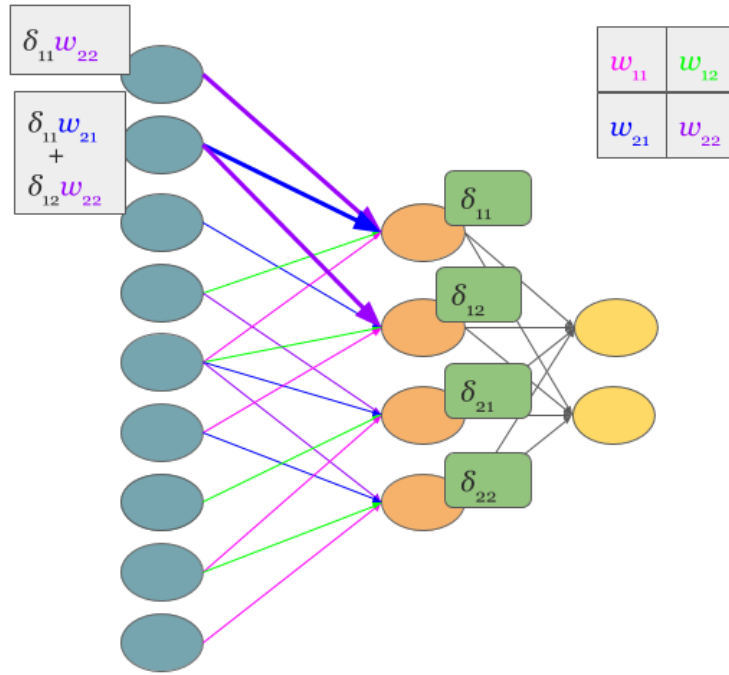
Yukarıdaki örnekte, konvolüsyon işleminden sonra elde edilen ikinci katman çıkış katmanına tam bağlanmıştır. Konvolüsyon işleminden sonra alt-örnekleme katmanına bağlanan bir model içinse aşağıdaki şekil incelenebilir (Şekil 3.18).



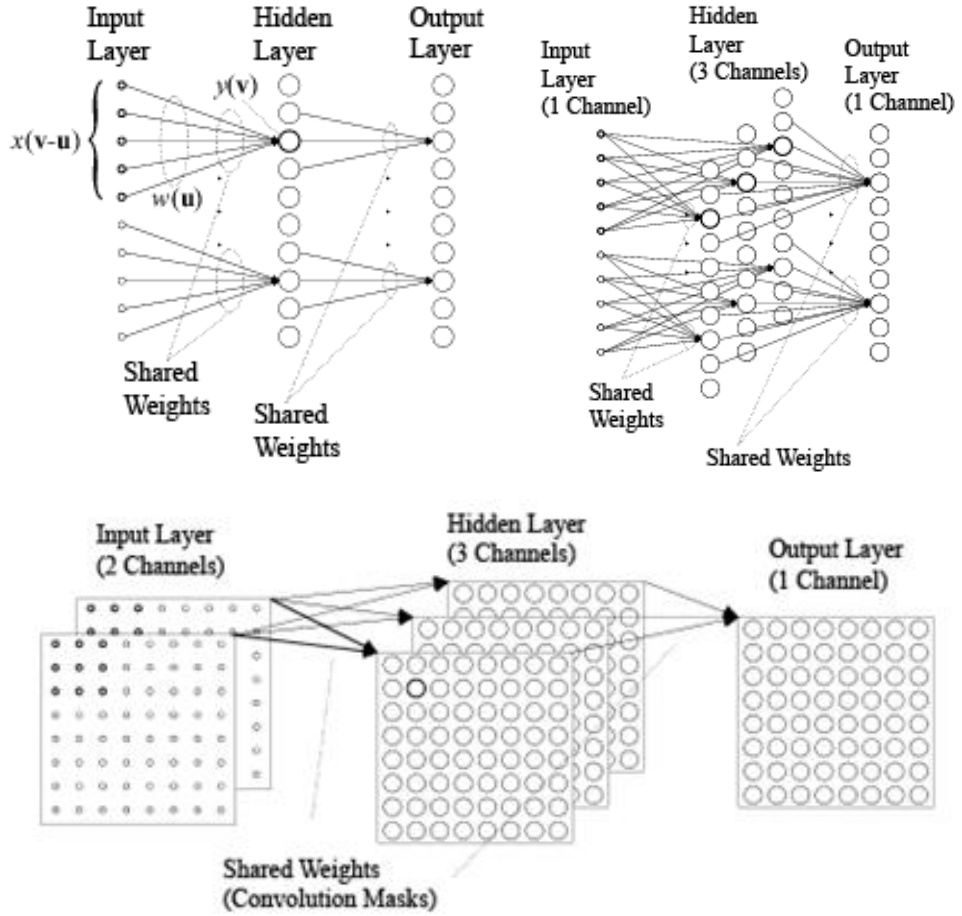
Şekil 3.18. Bir konvolüsyon ve alt-örnekleme katmanının MLP olarak sunumu [38].

Konvolüsyon işleminden sonra elde edilen özellik haritasında 1×2 maksimum alt-örnekleme işlemi gerçekleştirildiğinde, ilgili iki nöronun elde edilen en yüksek değer seçilecek ve sonuç olarak 3×3 boyutundaki görüntü 2×1 boyutunda ifade edilebilecektir. Tıpkı, MLP gibi, ağda geri yayılım yaparken hatayı direkt olarak ağırlıklar üzerine yayarak işlem yapmak mümkün değildir. Bunun için *zincir kuralı* kullanılarak üretilen çıkışların toplam hatası hesaplanır, hesaplanan türevlerden faydalanılarak hata adım adım geriye doğru yayılır.

Hata geriye yayılırken bir önceki katmanın çıkış değerleri de sadece ilgili ağırlıkların değerleri kullanılarak güncellenecektir (Şekil 3.19). Giriş olarak verilen görüntü birden fazla kanallı ise, ya da birden fazla filtre kullanıyorsak o zaman gizli katman da birden fazla boyuttan oluşacaktır. Bunun MLP olarak gösterilişi aşağıdaki gibidir (Şekil 3.20).



Şekil 3.19. CNN de hatanın geriye yayılması süreci [37].

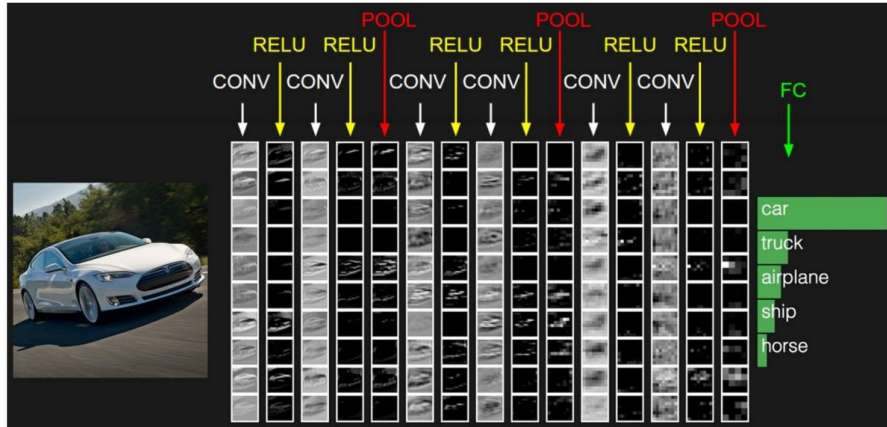


Şekil 3.20. Çok boyutlu CNN katmanı [39].

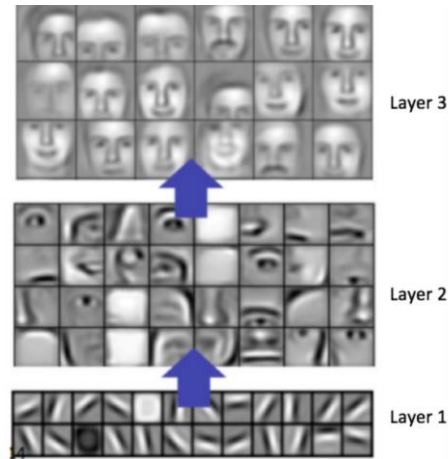
Tamamlanan eğitim sonrasında, modele yeni (daha önce görmediği) bir görüntü gösterildiğinde, görüntü ağı üzerinde bir ileri besleme sürecinden geçecek ve her bir sınıf için bir sonuç üretilacaktır.

3.4.1.3. Konvolüsyon Ağ Katmanlarının Görselleştirilmesi

Genel olarak, “katman sayımız ne kadar fazla ise öğrenilecek özelliklerin sayısı da o ölçüde artmaktadır” denilebilir. Mesela CNN, bir görüntü sınıflandırmada, ilk katmanda ham veriden kenar özelliklerini öğrenirken, sonraki katmanda bu kenar özelliklerini kullanarak şekilleri, daha sonra da bu şekillerden daha karmaşık özelliklerin çıkarımlarını yapabilmektedirler. Aşağıdaki iki örnekte bir araba ve insan yüzünün konvolüsyonel ağlar ile sunumu görülebilir (Şekil 3.21, Şekil 3.22).

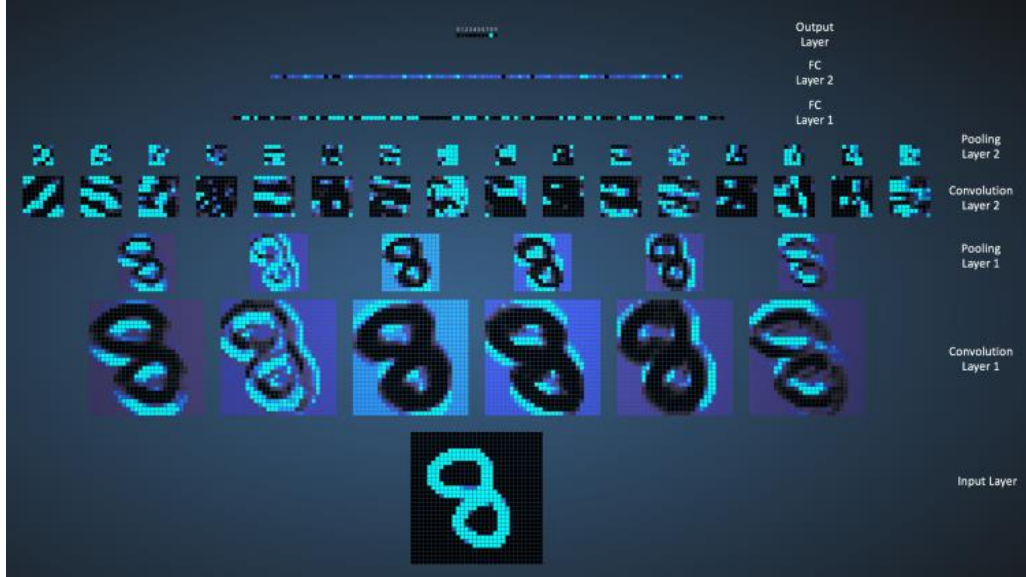


Şekil 3.21. Bir aracın CNN ile özellik çıkarımı [36].



Şekil 3.22. Bir insan yüzünün CNN ile özellik çıkarımı [40].

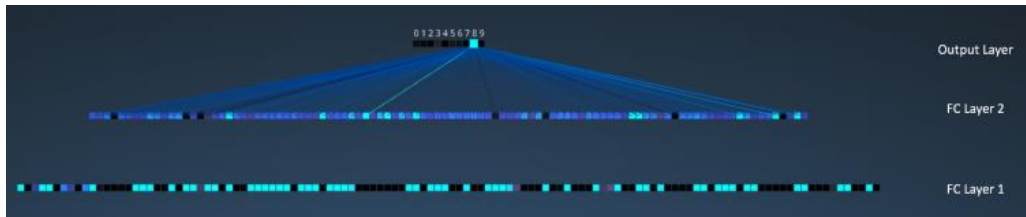
Adam Harley tarafından MNIST [41] veri seti üzerinde gerçekleştirilen ve ‘‘CNN özellik haritalarının nasıl oluřtuđu’’ nu gosteren gorsel sunumu daha yakından incelersek, CNN’in her bir katmanda ogrenmeyi nasıl gerekleřtirdiđini daha net anlamamız mumkun olabilir (řekil 3.23).



(a)



(b)



(c)

řekil 3.23. MNIST veri setinin CNN katmanlarında gorselleřtirilmesi, a) Tum ađđ, b) konvolusyon ve alt-ornekleme, c) tam bađđlı katman [42].

řekilde 8 rakamının CNN katmanları ile ogrenilme sureci gorulmektedir. 32×32 boyutlarındaki gri-tonlanmış goruntu 5×5 boyutunda *altı* tane filtre ile taranarak *altı*

farklı aktivasyon haritası oluşturulmuştur. Her bir filtrenin farklı bir özellik haritası ortaya çıkarttığı görülebilir. Daha sonra 2×2 boyutundaki filtreler ile alt-örnekleme işlemi gerçekleştirilerek, *altı* filtre üzerindeki verilerden en parlak olanlar, yani; maksimum değerler alınarak boyut azaltma gerçekleştirilmiştir.

Aynı işlemler bir kez daha gerçekleştirilerek bu sefer görüntünün daha karmaşık kısımları ortaya çıkarılmıştır. En sonunda vektörleştirme işlemi gerçekleştirilmiş ve de 8 sayısını ifade eden vektör sunumu elde edilmiştir.

120 ve 100 düğümden oluşan iki katmandaki aktivasyon değerlerinin ortaya çıkaracağı sonuç sayesinde, giriş değerinin [0–9] arasındaki hangi rakama karşılık geleceği hakkında tahmin yapılmaktadır.

3.4.2. Diğer Yöntemler

CNN’i özel kılan mimaride yer alan konvolüsyon operasyonudur. Bu operasyon sayesinde verilerdeki anlamlı ifadeler eğitilecek parametre sayısı azaltılarak çıkarılabilmektedir. Ağırlık paylaşımı ve yerel bağlantı gibi özellikleri sayesinde veriyi hiyerarşik olarak sunabilme yeteneğine sahiptirler. Bu gibi avantajları sayesinde konvolüsyonel operasyonları barındıran ama farklı amaçlar için kullanılan (nesne tespiti, segmentasyon) modeller elde etmek mümkün olmuştur (ör: R-CNN (Bölge tabanlı-CNN) ve FCNN (Tam-CNN)) [28]. Diğer yandan literatürde farklı sinir ağı mimarilerini de gözlemlemek mümkündür (ör: ResNET (Artık Ağlar)) [43].

DL alanında sıklıkla kullanılan yöntemlerden bir tanesi de transfer öğrenmedir. Fakat bu konu tezin *beşinci bölümünde* derinlemesine ele alınacaktır.

Bir sonraki bölümde, CNN yöntemindeki sorunlar ve yeni bir DL yöntemi olan *Kapsül Ağların*ın bu problemlerin üstesinden nasıl gelebildiği anlatılacaktır. Aynı zamanda bir sonraki bölüm WBC sınıflandırılmasında kullanılacak modeli, elde edilen sonuçları, diğer yöntemlerle kıyaslamaları ve çalışmadan elde edilen kazanımları da içermektedir.

BÖLÜM 4

BEYAZ KAN HÜCRELERİNİN KAPSÜL AĞLARI İLE SINIFLANDIRILMASI

Daha önce de bahsedildiği üzere WBC alt-tiplerinin doğru ve hızlı bir şekilde sınıflandırılması, kişinin sahip olduğu veya olabileceği kan hastalıklarının tespitinde oldukça büyük bir öneme sahiptir. Bu sebepten dolayı literatürde konu ile ilgili birçok çalışma gözlemek mevcuttur. Bu çalışmalar sınıflandırma aşamasında kullanılan yöntemlere göre *klasik makine öğrenmesi* ve *derin öğrenme* olarak iki başlık altında incelenecektir.

4.1. BEYAZ KAN HÜCRELERİNİN SINIFLANDIRILMASI ALANINDA YAPILMIŞ ÇALIŞMALAR

4.1.1. Klasik Makine Öğrenmesi Yöntemleri Kullanılarak Gerçekleştirilen Çalışmalar

Bu bölümde, 2004–2014 yılları arasında, ML yöntemleri kullanılarak gerçekleştirilen çalışmalardan bazılarını gözlemek mümkündür. Çalışmalar, analiz sırasında kullanılan yöntemler ve çalışma sonunda elde edilen kazanımlar açısından değerlendirilmiştir.

2004 yılında V. Piuri ve F. Scotti, hücrelerin morfolojik özelliklerini kullanarak özellik çıkarımı yapmışlar, sınıflandırma aşamasında ise k - En Yakın Komşu (KNN) ve ANN yöntemleri ile 113 görüntüden oluşan bir WBC veri seti üzerinde testler gerçekleştirmişlerdir. Çalışmalarının sonunda ANN yönteminin oldukça yüksek başarı değerine ulaştığını göstermişlerdir [44].

2005 yılında H. Ramoser vd. WBC'lerin tespiti ve sınıflandırılması adına tam otomatik bir model önermişlerdir. Görüntüler üzerinde çeşitli teknikler ile hücrelerin

segmentasyonunu gerçekleştiren yazarlar, görüntülerden elde ettikleri çıkarımları kullanarak, SVM ile sınıflandırma gerçekleştirmişlerdir. 1116 görüntü üzerinde %95 segmentasyon başarıları gözlemleyen yazarlar, sınıflandırma başarısının ise %75–%99 arasında değişen değerlere ulaştığını belirtmişlerdir [45].

N. Theera-Umpun yaptığı çalışmada Fuzzy C-means algoritması ve matematiksel morfoloji ile hücreleri segmente eden bir model önermiştir. Otomatik olarak segmentasyonu gerçekleştirilen hücreleri daha sonra ANN ile sınıflandıran yazar, eğitim verileri üzerindeki başarıyı %70,74 test verileri üzerindeki başarıyı ise %65,59 olarak rapor etmiştir [46].

2006 yılında S. Mrčić ve N. Jorgovanović WBC'lerin yüksek doğrulukla sınıflandırılabilmesi adına, sahip oldukları veri setindeki görüntüleri öncelikle RGB renk formatından gri seviyesine indirgemişler, daha sonra ise görüntülerdeki boyama prosedüründen kaynaklanan gürültüleri temizlemişlerdir. Threshold (eşikleme) yöntemi sayesinde hücrelerin belirginleşmesini sağlayan yazarlar, Gaussian kernel ile hücrelerin genel görüntüden ayrıştırılmasını sağlamışlardır. Elde edilen görüntülerde morfolojik analizler gerçekleştirilerek elde edilen özellikler ile ANN eğiten yazarlar, sınıflandırma sonunda elde ettikleri başarı oranının %86'nın üzerinde olduğunu göstermişlerdir [47].

2010 yılında ise, J. M. Ramirez-Cortes vd. gerçekleştirdikleri çalışmada hücrelerin morfolojik özelliklerini kullanarak özellik vektörleri elde etmişlerdir. Daha sonra bu özellik vektörleri ile dört farklı sınıflandırıcı eğitmişlerdir: KNN, ANN, SVM ve MLP. Testler sonrasında en yüksek başarının MLP tarafından sağlandığını belirtmişlerdir (%87,6) [48].

E. Xie vd. hücrelerin segmente edilmesi adına Poisson kısmi diferansiyel denklemini kullanmışlardır. Daha sonra hücrelerin özelliklerini çıkaran araştırmacılar, yaptıkları sınıflandırmada %75,5 ve %98,48 arasında değişen doğruluk başarılarına ulaşmışlardır [49].

2011'de B. C. Ko vd. Gradient Vector Flow yöntemi ile tespitini gerçekleştirdikleri WBC hücrelerinin şekil, renk ve yapı gibi özelliklerini kullanarak 71 boyutlu bir özellik vektörü elde etmişlerdir. Eğittikleri Rastgele Orman (RF) sınıflandırıcısından elde ettikleri sonuçları Çok Kkatmanlı SVM ile kıyaslayan araştırmacılar, RF'nin bu konuda çok daha başarılı olduğunu göstermişlerdir [50].

S. H. Rezatofghi ve H. Soltanian-Zadeh periferik yayma görüntülerinde yer alan WBC'lerin tanınmasına yönelik yaptıkları çalışmanın birinci aşamasında hücrelerin Gram-Schmidt ortogonalizasyonu ile segmentasyonunu gerçekleştirmiş, sonraki aşamada ise morfolojik ve yapısal özelliklerini kullanarak özellik çıkarımı yapmışlardır. Son aşamada ise elde edilen özellik vektörlerini kullanarak önerilen yöntemin SVM ve ANN üzerindeki başarısını sorgulamışlardır. Çalışma sonunda SVM'nin daha iyi sonuçlar ürettiğini görmüşlerdir [51].

2012 yılında N. Ramesh vd. hücrelerin segmentasyonu için, hücrelerin renk ve morfolojik özellikleri kullanan bir yöntem önermişlerdir. Yöntemlerini 1938 görüntü üzerinde test eden yazarlar, bunlardan 1804 tanesini doğru bir şekilde ayrıştırabildiklerini belirtmişlerdir. Segmentasyon sonrasında elde ettikleri özellik çıkarımlarını kullanarak Linear Discriminant Analysis (Lineer Diskriminant Analiz) gerçekleştiren yazarlar %93,9 toplam doğruluğa ulaşmışlardır [52].

J. V. Stadelman ve I. N. Spiridonov çalışmalarında hücrelerin görüntüden ayrıştırılmasını sağlamak amacıyla, öncelikle örneklerin renk uzayını HSV (Hue, Saturation, Value)'ye çevirmişler, daha sonra, bir kenar tespit algoritması ve morfolojik işlemler ile özellik çıkarımı yapmışlardır. AdaBoost algoritması ile sınıflandırma yapan modelin %90 üzerinde başarı yakaladığını göstermişlerdir [53].

2013 yılında M. D. Joshi vd. Akut Lösemi tespiti için kullanılabilecek bir model önermişlerdir, Hücre tespiti aşamasında Otsu otomatik eşikleme ve görüntü zenginleştirme yöntemlerini kullanmışlardır. Sınıflandırma aşamasında ise KNN kullanan yazarlar 108 veri üzerinde %93 doğruluk tespit etmişlerdir [54].

2014'te O. Sarrafzadeh vd. Fuzzy C-Means algoritmasını da içeren çeşitli tekniklerle hücrelerin çekirdeklerinin arkaplandan ayrıştırılmasını sağlamış, sonrasında, hücrelerin renk, geometrik ve yapısal özelliklerini barındıran bir vektör oluşturmuşlardır. Oluşturulan vektör ile dört farklı SVM eğitilmiş (her sınıf için bir SVM), modelin toplam başarısı %95,95 olarak ölçülmüştür [55].

S. Nazlibilek vd. hücrelerin otomatik olarak segmentasyonunu, sayılmasını, boyut tespitini ve sınıflandırmasını yapan bir model önermişlerdir. Segmentasyon işleminin akabinde, MLP kullanılarak sınıflandırma gerçekleştirmişlerdir. Temel Bileşenler Analizi (PCA) kullanılarak giriş sayısı azaltılmış vektör ile eğittikleri MLP'nin, PCA kullanılmadan eğitilen MLP'ye nazaran daha başarılı olduğunu göstermişlerdir [56].

Yine 2014 yılında, M.-C. Su vd. WBC'lerin sınıflandırılması adına kullanılabilecek yeni bir segmentasyon algoritması önermişlerdir. HSI (Hue, Saturation, Intensity) renk uzayına çevrilmiş görüntülerde ayırıcı bölgelerin tespitini yapan algoritma, bu bölgelerde morfolojik işlemler gerçekleştirmektedir. Segmentasyonu sağlanan hücrelerden geometrik, renk ve doku özellikleri çıkartılmıştır. Çalışma sonunda bu özellikler ile eğitilen MLP, 450 örnek üzerinde %99,11 doğruluğa ulaşmıştır [57].

ML yöntemleri kullanılarak gerçekleştirilen çalışmalar incelendiğinde gözlemlenebilecek ortak özellik, araştırmacıların verilerini öncelikle bir ön-işlem sürecinden geçirmiş olmalarıdır. Çalışmalar sonunda yüksek başarı oranları elde edilmiş olsa dahi, bu sürecin farklı özelliklere sahip veri setleri içinde tekrarlanması gerektiği unutulmamalıdır. Diğer yandan ön-işlem sürecinde yapılacak hataların da sınıflandırma başarısını direkt etkileyebileceği düşünüldüğünde, ön-işlem sürecine ihtiyaç duymadan medikal görüntülerin analizini gerçekleştirebilecek yöntemler bir adım öne çıkmaktadır.

Özellikle geçtiğimiz son on yılda oldukça fazla çalışılan bir konu olduğundan, literatürde DL yöntemleri ile WBC alt-tiplerinin sınıflandırılması konusunda gerçekleştirilmiş yayınlar görmek mümkündür.

4.1.2. Derin Öğrenme Yöntemleri Kullanılarak Gerçekleştirilen Çalışmalar

Bu bölümde 2013–2018 yılları arasında gerçekleştirilmiş çalışmalar verilmiştir. Çalışmalarda kullanılan yöntemler, veri seti boyutları, sınıflandırma aşamasından önce gerçekleştirilen ön-işlem süreci (varsa) ve ulaşılan başarı oranları incelenmiştir.

2013 yılında M. Habibzadeh vd. düşük çözünürlükteki WBC'lerin tespiti ve sınıflandırılması için bir model önermişlerdir. Veri setlerinde yer alan 140 görüntünün önce segmentasyonunu gerçekleştiren yazarlar, bu görüntülerin 115'i ile SVM, kernel-PCA SVM ve CNN eğitmişlerdir. Sınıflandırıcı modellerini 25 örnek üzerinde test etmişler ve en yüksek başarıyı CNN'de yakaladıklarını belirtmişlerdir [58].

2017'de X. Li vd. CNN kullanarak hiperspektral görüntülerde kan hücresi sınıflandırması gerçekleştirmişlerdir. PCA ile öznelik çıkarımı gerçekleştirmişler ve daha sonra 8 katmanlı bir mimari kullanarak elde ettikleri sonuçları eğitilen bir SVM ile kıyaslamışlardır. CNN'nin SVM'ye göre %30 oranında daha başarılı olduğunu göstermişlerdir [59].

D. Mundhra vd. preferik yayma görüntülerindeki WBC, kırmızı kan hücresi (RBC) ve plateletlerin otomatik olarak segmentasyonunu ve sınıflandırmasını gerçekleştiren bir yöntem önermişlerdir. Öncelikle veri arttırımı teknikleri sayesinde örnek sayılarını arttıran yazarlar, önerdikleri yöntem ile WBC çıkarımını %99,5 doğrulukla sağlayabilmişlerdir [60].

W. Yu vd. çalışmalarında transfer öğrenme ve ince-ayar (fine-tuning) yöntemlerini kullanarak WBC alt-tiplerinin sınıflandırmasını gerçekleştirmişlerdir. Çalışmalarının sonunda, klasik yöntemlere ve standart bir CNN'ye göre daha iyi sonuçlar elde etmişlerdir [61].

Yine 2017 yılında J. W. Choi vd. tarafından yapılan çalışmada, mevcut 2174 görüntüyü veri arttırma teknikleri ile 48000 seviyesine çıkarılmıştır. Renk kanallarında gerçekleştirdikleri bazı dönüşüm ve normalizasyon işlemlerinin akabinde oluşturdukları CNN mimarisinin %95,68 doğruluğa ulaştığını belirtmişlerdir.

A. I. Shahin vd. 2551 görüntü kullandıkları çalışmada iki adımlı bir sınıflandırma modeli önermişlerdir. Öncelikle veri setini bir ön-işleme sürecinden geçirmişler, sonra ise transfer öğrenme yöntemi ile özellik çıkarımı gerçekleştirmişlerdir. Son olarak, elde ettikleri özellik vektörleri ile bir SVM eğitmişlerdir. Başarı oranlarını standart bir CNN ile kıyaslayan yazarlar, önerdikleri modelin daha iyi sonuçlar ürettiğini göstermişlerdir [62].

J. Zhao vd. öncelikle görüntülerin renk kanallarını gri tonlamalı olarak dönüştürmüşler, daha sonra ise gürültüleri temizlemek adına bir ön-işlem süreci kullanmışlardır. Hücrelerden çıkardıkları özellikler ile bir SVM eğitilmiş bu sayede bazofil, eozinofil ve diğer hücreler olarak üç sınıflı bir ayrıştırma sağlamışlardır (bazofil ve eozinofil örnek sayıları yetersiz olduğundan dolayı). Akabinde CNN ile diğer hücrelerin özellik çıkarımı yapılmış ve RF algoritması aracılığıyla nötrofil, lenfosit ve monosit sınıflandırılması gerçekleştirmişlerdir. Çalışmaların sonunda %70 ile %100 arasında değişen sınıflandırma başarılarına ulaşmışlardır [63].

2018'de M. Habibzadeh vd. dört WBC alt-tipi için bir sınıflandırma gerçekleştirmişlerdir. Örneklerini ön-işlem (veri artırma, ren kanalı dönüşümleri, vs.) sürecinden geçiren araştırmacılar, 11200 görüntüyü kullanarak ResNet ve Inception [64] mimarileri eğitmişler, 1244 görüntüyü ise test sürecinde kullanmışlardır. Çalışma sonunda %100'e varan başarılarla ulaşan yazarlar, en yüksek başarıyı ResNET 50 mimarisi ile yakaladıklarını belirtmişlerdir [65].

M. Jiang vd. ise öncelikle veri setlerini bir ön-işlem (veri artırımı, PCA, vs.) sürecinden geçirmiş, daha sonra ise veri setlerini WBCNet adını verdikleri 33 katmanlı bir CNN mimarisi ile eğitmişlerdir. Çalışma sonuçlarını diğer mimariler ile kıyaslayan yazarlar, kendi mimarilerinin daha iyi sonuçlar ürettiğini göstermişlerdir [66].

G. Liang vd. çalışmalarında CNN ve Yinelemeli Sinir Ağları (RNN) yöntemlerini birleştiren hibrit bir model kullanmışlardır. Veri artırma teknikleri ile veri seti boyutunu arttıran yazarlar, modelin CNN tarafında transfer öğrenme ile özellik çıkarımı yapmışlardır. Veri setlerini RNN tarafında ise Uzun Kısa-Vadeli Hafıza (LSTM) yöntemi ile eğiten yazarlar, iki modelin ürettiği özellik çıkarımlarını

birleştirek tüm ağı tekrardan eğitmişlerdir. Veri setlerinde bazofil sayısı yetersiz olduğundan, dört sınıflı bir modelleme yapılmış, çalışma sonunda önerdikleri hibrit modelin standart CNN'ye nazaran daha başarılı olduğunu iddia etmişlerdir [67].

F. Qin vd. çalışmalarında derin artık (residual) ağlar yönteminden faydalanan ve Cell3Net olarak adlandırdıkları bir mimari ile WBC alt-tiplerinin sınıflandırılmasını sağlamışlardır. Oluşturulan ağın diğer mimarilere oranla daha başarılı olduğu belirtmişlerdir [68].

T. T. P. Thanh vd. çalışmalarında kan kanseri teşhisi için kullanılabilir bir sınıflandırma modeli önermişlerdir. Yedi katmanlı bir CNN modeli kullanılmış ve %96,6 başarı elde edilmiştir [69].

P. Tiwari vd. çalışmalarında CNN yönteminden faydalanmış ve WBC alt-tiplerini tek-granüllü ve çok-granüllü olarak ikili sınıflandırma yoluna gitmişlerdir. Bazofil hücrelerini sınıflandırma dışında tutan yazarlar, SVM ve Naive Bayes'e göre daha iyi skorlar üretmişlerdir [70].

L. H. S. Vogado vd. CNN tabanlı çalışmalarında transfer öğrenme yöntemi ile özellik çıkarımı gerçekleştirmiş, elde edilen özellik vektörleri kullanılarak SVM ile sınıflandırma yapmışlardır. Özellik vektörlerinin sahip olduğu eleman sayısını gain ratio algoritması ile azaltmışlar ve en az parametre ile en iyi sonuçlara ulaşmayı hedeflemişlerdir. Önerdikleri modelin en iyi sonuçlara sahip olduğunun göstermişlerdir [71].

Literatürde de gözlemlenebileceği üzere DL ile yüksek başarı oranlarına ulaşmak mümkün olsa da bu yöntemlerin eğitim sırasında yüksek sayıda verilere ihtiyaç duyması, medikal görüntülerle çalışan araştırmacılar için büyük bir problem arz etmektedir [8]. Bu sebepten dolayı araştırmacıların veri setlerinde *veri artırma* tekniklerinden faydalandığı, bu sayede veri sayısını yeterli seviye çıkarmaya çalıştıkları gözlemlenmektedir. Fakat bu yöntemlerin başarıyı sadece bir noktaya kadar yükseltebildiği, başarının daha da artırılabilmesinin özgün verilerle sağlanabileceği bilinen bir gerçektir [10].

Ek olarak, yapılan çalışmalarda gözlemlendiği üzere, bazı araştırmacılar [61,62,65,71] yetersiz veri problemiyle başa çıkabilmek adına transfer öğrenme/ince-ayar yöntemlerinden faydalanırken, bir kısmı ise [62,63,67,71] çalışmalarında hibrit modeller kullanmışlardır. Diğer yandan DL yöntemleri direkt görüntünün kendisinden çıkarımlar yapabilmesine rağmen, araştırmacıların [58–60,63,65,66,68,69,72] başarı oranını arttırabilmek adına ön-işlem süreçlerinden faydalandıkları görülmüştür. Diğer yandan veri setinin sınıflarında yer alan örneklerin yetersiz olduğu durumlarda, bazı çalışmalarda [67,70] bazı sınıfların eğitime dahil edilmediği gözlemlenmiştir Çizelge 4.1.

Çizelge 4.1. Literatür özeti.

Yazar (Yıl)	Yöntem	Veri seti Boyutu	Ön-işleme	Sınıf Azaltma	Hibrit Mimari	Transfer Öğrenme	Başarı Oranı (%)
Habibzadeh vd. (2013)	CNN	140	Var	-	-	-	85,00
Choi vd. (2017)	CNN	48.000	Var	-	-	-	97,00
Li vd. (2017)	PCA + CNN	24.222	Var	-	-	-	94,75
Mundhra vd. (2017)	U-Net + CNN	17.340	Var	-	-	-	98,00
Shanin vd. (2017)	Transfer Öğrenme	2.551	-	-	Var	Var	96,00
Yu vd. (2017)	CNN	2.000	-	-	-	Var	88,50
Zhao vd. (2017)	SVM + CNN + RF	1.425	Var	-	Var	-	92,80
Habibzadeh vd. (2018)	İnce-ayar	11.200	Var	-	-	Var	100,00
Jiang vd. (2018)	PCA + CNN	102.000	Var	-	-	-	83,00
Liang vd. (2018)	TL + LSTM (RNN)	12.444	-	Var	Var	-	90,79
Qin vd. (2018)	DRN	92.480	Var	-	-	-	76,84
Thanh vd. (2018)	CNN	1.188	Var	-	-	-	96,60
Tiwari vd. (2018)	CNN	13.000	-	Var	-	-	88,00
Vogado vd. (2018)	TL + PCA + SVM	1.268	-	-	Var	Var	100,00

Sonuç olarak, özellikle medikal görüntülerde sıklıkla karşılaşılan *veri seti yetersizliği* ile başa çıkabilecek ve ön-işleme sürecine ihtiyaç duymayan bir model ihtiyacı bulunmaktadır. Bir sonraki bölümde hem küçük veri setlerinde dahi başarılı sonuçlara ulaşabilen, hem de DL'nin sağladığı avantajlardan faydalanılmasını sağlayan Kapsül Ağları tanıtılacaktır. Ama öncesinde, DL ve CNN yönteminin sahip olduğu genel ve yapısal problemler incelenecektir.

4.2. DERİN ÖĞRENMEDEKİ PROBLEMLER

Sahip olunan genel bilgiler ve incelenen literatür doğrultusunda, tezin bu aşamasına kadar yapılan çıkarımlar özetlenirse;

Medikal görüntü işleme zordur. Çünkü;

- Medikal görüntülerde sağlıklı bir analiz gerçekleştirilebilmesi için gereken veri sayısına ulaşmak gerek etik nedenlerden gerekse medikal cihazlardan görüntü elde etme süreci maliyetli olduğundan oldukça zor bir süreçtir.
- Bir diğer problem olan *dengesiz veri seti* probleminden bahsetmek gerekirse; beyinde spesifik bir lezyonun var olup olmadığını tespit eden bir sistem geliştirilmek istendiğinde, o lezyonun bulunmadığı görüntülerden yüzlerce bulmak mümkünken, lezyonun var olduğu görüntülerden belki de birkaç tane edinilebilecektir.
- Görüntülerde segmentasyon veya sınıflandırılma gerçekleştirilecekse, verilerin bir ön-işlem sürecinden geçirilmesi gerekmektedir. Bu süreç oldukça zaman alıcıdır. Diğer yandan, bu süreçte yapılacak hatalar yüzünden tüm modelin olumsuz etkilenmesi ihtimali vardır. Kısaca, “yanlış segmentasyon”, “yanlış özellik vektörü” oluşumuna yol açar. Bu da “yanlış sınıflandırma” anlamına gelir.

DL bu problemlerin üstesinden gelebilir. Çünkü;

- Veri artırma sayesinde yetersiz veri probleminin üstesinden gelinebilir.
- Üst-örnekleme ve alt-örnekleme sayesinde dengesiz veri seti sorununun üstesinden gelinebilir.
- Ön-işleme süreci gerektirmediğinden daha hızlı ve ön-işleme sürecindeki hatadan etkilenmeyen bir sistem geliştirilebilir.
- Diğer yandan transfer öğrenme sayesinde daha önce yüksek boyutlu veriler üzerinde başarılı bir modelleme gerçekleştiren bir sistemin yetenekleri kullanılarak, düşük boyutlu veri setlerinde de başarılı sonuçlara ulaşmak mümkün olabilmektedir.

Fakat DL yöntemlerinin de bazı problemleri mevcuttur. Bunlar;

- DL yöntemlerinin başarısı kullanılan veri setinin boyutuyla doğru orantılı olarak artmaktadır. Veri seti ne kadar fazla örneğe sahip ise model o kadar fazla çıkarım gerçekleştirecek ve o oranda bir genelleştirme yeteneğine sahip olacaktır.
- Veri arttırma teknikleri ile örnek sayısı arttırılabilsede üretilen örnekler mevcut görüntülerin farklı varyasyonlarındaki kopyaları olduğundan başarı oranları belli bir noktaya kadar yükselebilecektir. Dolayısıyla, sınıflar arasında ayırım yapmayı sağlayacak özgün verilere ihtiyaç duyulmaktadır.
- Diğer yandan bilgisayarlı görü görevleri üzerinde çalışan DL yöntemleri bazı yapısal problemlere sahiptir.

Kapsül Ağları bu üç problemin de üstesinden gelinebilmesinde efektif bir şekilde kullanılabilir bir yöntemdir. Bu yöntemin nasıl çalıştığı derinlemesine anlatılmadan önce, bilgisayarlı görüde oldukça başarılı bir şekilde kullanılan CNN yönteminin sahip olduğu yapısal problemler incelenecektir.

4.2.1. CNN Yönteminin Sahip Olduğu Problemler

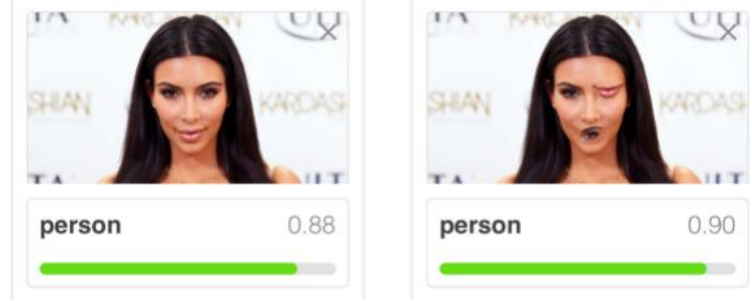
CNN yöntemi konvolüsyonel operasyonlar aracılığıyla verilerin nitelikli özelliklerini çıkarmada ve bu özellikleri özetleme de oldukça başarılı olsa da temel olarak üç önemli probleme sahiptir. Bu problemler sebebiyle CNN modeli bazı önemli özellikleri aktaramamakta ve veri kaybı yaşanmaktadır. Bunlar:

- Yer değiştirme değişmezliği (translational invariance),
- Bakış açısı değişmezliği (viewpoint invariance),
- Maksimum havuzlama (max-pool) yüzünden yaşanan veri kaybıdır [73].

Yer değiştirme değişmezliği bilgisayar grafiklerinde kullanılan bir terimdir [74]. Eğer başarılı bir modelleme gerçekleştirmek isteniyorsa girişte kullanılan verinin üzerinde yapılacak dönüşümsel değişikliklerin çıkışı değiştirmemesi beklenir. Yani bir “eşitlik” durumuna ihtiyaç vardır.

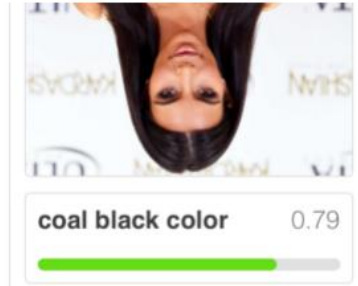
Şekil 4.1’de görüldüğü üzere bir insan yüzünde mevcut olan ağız ve göz olması gereken yerlerde bulunmamasına rağmen CNN tarafından insan olarak algılanmıştır.

Çünkü model bir yüzde iki göz, bir burun ve bir ağız bulunması gerektiğini öğrenmiştir. Bu özelliklerin nerede olduğu veya hiyerarşisiyle ilgilenmemiştir.



Şekil 4.1. CNN'deki yer değiştirme değişmezliği problemi [75].

Bakış açısı değişmezliği ise girişte yapılacak açı değişimlerinin çıkışı etkilememesi durumudur [74]. Fakat, görüntünün farklı rotasyonlardaki halleri CNN tarafından yanlış algılanmaktadır. Bu aşamada görüntüyü tanımlayacak modelin, giriş verisinde yapılan rotasyona göre çıkışta da ilgili rotasyonu gerçekleştirmesi ve görüntüyü doğru sınıflandırması gerekmektedir. Fakat Şekil 4.2'de de görülebileceği üzere görüntü 180° derece döndürüldüğünde CNN, yüzdeki organların hiyerarşisi ile ilgilenmediğinden, siyah saçlara odaklanmış ve görüntüyü kömür olarak algılamıştır.

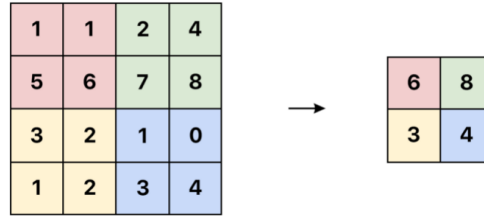


Şekil 4.2. CNN'deki bakış açısı değişmezliği problemi [75].

G. Hinton CNN modelinde kullanılan havuzlama operasyonunu şu cümle ile tanımlamıştır: “CNN'de havuzlama operasyonunu kullanmak büyük bir hataydı, fakat, bu işlemin bu kadar iyi çalışması ise tam bir felaket oldu” [76].

Bilindiği üzere, CNN'de konvolüsyon operasyonları akabinde elde edilen özellik haritaları bir alt-örnekleme (havuzlama) operasyonundan geçirilerek, modele hem yer

değiştirme değişmezliği kazandırılmaya çalışılır hem de eğitim parametresi sayısı yarı-yarıya azaltılır (Şekil 4.3). Fakat daha önce de bahsedildiği üzere, havuzlama işlemi yer değiştirme değişmezliği ve bakış açısı değişmezliğini sunmakta yetersiz kalmaktadır [73]. Bu operasyonun bir diğer handikabı ise piksellerde saklanan bazı önemli özelliklerin kaybına neden olmaktadır. Medikal görüntülerde her bir pikselin oldukça önemli olduğu düşünüldüğünde, kaybın büyüklüğü çok daha yüksek olabilmektedir.



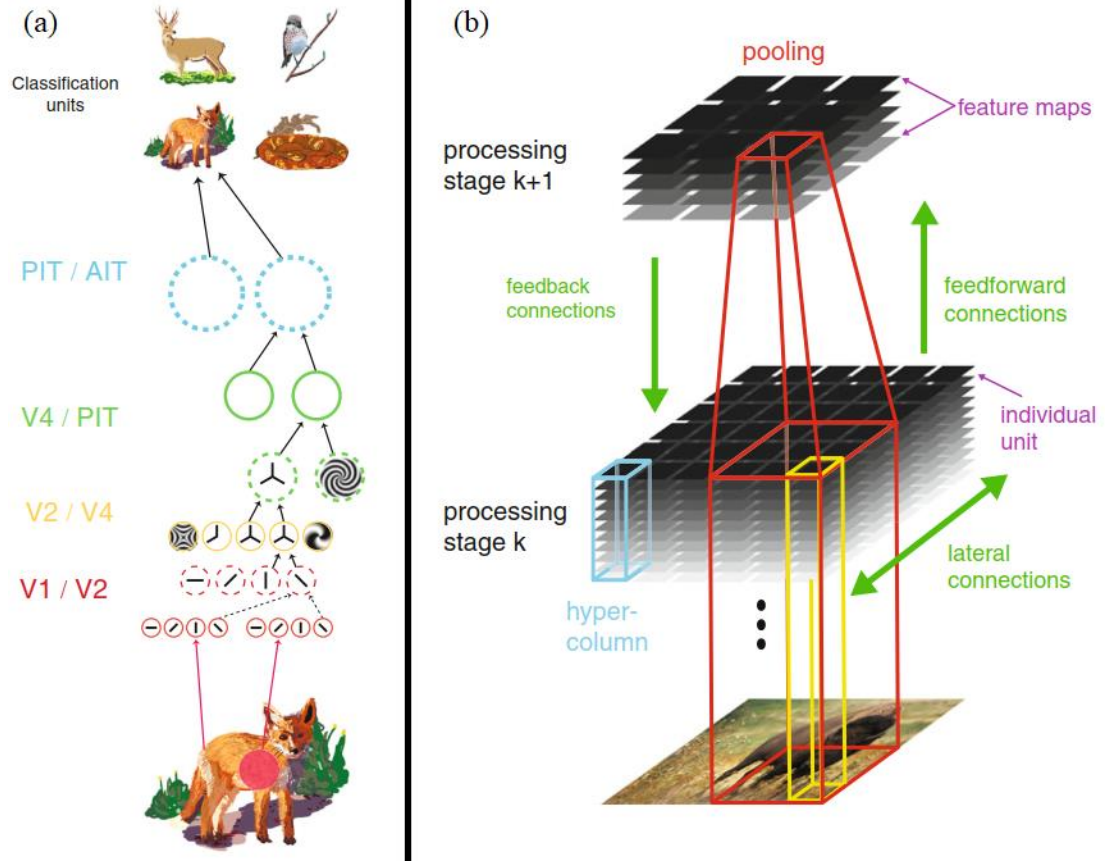
Şekil 4.3. Maksimum havuzlama işlemi [77].

4.3. KAPSÜL AĞLARI

ML, insanların sahip olduğu bazı yeteneklerin bilgisayarlara kazandırılması olarak tanımlanabilir. Böylece makineler de insanlar gibi öğrenebilir ve öğrendikleri bilgileri kullanarak yeni verileri yorumlayabilirler. Örneğin, ANN, insan beyninde bulunan nöronların birbirleriyle ilişkisinin incelenmesiyle ve biyolojik sinir yapısının taklit edilmesiyle elde edilmiştir. Bilgisayarlı görü problemlerinde kullanılan DL yöntemlerinin çalışma prensipleri de insan ve hayvan beyninin görsel korteksinde görüntülerin işleme sürecini taklit etmek üzere dizayn edilmiştir.

Riesenhuber ve Poggio tarafından 1999 yılında gerçekleştirilen çalışmada insan beyninin görüntüleri nasıl analiz ettiği *HMAX* modeli ile açıklanmaya çalışılmıştır [78]. Şekil 4.4-a'da görüldüğü üzere, görsel korteksin ilk katmanlarında reseptif alan tarafından algılanan görüntüden, temel özellik çıkarımları gerçekleştirilmektedir. Daha sonra bu özellikler sonraki katmanlara aktarılır ve daha karmaşık özelliklere odaklanılır. Katmanlar arasında gerçekleştirilen maksimum havuzlama benzeri operasyonlar aracılığıyla da benzer özellikler özetlenir ve karmaşıklık artırılır. Böylece, görüntünün beyinde hiyerarşik olarak sunumu gerçekleştirilir ve görüntünün

sınıfı tanımlanır. Fakat, beyin tarafından görüntü işlenirken mükemmel çalışan bu sistem, iş bilgisayarlar tarafından modellenmeye gelince daha önce de bahsedilen problemler ortaya çıkmaktadır.



Şekil 4.4. a) HMAX modeli ve b) görsel korteksin hiyerarşik modeli [79].

1979 yılında Hubel ve Wiesel tarafından yayınlanan çalışmada ise beynin sahip olduğu farklı bir özellikten bahsedilmiştir: *hiper kolonlar*. Beynin görsel korteksinde yer alan ve her biri yaklaşık 80–120 nöron içeren mikro kolonlar bulunmaktadır. Bu kolonlar aynı reseptif alanı kullanır ve beyne gelen görüntüden çıkarımlar gerçekleştirirler. Mikro kolonlar aynı tip çıkarımları gerçekleştirirken, bir çok mikro kolonun bir araya gelmesiyle oluşan hiper kolonlar (Şekil 4.4-b) ise görüntüden çıkarımı gerçekleştiren tüm özellikleri barındırır [80].

Kapsül Ağları da CNN'nin sahip olduğu problemlerin üstesinden gelebilmek adına, görsel korteksin sahip olduğu bu özelliği taklit eder ve görüntünün özelliklerini hiyerarşik bir düzende temsil etmeye çalışır. Bu amaçla objeleri ve objeleri oluşturan

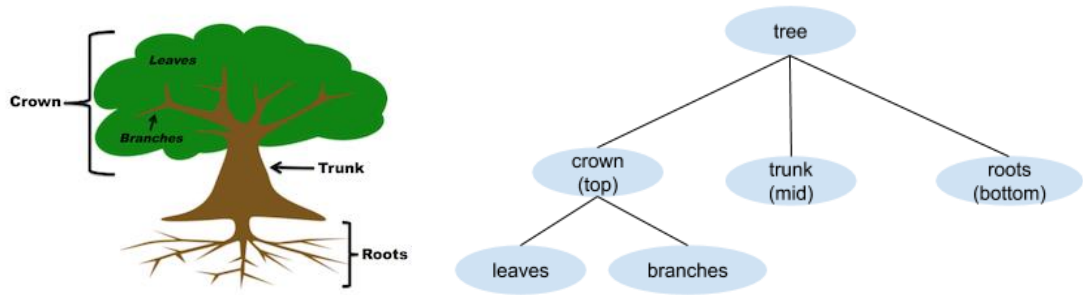
parçaların uzaysal ilişkilerini poz matrislerinde (çeviri ve dönüşüm matrisleri) saklar. Daha sonra bu bilgileri kullanarak, objeyi yeniden inşa etmeye çalışır. Böylece objenin açısı, konumu gibi özellikleri değişse dahi objeyi doğru bir şekilde tanımlayabilme yeteneğine sahip olur (ters grafik) [81].

G. Hinton bu tip bir modelin oluşturulması ile ilgili fikirlere 1981 [82] yılından itibaren sahip olsa da bu fikirleri gerçek dünya problemlerinde test edebilmesi 2011 yılını bulmuştur [83]. Bunun sebebi ise Kapsül Ağlarının oldukça yüksek eğitim parametresine sahip olmasından dolayı, işlem gücü yüksek makinelere ihtiyaç duyulmasıdır.

İlerleyen bölümlerde Kapsül Ağlarının sahip olduğu özellikler, problemleri çözerken yaptığı varsayımlar ve arka planda gerçekleştirdiği hesaplamalar görülebilir.

4.3.1. Hiyerarşik Sunum

Bir önceki bölümde verildiği üzere insan beyni görüntüleri, görüntüyü oluşturan parçaların hiyerarşik bir düzende sunulması sayesinde tanımaktadır. Çünkü, her bir obje, daha küçük objelerin bir araya gelmesi sayesinde oluşmaktadır. Örneğin, bir ağaç: *gövde*, *taç* ve *kökten* oluşur. Taç adı verilen bölümde de *dallar* ve *yapraklar* yer alır (Şekil 4.5). Objeyi ve objenin bölümlerini oluşturan bu parçalar belli bir hiyerarşik düzendedir.



Şekil 4.5. Ağacın hiyerarşik sunumu [84].

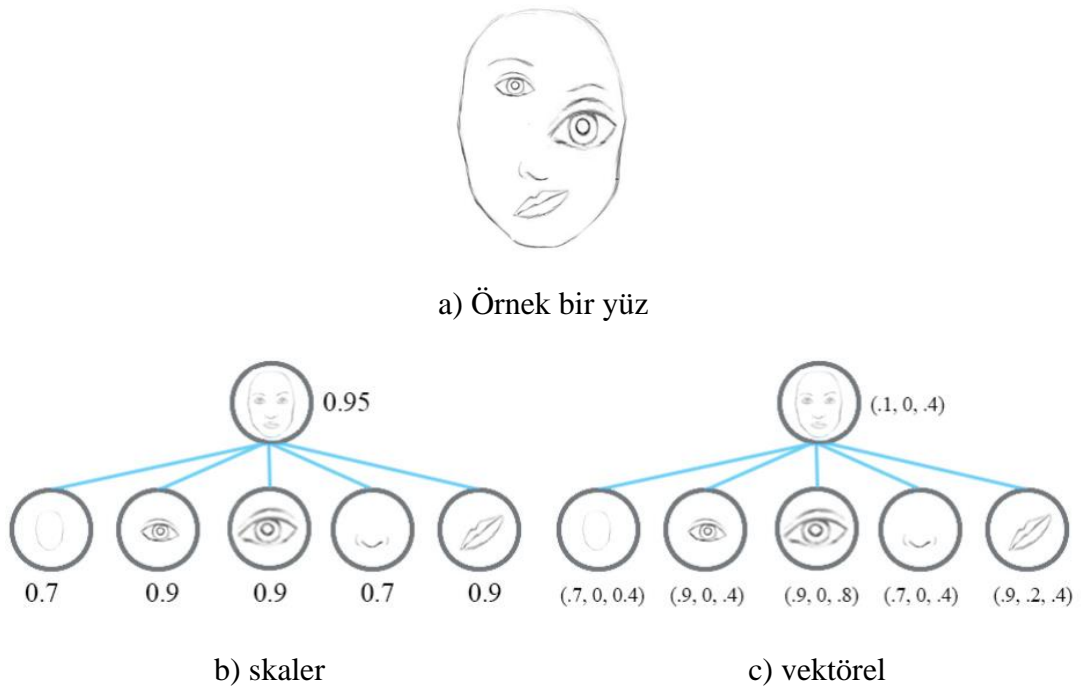
Bir obje gördüğümüzde beyin bazı sabit noktalar oluşturur ve bu noktalara göre objeyi oluşturan kısımların pozisyonunu tutarak objeyi tanır. Böylece beynimiz her bir detay

ile derinlemesine ilgilenmeden, sadece dallar ve yapraklar ile objede bir üst bölüm olduğunu fark edebilir. Bu gibi hiyerarşik bilgiler sayesinde de objenin bir ağaç olduğu tahmin edilebilmektedir.

4.3.2. Bakış Açısı Değişmezliği (Invariance)

CNN özellik çıkarımında iyi olsa da perspektif, boyut, oryantasyon gibi uzaysal ilişkileri çıkarmada yetersizdir. Örneğin Şekil 4.6-a'da görülen yüz örneği CNN modelini kolayca kandırabilir. Basit bir CNN modeli burun, göz ve ağız gibi özelliklerin başarılı bir şekilde çıkarımını sağlasa da yüzü tespit eden nöronu yanlış bir şekilde aktive edecektir. Uzaysal oryantasyon ve boyut olmadan, yüzü tespit eden nöronunun aktivasyonu da oldukça yüksek olacaktır (Şekil 4.6-b).

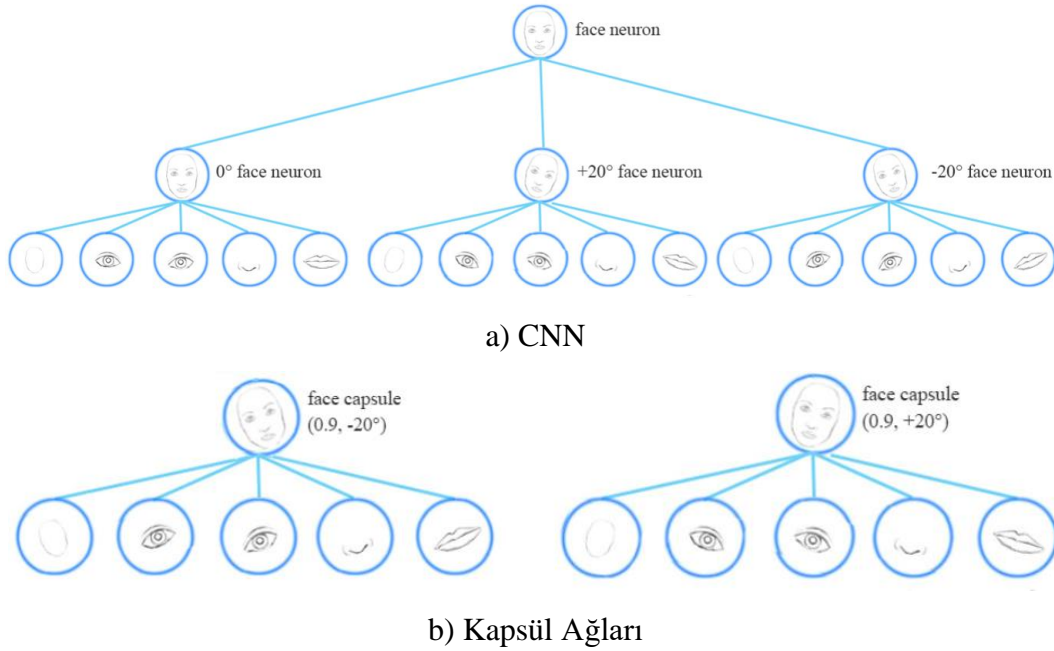
Fakat, her bir nöronun özellik çıkarımının yanında benzerlik değerlerine de sahip olduğunu farz edelim. Böylece çıkış olarak benzerlik, oryantasyon ve boyut tutan *vektörler* oluşacaktır. Uzaysal ilişkilere sahip olduğumuzdan, burun, gözler ve kulaklar arasındaki oryantasyon ve boyut gibi tutarsızlıklar tespit edilecek ve yüz tespiti için daha düşük bir aktivasyon oranı ortaya çıkacaktır (Şekil 4.6-c).



Şekil 4.6. Nöronların özelliklere göre aktivasyonu [85].

4.3.3. Bakış Açısı Eşitsizliği (Equivarance)

CNN farklı özelliklerin varyantlarını yakalarken çoklu nöron yapısını kullanmaktadır (Şekil 4.7-a). Kapsül ağında ise, çoklu varyantların tespitinde aynı kapsüller paylaşılmaktadır (Şekil 4.7-b). Eşitlik dediğimiz kavram, birbirine dönüşebilen objelerin tespitidir. Kapsüllerin sahip olduğu bu özellik sayesinde, bir yüzün sağa ya da sola dönmüş olup olmadığı, aynı kapsül tarafından tespit edilebilirken, CNN modelinde bu açı değişikliğini belirten farklı kombinasyonlara ihtiyaç duyulmaktadır. Bu bakımdan, Kapsül Ağlarında, model oluşabilecek tüm varyantları öğrenmeye zorlandığından daha az eğitim verisi ile daha efektif bir çıkarım sağlanır.

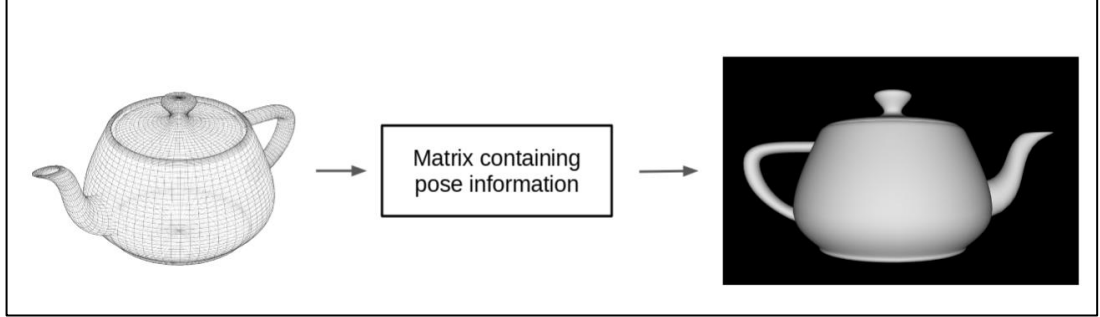


Şekil 4.7. Kapsüllerin objenin uzaysal değişikliklerine göre adaptasyonu [85].

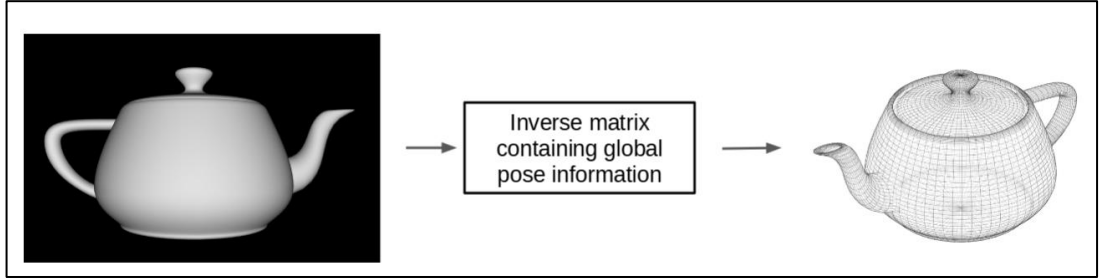
4.3.4. Ters Grafik (Inverse Graphic)

G. Hinton bilgisayarlı görüyü *ters grafikleme* olarak tanımlar [73]. Kapsül Ağları da objeleri ve objeleri oluşturan parçaları ortaya çıkarırken ters grafik adı verilen bir yöntem kullanır. İnsan beyninin de bu tip bir algılama süreci ile objelerin uzaysal ilişkilerini ortaya çıkardığı düşünülmektedir. Bu süreç, bilgisayarların görüntüleri ekrana yansıtmak için kullandığı *rendering* işleminin tersidir. Rendering işleminde

tüm objenin sahip olduğu poz bilgisi bir dönüşüm matrisiyle çarpılarak 2B olarak bilgisayar ortamına aktarılır (Şekil 4.8-a).



a) Bilgisayarlardaki rendering işlemi



b) Ters grafikleme

Şekil 4.8. Rendering ve ters grafik [86].

Ters grafik sürecinde ise bunun tam tersi yapılır. Yani; 2B görüntü dönüşüm matrisinin tersiyle çarpılarak objenin poz bilgisi elde edilir (Şekil 4.8-b). Bu matrisler aynı zamanda bakış açısı değişmezliğine de sahiptir. Yani objeyi oluşturan parçaların oryantasyonu değişse de tüm objenin pozisyonuna aynı matris ile ulaşılabilir. Objenin bakış açısı özelliklerinin matrislerde tutulması bize tam bir bağımsızlık kazandırır. Böylece objenin yer değiştirme bilgileri nöronsal aktivite olarak değil, matris ağırlıkları olarak saklanır.

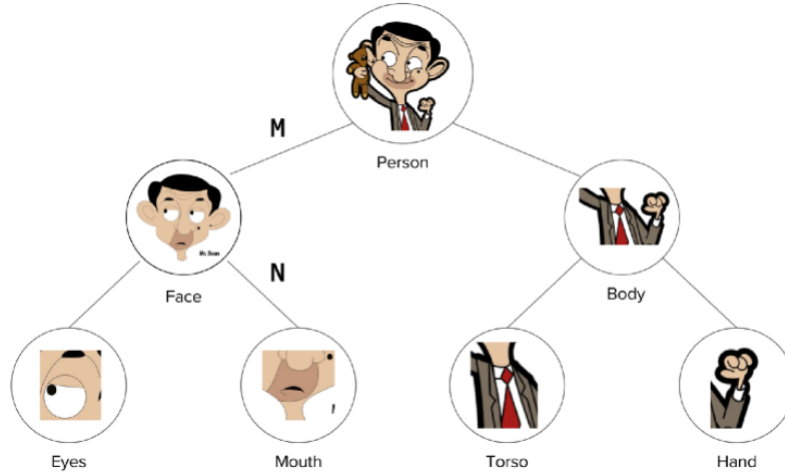
4.3.5. Poz Matrisi

Poz matrisi ya da dönüşüm matrisi 4×4 boyutundadır ve objenin özelliklerini koordinat düzleminde tanımlamaya yarar. T değerleri dönüşüm, R değerleri ise rotasyon ve ölçek değerlerini tutmaktadır (Şekil 4.9-a).

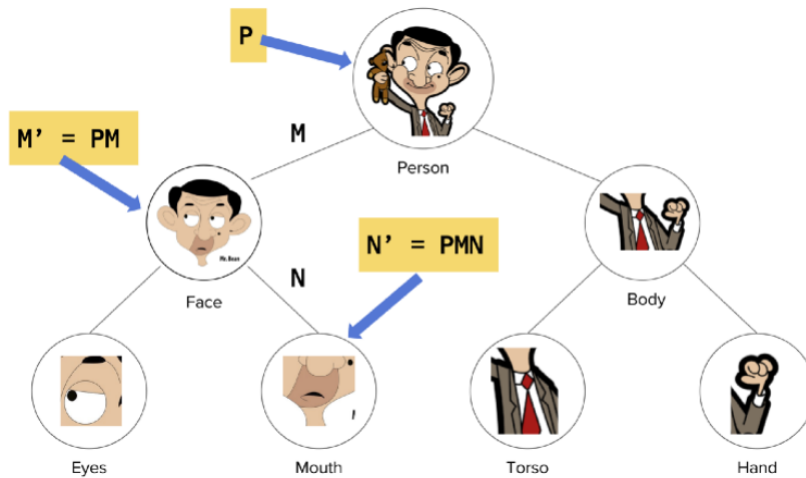
Daha önce de bahsedildiği üzere, bir bütün parçalardan oluşur ve bu parçalar hiyerarşik olarak tüm obje ile poz matrisi ile ilişki altındadır. Poz matrisi ile küçük parçalar, büyük objenin koordinat düzleminde ifade edilebilmektedirler. Eğer M matrisi yüz ile insan arasındaki ilişkiyi tanımlayan matris, N ise ağzın yüze göre pozisyonunu tutan matris ise, o zaman bu iki matrisin çarpımı sayesinde ağzın koordinat düzlemindeki ifadesi ortaya çıkarılabilir (Şekil 4.9-b).

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

a) Poz matrisi



b) Objeleri oluşturan parçaların koordinat düzlemindeki ifadesi



c) Tüm bakış açılarının çıkarımı

Şekil 4.9. Poz matrisleri aracılığıyla özelden genele gitmek [75].

Kamera aracılığıyla bir kare aldığımızda ve bu karenin poz matrisi P biliniyorsa, o zaman, objeyi oluşturan tüm parçaların gerekli özelliklerini ve koordinatlarını poz matrislerinin birbiriyle çarpımı ile elde etmek mümkün olacaktır (Şekil 4.9-c).

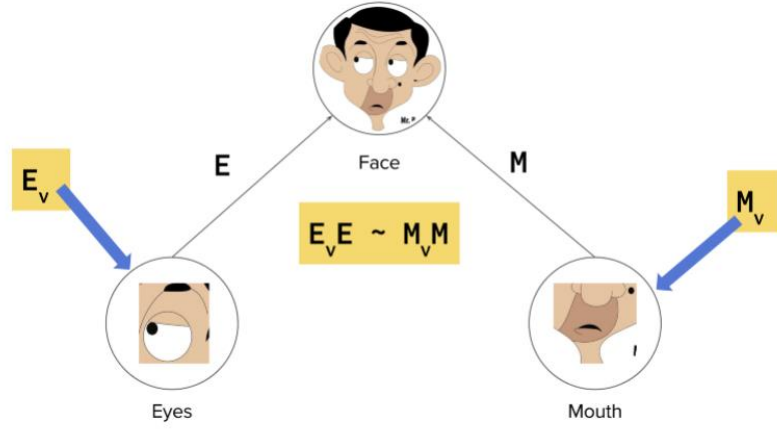
Örnekte de görüldüğü üzere, yüzün kameraya göre koordinatları, kameradan gelen poz matrisi ve yüzün insana göre koordinatlarıyla çarpılarak bulunmuştur. Yine aynı şekilde ağzın pozisyonu da ilgili matrislerin çarpımı ile ifade edilebilir.

Poz matrisi, baktığımız objenin farklı bakış açılarını sunmaktadır. Yüzün sahip olduğu tüm özellikler aynı olacakken, değişen kısım bakış açımıza göre yüzün pozisyonudur. Tüm bakış açılarının çıkarımı sadece P matrisi kullanılarak gerçekleştirilebilir. Ayrıca, poz matrisi ile koordinat dönüşümü gerçekleştirilmektedir. Burada, X_w insanın gerçek dünyadaki koordinatı iken, X_c ise kişinin kamera açısından sahip olduğu koordinattır (Şekil 4.9-a).

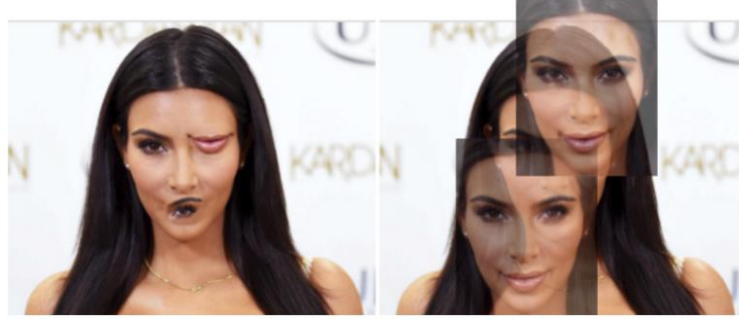
4.3.6. Poz Matrisi ile Tahmin Gerçekleştirmek

Eğer sol gözün nerede olduğunu biliyorsak, yüzün geri kalanı nerededir tahmin edebiliriz. Benzer şekilde, ağzın pozisyonu kullanılarak, yüzün nerede olduğu ile ilgili tahmin yürütülebilir. Eğer düzgün bir yüz giriş görüntüsü olarak kullanılırsa hem sol göz hem de ağız açısından yüz ile ilgili yapılacak tahminler benzer şekilde olacaktır. Böylece bu parçaların aynı yüze ait olduğu ve ilişkili oldukları söylenebilir. Bu sebeple, görüntüyü ters çevirsek dahi, ters dönen ağız ve göz sayesinde, yüzün de ters dönmüş olduğu anlaşılacaktır. Sonuç olarak, bu parçaların bir bütüne ait olduğu görülmüş olacaktır.

Şekil 4.10'da görüldüğü üzere, eğer gözün ve ağzın pozisyonları (E_v ve M_v) ML yöntemleri ile tahmin edilebilirse, yüze ait poz matrisi, bu tahminlerin ters poz matrisleriyle (E ve M) ile çarpılmasıyla elde edilebilir (Şekil 4.10-a). Şekil 4.10-b'ye bakarsak, ağıza göre yüzün pozisyonu üst-köşede olması gerekirken, göze göre ise aşağıda olduğu tahmin edilmiştir. Sonuçta bir anlaşamama durumu söz konusudur. Yani bu parçaların bir bütüne ait olmadığı sonucuna varılabilir.



a) Ağız ve gözün pozisyon tahmini



b) Gerçek dünya probleminde tahmin gerçekleştirilmesi

Şekil 4.10. Ağız ve gözün pozisyonuna göre yüzün pozisyon tahmini [75].

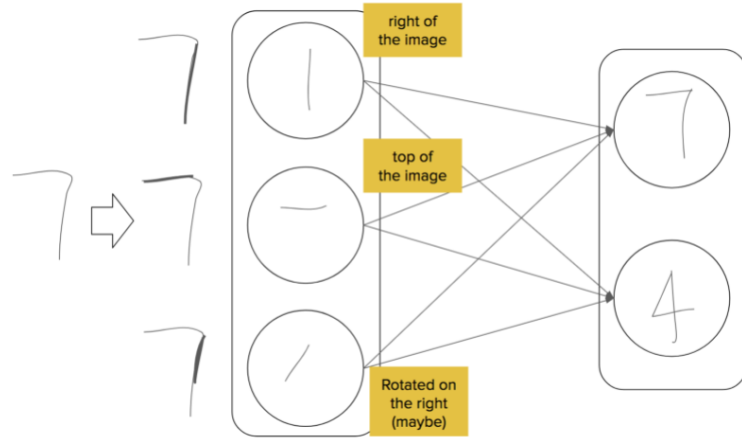
Eğer bir anlaşma olacaksa, bu anlaşmanın dayandığı en temel özellik, parçaların doğru pozisyonlarda ve birbiriyle ilişkili durumda olması gerekir. Sonuçta, oluşturulan modelin doğru tanımlama için gereken doğru pozisyonları öğrenmesi gerekir. Böylece görüntüde sadece ağız ve gözlerin varlığını sorgulamak yerine, bir yüzün parçası olabilmesi için, ağzın aşağıda ve iki gözün ortasında olması gerektiğini de öğrenecektir. Bu sayede, lineer dönüşümler gerçekleştirilerek uzaysal ilişkilerin modellenebilmesi, birçok bakış açısının genelleştirilebilinerek bilgilerin hiyerarşik olarak dizayn edilebilmesi mümkün olacaktır.

4.3.7. CNN Nerede Başarısız Oluyor?

CNN özelliğın olup olmadığını belirten ve buna göre aktive olan skaler çıkışlara sahiptir. Kapsül Ağları ise sadece aktivasyon değeri üretmeyen, aynı zamanda, obje ile ilgili bilgileri de barındıran vektör/matrislere sahiptir. Bu özellikler pozisyon,

rotasyon, kalınlık vb. her şey olabilirler. Peki bu özellikleri elde etmek için neden sekiz adet konvolüsyon katmanı yerine 8B kapsüller kullanmamız gerekiyor?

Şekil 4.11-a'daki sinir ağı yapısında da görüldüğü üzere, 7 rakamının özelliklerinin çıkarımı yapıldıktan sonra eğer görüntü yatay, dikey ve hafif bir diyagonal özelliğe sahip ise yüksek aktivasyon üretmesi şeklinde kodlanmıştır. Eğer çıkışta bu rakamın 7 mi yoksa 4 mü olduğuna karar verecek bir model kullanılıyorsa, iki rakam da benzer özelliklere sahip olduğundan model 4 rakamı için de yüksek bir olasılık değerine sahip olacaktır.



a) Standart sinir ağı mimarisinin 7 rakamını yorumlaması



b) Kapsül Ağlarının her bir özelliğe göre 4 rakamını farklı pozisyonlarda tanımlaması

Şekil 4.11. CNN ve Kapsül Ağlarının giriş özelliklerini yorumlaması [75].

Fakat kapsüller her bir çıkışta özellik hakkında pozisyon, ölçek ve rotasyon bilgilerine de sahip olduğundan (Şekil 4.11-b);

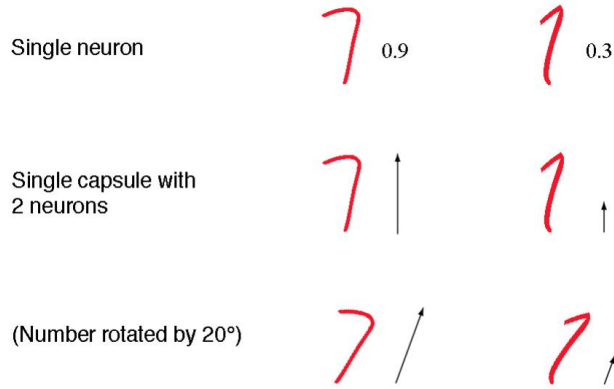
- 4 rakamını tanımlayan kapsül 7 girişini aldığı anda, yeri yanlış olan dikey ve yatay kenarla, zorlukla görülebilen bir diyagonal ile karşılaştığından düşük bir aktivasyon değeri üretecektir.

- Fakat, 7 rakamını tanımlayan kapsül 7 girişini aldığıında, doğru şekilde yerleştirilen dikey ve yatay kenar, zorlukla görülen bir diyagonal çizgi görecektir. Böylece, çıkış değeri de yüksek bir değer olacaktır.

Şekil 4.11-b'de görüldüğü üzere tespit edilen özelliklere göre 4 rakamının pozisyonunun tahmini her bir özellik için farklıdır.

4.3.8. Vektörler ile Objelerin Sunumu

7 rakamının varlığı tek bir nöron ile tahmin etmeye çalışıldığında (Şekil 4.12-a), ilk rakam 7'ye benzer olduğundan yüksek bir aktivasyon, ikinci rakam ise daha çok 1'e benzediğinden daha düşük bir aktivasyon değerine sahip olacaktır. Kapsül ise birden fazla nöronu barındırmaktadır (Şekilde '2' tane). Kapsüller de 7 rakamı için, sahip olduğu nöronların çıkış değerlerini kullanacak fakat, tek bir nörondan farklı olarak bunların uzunluklarını (değerlerin karesinin toplamının karekökü) hesaplayacaktır. Vektörün sahip olduğu uzunluk değeri, özelliğin var olup olmadığını belirten olasılık değeri olarak kabul edilecektir.



a) 7 rakamının vektörel olarak ifade edilmesi



b) 7 rakamının farklı özelliklerinin temsili

Şekil 4.12. Objelerin vektörel olarak ifadesi [85].

Eğer obje 20° sağa yatık ise, obje vektörel olarak aynı büyüklük değerine sahip olacak, fakat, farklı bir oryantasyonda ifade edilecektir. Yani, görüntünün açısı değişse dahi var olma olasılığı aynı kalacaktır. Vektörün sahip olduğu açı değeri objenin sahip olduğu açıyı tanımlayacaktır.

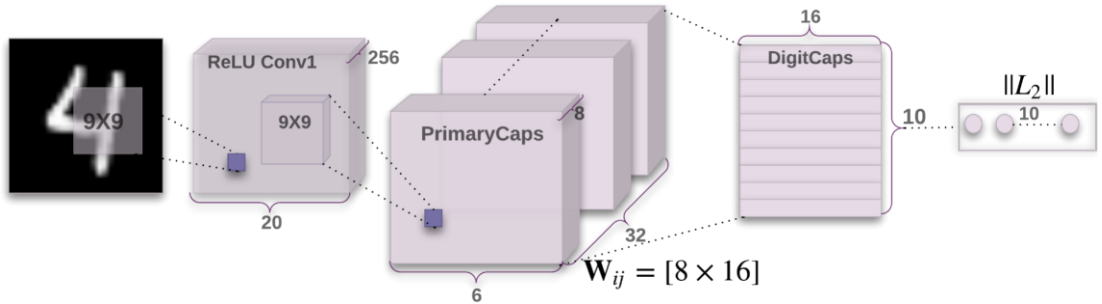
Diğer yandan, kapsüle '2' tane daha nöron eklenerek boyut ve kalınlık gibi özellikler de tutulabilir ve karşılaştırmalarda kullanılabilir (Şekil 4.12-b).

4.3.9. Kapsül Ağlarının Yapısı

Kapsül Ağları temelde iki bölümden oluşur: *kodlayıcı* ve *çözücü*. Kodlayıcı bölümü görüntünün özelliklerinin çıkartılmasından ve sınıflandırılmasından sorumlu iken, çözücü bölümü ise görüntüyü tekrar inşa etmek için kullanılmaktadır.

4.3.9.1. Kodlayıcı

Bu bölümde, görüntülerin özellikleri çıkarılır ve *örnekleme parametrelerini* içeren vektörlere dönüştürülür. Daha sonra bu vektörler kullanılarak eğitim gerçekleştirilir ve yeni gelen görüntünün sınıfı tahmin edilmeye çalışılır. 2017 yılında Sabour vd. [73] tarafından MNIST [41] veri setinin analizinin gerçekleştirildiği çalışmada kullanılan kodlayıcı mimarisi Şekil 4.13'te görülmektedir.



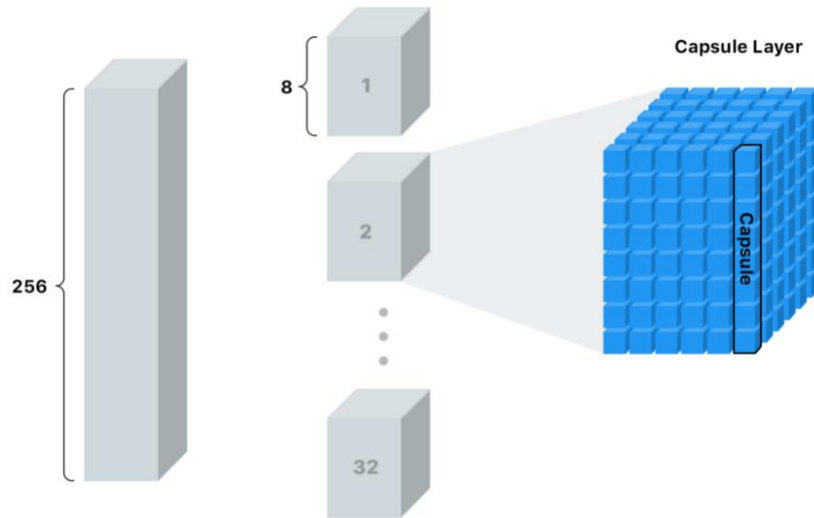
Şekil 4.13. Kapsül Ağlarının kodlayıcı bölümü [73].

Kapsül Ağlarının kodlayıcı kısmı üç ana bölümden oluşmaktadır; *Konvolüsyon* katmanı, *PrimaryCaps* (Birincil Kapsüller) ve *DigitCaps* (Makaledeki problem rakam tanıma olduğu için DigitCaps olarak adlandırılmıştır). Konvolüsyon katmanında,

görüntünün konvolüsyon filtreleri aracılığıyla özellik çıkarımı gerçekleştirilmektedir. 28×28 çözünürlüğe sahip görüntü 9×9 boyutlarına sahip filtreler ($padding = valid$, $strides = 1$) vasıtasıyla taranarak, kullanılan filtre kadar (256) derinliğe sahip olacak, 3B özellik haritaları elde edilir ($20 \times 20 \times 256$). Daha sonra, *ReLU* fonksiyonu aracılığı ile de özellik haritasının sahip olduğu değerlere non-lineerlik kazandırılarak aktive edilmiştir.

Sonraki aşamada ise, aktive edilmiş özellik haritası yine 256 adet, 9×9 boyutundaki filtreler vasıtası ile taranmıştır ($padding = valid$, $strides = 2$). Bu bölümde, 1×1 konvolüsyon yöntemi uygulanarak özellik haritasından eğitim parametresini fazla arttırmadan daha fazla çıkarım sağlanması amaçlanmıştır [87]. Konvolüsyon işlemleri sonunda $6 \times 6 \times 256$ boyutlarına sahip bir özellik haritası elde edilmiştir.

PrimaryCaps bölümü ise, konvolüsyon işlemlerinden sonra elde edilen özellik haritalarının yeniden şekillendirildiği ve vektörlere dönüştürüldüğü kısımdır. Bu bölümde, 256 katmana sahip özellik haritası, her biri 8 derinliğe sahip olacak şekilde bölünerek 32 adet ve $6 \times 6 \times 8$ boyutuna sahip olan birincil kapsüllere dönüştürülür (hiper kolonlar). Her bir birincil kapsül ise $6 \times 6 = 36$ adet düşük seviye kapsülden (mikro kolonlar) oluşmaktadır. Sonuç olarak; her biri 8 boyutlu $36 \times 32 = 1152$ adet düşük seviye kapsül, 32 adet ise birincil kapsül elde edilmiş olur (Şekil 4.14).



Şekil 4.14. Kapsül Ağlarındaki birincil kapsüller bölümü [77].

Olasılık Hesabı

Bilindiği üzere, ANN ve CNN gibi klasik nöron yapısına sahip ağlar, girişten veya diğer nöronlardan gelen skaler değerleri, yine skaler ağırlık değerleriyle çarparak ve toplayarak bir sonraki nörona iletir (Eşitlik (4.1)).

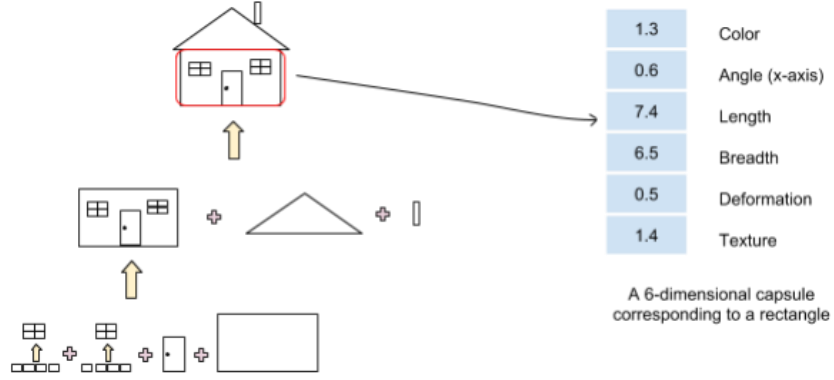
$$a_j = \sum_i w_i x_i + b \quad (4.1)$$

Hesaplama sonunda elde edilen skaler değer bir sonraki aşamada bir aktivasyon fonksiyonuna (ör: sigmoid, tanh vs.) verilir, bu sayede hem değer normalize edilmiş hem de non-linear özellik kazandırılmış olur (Eşitlik (4.2)).

$$h_j = f(a_j) \quad (4.2)$$

Böylece bir sonraki katmanda giriş değeri olarak kullanılacak değerlerin hesaplanması sağlanır. Kapsül Ağlarında ise, girişten gelen skaler değerler ağırlık matrisleriyle çarpılarak vektör özelliği kazandırılmaktadır. Hesaplama sırasında kullanılan matris daha önce bahsedilen *poz matrisidir* ve *afin transformasyon* sağlaması amacıyla kullanılır. İşlemlerin akabinde elde edilen vektörlerin bir C matrisi (daha sonra açıklanacak) ile ağırlık çarpımı ve toplaması gerçekleştirilir. Son bölümde ise klasik nöronlarda olduğu gibi bir aktivasyon fonksiyonu aracılığı ile vektörün uzunluğunun $[0,1]$ aralığına taşınması sağlanır. Bu değer aynı zamanda aranılan özelliğin *bulunma olasılığını* belirtir.

Daha önce de bahsedildiği üzere bir kapsül birkaç nöronun bir araya gelmesiyle oluşur. Bu kapsüllerin her biri de örnekleme parametreleri (pozisyon, ölçek vb.) barındırır. Şekil 4.15’de görüldüğü gibi *altı* nörona sahip bir kapsülümüz olduğunu farz edelim. Bu kapsül de evin dikdörtgen bölümünü tanımlayan özellik değerlerini barındırır. Eğer bu kapsülün uzunluğu dikdörtgenin bulunma olasılığını belirtecekse, bu olasılık değeri karelerin toplamının karekökü ile hesaplanır (Eşitlik (4.3)).



Şekil 4.15. Bir kapsülün sahip olduğu örnekleme parametreleri [84].

$$\sqrt{1.3^2 + 0.6^2 + 7.4^2 + 6.5^2 + 0.5^2 + 1.4^2} = 10.06 \quad (4.3)$$

Fakat bu hesaplanan uzunluk değerinin, ya da başka bir deyişle $6B$ vektörlerin veya kapsüllerin $[0,1]$ skalasında bir değer ifade etmesi adına, G. Hinton *ezme (squash)* adı verilen yeni bir aktivasyon fonksiyonu önermiştir (Eşitlik (4.4)). Bu sayede kapsül normalize edilmiş ve non-linear özellik kazandırılmış olur.

$$squash(u) = \frac{\|u\|^2}{1 + \|u\|^2} \frac{u}{\|u\|}; \quad 0 \leq P \leq 1 \quad (4.4)$$

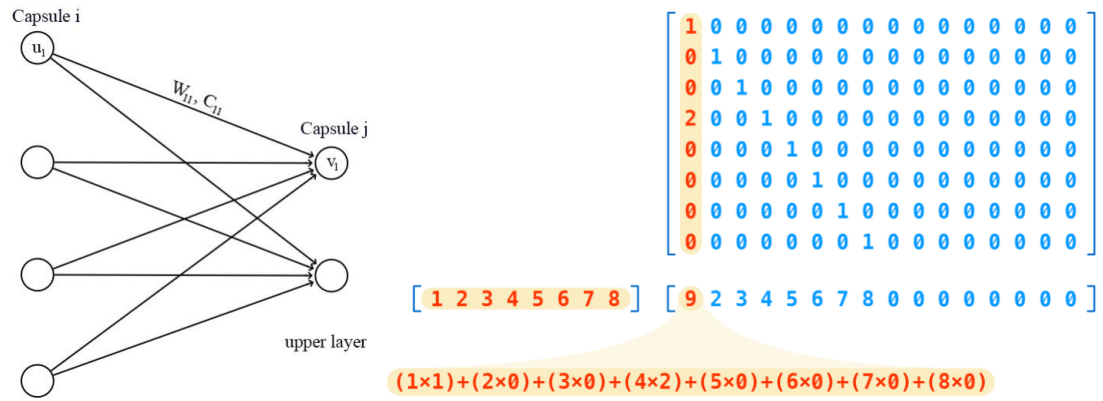
Bu fonksiyonun sağ bölümünde kapsülün birim vektör (*uzunluk = 1*) değeri hesaplanırken, sol tarafta ise değere fazladan ezme uygulanmaktadır. Eğer vektörün sahip olduğu uzunluk büyük ise hesaplanan değer '1' e, küçük ise '0' a yaklaşacaktır. Bu değer aynı zamanda aranan özelliğin kapsül tarafından tespit edilebilme olasılığını belirtmede kullanılır. Vektörün yönü değişse de olasılık değeri aynı kalacaktır. Böylece nöronsal ve konumsal değişikliklerden etkilenmeyen bir tanımlama sağlanır.

Diğer yandan, *PrimaryCaps* katmanı her biri 8 adet skaler büyüklüğe sahip kapsüllerin, yön bilgisine sahip vektörlere dönüştürüldüğü bölümdür. Bu amaçla, objenin koordinat düzleminde ifade edilmesini sağlayan ve afin transformasyon sağlayan ağırlık matrisleri kullanılır (Eşitlik (4.5)). Böylece düşük seviye ve yüksek seviyeli kapsüller arasındaki uzaysal ilişkilerin çıkarımı sağlanır. Bu matrisin sahip olduğu/olacağı değerler geri-yayılım ile güncellendiğinden, eğitim sonunda, tanımlanmak istenen objeyi bir araya getiren parçaların sahip olduğu özellikleri tutan

düşük seviye kapsüller kullanılarak, yüksek seviye kapsülde yer alacak objenin yeri, yönü vd. özelliklerini doğru bir şekilde tanımlamak mümkün olacaktır.

$$\hat{u}_{ji} = W_{ij} u_i \quad (4.5)$$

Bu süreçte kullanılan matrisin sahip olduğu satır sayısı düşük seviyeli kapsülün boyutu, kolon sayısı ise çıkışta üretilen vektörün boyutunu ifade eder. Kullanılan matrisinin boyutu ile doğru orantılı olarak oluşturulan mimarinin derinliği ve çıkış vektörlerinin boyutu da artacaktır. Bu da daha fazla özelliğin yorumlanabilmesi anlamına gelmektedir. Eğer girişte $8B$ vektörler kullanıyorsak, fakat çıkışta üretilen vektörün $16B$ olmasını istiyorsak, o zaman kullanılacak matrisin boyutu 8×16 olarak tanımlanmalıdır ($1 \times 8 \times 8 \times 16 = 1 \times 16$).



Şekil 4.16. Ağırlık matrisi ile vektörel özelliklerin kazandırılması [77,85].

Şekil 4.16'da görülen W_{11} birinci kapsül ile çarpılacak birinci ağırlık matrisini ifade etmektedir. C_{11} ise düşük seviye kapsülün, yüksek seviye kapsüle bağlanma olasılığıdır (daha sonra açıklanacak).

Dinamik Yönlendirme Algoritması

1152 kapsül ile 10 sınıf için gerçekleştirilecek 11520 tahminin tamamı her bir sınıf için anlamlı olmayacaktır. Bu sebepten, Kapsül Ağlarının ileri besleme sürecinde hangi düşük seviye kapsülün hangi yüksek seviyeli kapsüle bağlanacağını belirleyen

ve *dinamik yönlendirme algoritması* adı verilen bir prosedür kullanılmaktadır (Şekil 4.17).

Algorithm Dynamic routing algorithm.

```

1: procedure ROUTING( $\hat{u}_{j|i}, r, l$ )
2:   for all capsule  $i$  in layer  $l$  and capsule  $j$  in layer  $(l + 1)$  :  $b_{ij} \leftarrow 0$ .
3:   for  $r$  iterations do
4:     for all capsule  $i$  in layer  $l$  :  $c_i \leftarrow \text{softmax}(b_i)$ 
5:     for all capsule  $j$  in layer  $(l + 1)$  :  $s_j \leftarrow \sum_i c_{ij} \hat{u}_{j|i}$ 
6:     for all capsule  $j$  in layer  $(l + 1)$  :  $v_j \leftarrow \text{squash}(s_j)$ 
7:     for all capsule  $i$  in layer  $l$  and capsule  $j$  in layer  $(l + 1)$  :  $b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \cdot v_j$ 
return  $v_j$ 

```

Şekil 4.17. Dinamik yönlendirme algoritması [73].

Bu prosedür tüm düşük seviye katmanlarda (l) yer alan kapsüller ve çıkışları (\hat{u}) için, r iterasyon kadar çalışır ve sonuç olarak yüksek seviyeli kapsülün çıkış değerini döndürür. Prosedür sırasında kullanılan b_{ij} başlangıç değeri '0' olan geçici bir değişkendir ve her bir iterasyonda güncellenmektedir. Prosedür tamamlandığında ise, b_{ij} değerleri C matrisine depolanmaktadır. 3. satırda tanımlanan iterasyon sayısı, r , ise dinamik yönlendirme algoritmasının kaç kez çalışacağını belirten bir değerdir. Sabour vd. çalışmalarında bu sayıyı 3 olarak belirlemişlerdir ve fazla arttırılmasının ezberleme problemine yol açabileceğini belirtmişlerdir [73].

4. adımda ise, her bir düşük seviye kapsülün yüksek seviye kapsüllere bağlanma olasılığını ve ne kadar benzediğini tanımlayan c_{ij} değerleri hesaplanacaktır. 1152 düşük seviyeli kapsülün her bir yüksek seviyeli kapsül için gerçekleştireceği 10 adet tahminin toplamalarının '1' olması ve olasılık belirtmesi amacıyla c_{ij} değerlerinin hesaplamasında *softmax* fonksiyonu kullanılır (Eşitlik (4.6)).

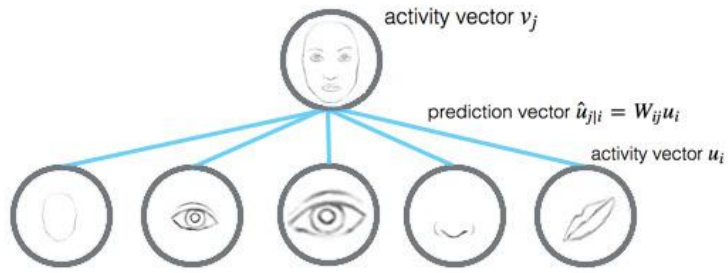
$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \quad (4.6)$$

Bir sonraki adımda klasik nöronların çıkış değerinin hesaplanmasına benzer şekilde her bir yüksek seviye kapsüle gelen düşük seviye kapsüllerin çıkış değerleri, c_{ij} ağırlıkları ile çarpılarak toplanır (Eşitlik (4.7)). Akabinde ise, elde edilen vektör ezme

aktivasyon fonksiyonu ile normalize edilir (Eşitlik (4.8)) ve böylece yüksek seviyeli kapsülün çıkış değeri hesaplanmış olur (Şekil 4.18).

$$s_j = \sum_i c_{ij} \hat{u}_{ji} \quad (4.7)$$

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \quad (4.8)$$



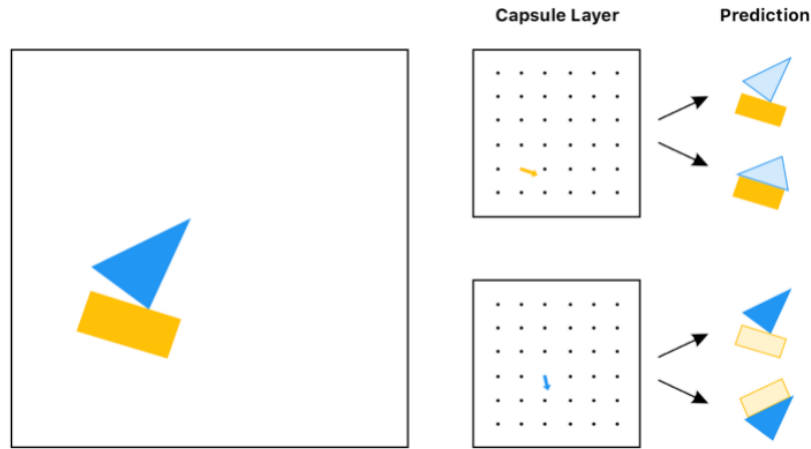
Şekil 4.18. Düşük-yüksek seviyeli kapsüller arasındaki bağıntı.

6. ve son adım ise, düşük seviyeli kapsüllerin ürettiği çıkış vektörleri ile yüksek seviyeli kapsüllerin sahip olduğu çıkış vektörlerinin değerlendirildiği ve ilgili güncelleştirmelerin gerçekleştirildiği bölümdür. Nokta çarpımın ürettiği sonuç doğrultusunda, iki vektör arasında pozitif bir yakınsama varsa, düşük seviyeli kapsüle ait olan b_{ij} değeri artırılır, negatif bir yakınsama (zıtlık) var ise azaltılır (eşleştirme etkisi). Böylece, r iterasyon sonrasında yapılacak güncellemeler ile, düşük ve yüksek seviye kapsüller arasındaki benzerliği ve eşleştirmeyi en iyi şekilde tanımlayacak sabitler elde edilecektir (anlaşma). Diğer yandan, dinamik yönlendirme algoritması sayesinde düşük seviyeli kapsüllerin çıkışları sadece benzer özelliklere sahip (benzer özelliklere sahip olmayanlar '0' a yaklaşacak) yüksek seviyeli kapsüle gönderileceğinden daha temiz giriş sinyallerine sahip olan yüksek seviyeli kapsüller elde edilecektir. Böylece, objeleri oluşturan bölümlerin hiyerarşik düzeni daha net incelenebilinerek, objeler ile ilgili daha doğru tahminler yürütülebilecektir. Prosedürün sonunda yüksek seviyeli kapsül (yüz) ile düşük seviyeli kapsüller (çehre, göz, ağız vs.) arasındaki hiyerarşik ilişki kurulmuş olur (Şekil 4.18).

Eşleştirme Etkisi (Coupling Effect)

Eşleştirme etkisinin tam olarak ne anlama geldiğinden biraz daha detaylı bahsederseniz; örneğin giriş verimizin Şekil 4.19 görüldüğü gibi bir bot görüntüsünün olduğunu farz edelim. Botu meydana getiren iki parçanın ise iki düşük seviye kapsül ile ifade edildiğini düşünebiliriz. Aynı zamanda bu kapsüllerin, birincil kapsül katmanında yer alan iki katmandan bir tanesinde botun alt kısmının, başka bir katmanda ise botun üst kısmının olasılık ve yön bilgisi tutan vektörler olarak ifade edildiğini düşünelim.

Sadece, bu kapsüllerin yaptığı tahminlere bakarak, eğer görüntünün ne olduğunu da bilmiyorsak, bir sonraki katmana hangi tahmin ile devam ederiz? Muhtemelen cevabımız 'bot' olacaktır. Çünkü iki kapsülde botun nasıl görüldüğü hakkında anlaşmışlardır. Fakat evin tam olarak neye benzediği konusunda bir ihtilaf söz konusudur.

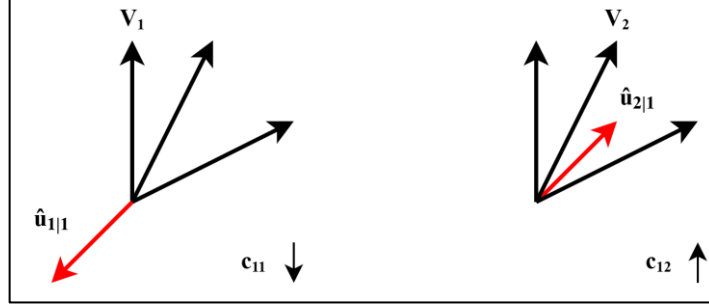


Şekil 4.19. Kapsül Ağlarında eşleştirme etkisi [77].

Kapsüllerin benzerlik oranları hesaplanırken vektörler arasında nokta çarpım gerçekleştirilir. Bu çarpıma göre elde edilen değer negatif ise düşük seviyeli kapsülün çıkış vektörüne benzemediği, pozitif ise benzediğine karar verilir. Buna göre kapsüller arasındaki c_{ij} bağlantısının değeri güncellenir.

Şekil 4.20'de birinci örnekteki V_1 'in yüksek seviye kapsül, \hat{u}_{11} 'in ise düşük seviye kapsül olduğunu farz edelim. Buna göre gerçekleştirilecek nokta çarpım sonrası

aradaki bağlantı değeri (c_{11}) azaltılacakken, 2. örnekte benzerlik söz konusu olduğundan ilgili değer (c_{12}) yükseltilecektir. Zamanla, alakasız kapsüllerin bağlantı değeri 0'a yaklaşacak ve ağ tarafından göz ardı edilmeye başlayacaktır [88].



Şekil 4.20. Eşleştirme sabitlerinin güncellenmesi.

Güncellemelerin akabinde, düşük seviyeli ve yüksek seviyeli kapsüllerin çıkışları arasında en iyi eşleşmeleri sağlayacak sabitler elde edilecektir.

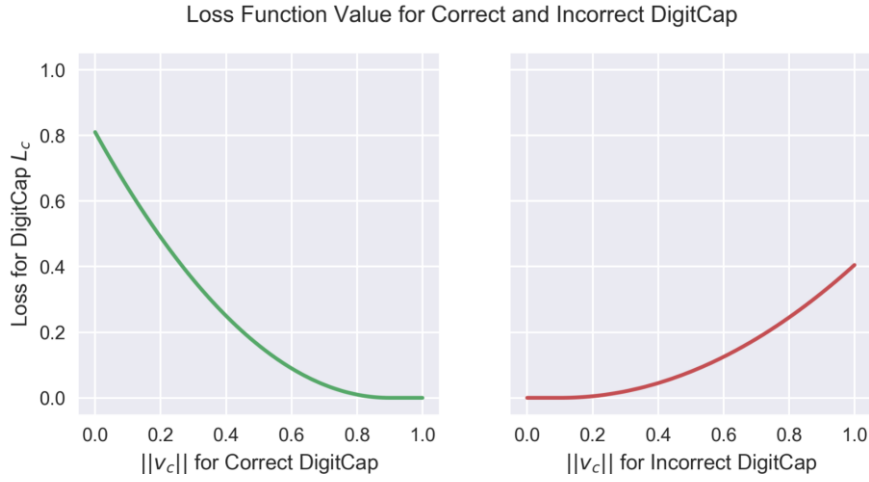
DigitCaps katmanı ise, her biri $16B$ olan 10 adet (sınıf sayısı kadar) yüksek seviye kapsülü (vektör) barındıran bölümdür. Bu vektörlerin ürettiği tahmin değerleri ve *kayıp* fonksiyonu kullanılarak, ağın tüm çıkışlar için ürettiği toplam hata hesaplanmaktadır. Bu hata değeri sayesinde de ağın geri yayılımı sırasında kullanılan optimizasyon fonksiyonuna göre ağırlık değerlerinin (dönüşüm matrisi) güncellenmesi sağlanır. Ağın kodlayıcı bölümünün sonunda her bir çıkış için hesaplanan kayıp değerinde Eşitlik (4.9) kullanılır.

$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda (1 - T_k) \max(0, \|v_k\| - m^-)^2 \quad (4.9)$$

Burada; doğru sınıflar için $T_k = 1$, diğer sınıflar için $T_k = 0$, $m^+ = 0.9$, $m^- = 0.1$ ve $\lambda = 0.5$ değerini almaktadır.

Eğitim sırasında her bir örnek için kullanılan hedef vektörü $10B$ bir *one-hot encoding* vektörüdür (Doğru sınıf için 1, diğer sınıflar için 0). Her bir görüntü için kayıp değeri hesaplanırken doğru etikete sahip örnekler için eşitliğin sol tarafından gelecek değer, yanlış sınıflar için ise sağ taraftan gelen değerler kullanılır.

Eğer görüntü %90 ve üzeri bir sınıflandırma başarısına sahip ise kayıp değeri “sıfır”, değil ise “sıfırdan farklı bir değer” olacaktır. Diğer taraftan eğer görüntünün yanlış sınıfa ait olduğu %10 ve altı bir oran ile tahmin edildiyse kayıp değeri “sıfır”, aksi halde “sıfırdan farklı bir değer” olacaktır. Formülde bulunan λ değeri nümerik stabilitenin sağlanabilmesi için kullanılırken, karelerinin alınması ile de L_2 normalizasyonu sağlanmaktadır. Eğitim sonunda ise, eğitim verilerini en az %90 ve üzeri başarı ile öğrenmeye çalışan bir model elde edilmiş olacaktır.



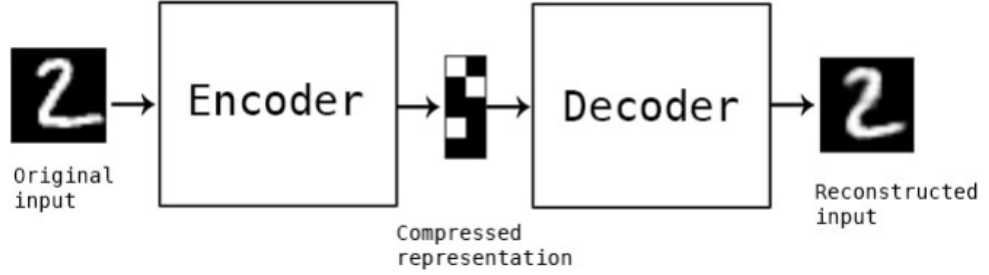
Şekil 4.21. Doğru ve yanlış tahmin edilen için yüksek seviye kapsüller için çizilen kayıp değerleri [89].

Şekil 4.21’de doğru ve yanlış tahmin edilen yüksek seviye kapsüller için çizilen kayıp değerleri grafiğinde de görüleceği üzere, kayıp fonksiyonu doğru tahmin edilen çıkış kapsüllerinin olasılık değeri arttıkça azalan, diğer yandan, yanlış tahmin edilen kapsüllerin olasılık değeri arttıkça artan bir eğilimdedir. Yanlış tahmin edilen kapsüllerin kayıp değerinin, doğru tahmin edilene göre dikey olarak daha ezilmiş olmasının nedeni fonksiyonda kullanılan λ ’dan kaynaklanmaktadır.

4.3.9.2. Çözücü

Eğer, bize birkaç kedi görüntüsü gösterilirse ve bunları tekrar çizmemiz istenirse muhtemelen görüntülerde sadece kedi olduğunu hatırladığımızdan, bir kare içerisine ortalanmış bir şekilde kedi çizersiniz. Fakat çizilen kedilerin orijinal görüntülere yakın olması istenirse ne yaparsınız? O zaman kedinin tam olarak nerede olduğu ve

boyutuyla ilgili bilgileri de hatırlayarak buna göre bir çizim yapmaya çalışırız. Kapsül Ağlarının çözücü kısmının amacı da tam olarak budur [75].



Şekil 4.22. Çözücü bölümü ile görüntülerin yeniden inşa edilmesi [86].

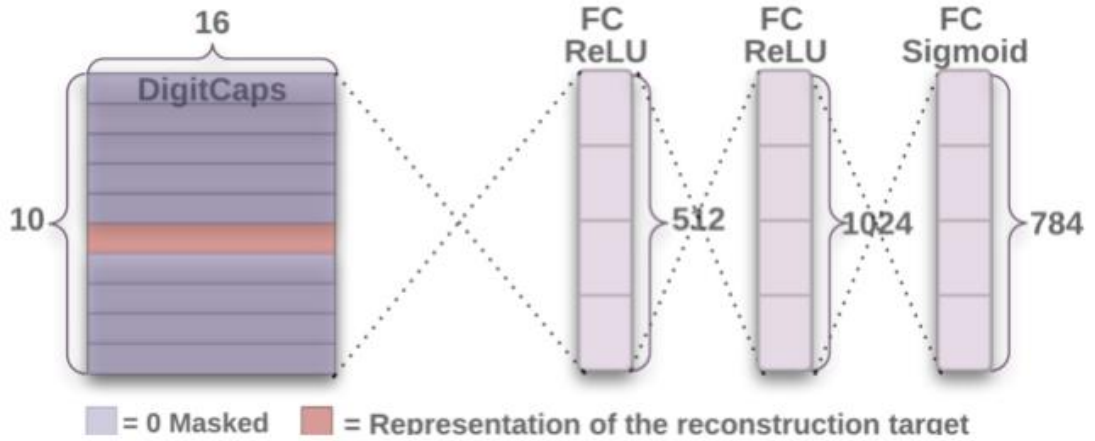
Ağın çözücü kısmının direkt olarak sınıflandırmaya katkısı yoktur. Yazarlar çalışmalarında bu bölümü kullanarak sınıflandırmaya düzenleyici bir etki sağlamayı, böylece daha kararlı bir model oluşturmayı amaçlamışlardır. Varyasyonel Oto-Kodlayıcıların çalışmasına benzeyen bu bölümde, sadece eğitim sonunda elde edilen doğru sınıflandırılmış *DigitCap* vektörü kullanılarak ve diğerleri maskelenerek, 28×28 boyutuna sahip MNIST giriş verisi tekrar oluşturulmaya çalışılmaktadır (Şekil 4.22). Bu bölümün sınıflandırma sürecine sağladığı en büyük katkı, hesaplanan hata değerlerinin toplam kayıp değerinde kullanılması ile her bir yüksek seviye kapsülü giriş verisinden elde edebileceği maksimum çıkarım yapmaya zorlamasıdır. Bu sayede, model görüntülerin sınıfları arasında ayırım yapmayı sağlayabilecek değişkenleri de öğrenebilecektir.

Çözücü bölümünün eğitimi düzenleyici bir etkiye sahip olması sebebiyle, Kapsül Ağları tüm obje ve objenin pozları ile ilgili bilgileri tanımlayan global değerleri danışmansız olarak öğrenecektir. Bu bölümden elde edilen hata değeri de toplam hatada kullanıldığından, çıkış kapsüllerinin giriş görüntüsünden elde edilebilecek en faydalı bilgiyi edinmesine zorlayacaktır. Oluşturulan görüntü orijinale ne kadar benzerse, o kadar iyi bir sınıflandırma sağlanacaktır. Bu şekilde farklı vektörler aracılığıyla görüntülerin farklı örnekleme parametrelerine sahip örnekleri de doğru şekilde oluşturabilecektir. Şekil 4.23'te görüldüğü gibi aynı rakamların farklı pozisyonları, kalınlığı, boyu gibi değişkenlik gösteren özelliklerinin de çıkarımı sağlanabilir.

Scale and thickness	6 6 6 6 6 6 6 6 6 6 6
Localized part	6 6 6 6 6 6 6 6 6 6 6
Stroke thickness	5 5 5 5 5 5 5 5 5 5 5
Localized skew	4 4 4 4 4 4 4 4 4 4 4
Width and translation	3 3 3 3 3 3 3 3 3 3 3
Localized part	2 2 2 2 2 2 2 2 2 2 2

Şekil 4.23. Kapsüller sayesinde objelerin farklı özelliklerinin algılanması ve yeniden oluşturulması [75].

Orijinal makalede kullanılan çözücü bölümünün (Şekil 4.24) ilk kısmında, $16B$ vektörler 512 tane nörona, daha sonra ise 1024 nörona tam bağlanmıştır. Bu iki katmanda da aktivasyon fonksiyonu olarak *ReLU* tercih edilmiştir. Son katmanda ise, bir önceki katmandan gelen 1024 çıkış, oluşturulmak istenen görüntünün sahip olduğu piksel sayısı kadar ($28 \times 28 = 784$) nörona tam bağlıdır. Fakat burada diğer katmanların aksine, çıkışın $[0,1]$ aralığında bir piksel değeri ifade etmesi amacıyla *sigmoid* aktivasyon fonksiyonu kullanılmıştır.



Şekil 4.24. Kapsül Ağlarının çözücü bölümü [73].

Elde edilen çıkış değerleri yeniden şekillendirildikten sonra 28×28 çözünürlükte *gri ölçekli* bir görüntü elde edilir. Bu görüntünün ve giriş görüntüsünün arasında hesaplanan ortalama karelerin hatası (MSE) farkı ile de *yeniden inşa kayıp değeri* ifade edilir (Eşitlik (4.10)).

$$\text{decoder loss} = \| \text{image} - \text{reconstructed image} \| \quad (4.10)$$

Ağın kodlayıcı kayıp değeri ve bu çözücü kayıp değerinin toplanması ile tüm ağın bir görüntü için ürettiği toplam kayıp değeri hesaplanmış olur (Eşitlik (4.11)).

$$\text{Loss} = \text{encoder loss} + \alpha \text{ decoder loss} \quad (4.11)$$

Formüldeki $\alpha = 0.0005$ seçilerek eğitim sırasında hesaplanacak yeniden inşa kayıp değerinin kodlayıcı kayıp değerini domine etmesi engellenir ve eğitimde düzenleyici bir etki sağlanır [73].

4.3.10. Kapsül Ağlarının Avantajları ve Dezavantajları

Kapsül Ağlarının genel olarak avantajlarını ve dezavantajlarını sıralarsak;

Avantajları:

- Daha az veri ile başarılı bir şekilde çalışabilir,
- Daha temiz giriş sinyaline sahiptir,
- Pozisyon ve poz bilgisi korunur,
- Üst üste binen görüntüleri ayırabilir,
- Objeyi oluşturan parçaların hiyerarşisi doğru şekilde yorumlanabilir,
- Afın dönüşümlerinde kararlıdır,
- Aktivasyon vektörlerini yorumlamak kolaydır (döndürme, kalınlık, eğrilik vb.).

Dezavantajları:

- Çok fazla eğitim parametresi vardır,
- Yavaş çalışır (Dinamik yönlendirme iç içe döngülerinden dolayı) [90].

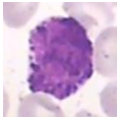
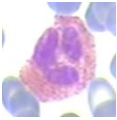
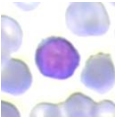
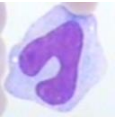
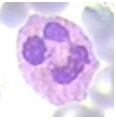
4.4. VERİ SETİ

Vücudumuzda dolaşan kanın 1 mikro litresinde 4,000 ile 11,000 arasında değişen WBC bulunur. Bu hücrelerin yaklaşık yüzde %40-%70'i *nötrofil*, %20-%45'i *lenfosit*, %2-%10'u *monosit*, %1-%6'sı *eozinofil* ve <%1 ise *bazofildir* [2]. WBC'lerin alt-tiplerinin, kan örneğinde bulunma oranlarının yüksek farklılıklara sahip olması, ML

ile kararlı bir analiz yapmayı zorlaştıran etmenlerin başında gelir. Çünkü, sınıflandırılması gerçekleştirilecek veri setindeki her bir sınıfa ait görüntülerin sayıca birbirine yakın olması gerekmektedir. Aksi halde, *dengesiz veri seti problemi* adı verilen durum ortaya çıkar ve tüm sınıfları kapsayan doğru bir sınıflandırma modeli elde edilemez [91]. Diğer yandan, ezberlemenin önüne geçilebilmesi adına kullanılan verilerin sınıflar arasında ayırım yapmayı kolaylaştıracak ve sınıfların spesifik özelliklerini yansıtacak özgün veriler olması gerekmektedir.

Analizi gerçekleştirilecek veri seti seçilirken yukarıda bahsedilen detaylar da göz önünde bulundurularak, LISC [51] veri seti tercih edilmiştir. Veri setinde yer alan 720×576 çözünürlükteki görüntülerden, 128×128 boyutlarında ve her bir görüntüde bir adet WBC alt-tipi bulunacak şekilde kesitler alınmıştır. Veri setinde yer alan görüntülerden bazı örnekler ve veri setinin diğer özellikleri Çizelge 4.2’de verilmiştir.

Çizelge 4.2. Kullanılan veri setinin özellikleri.

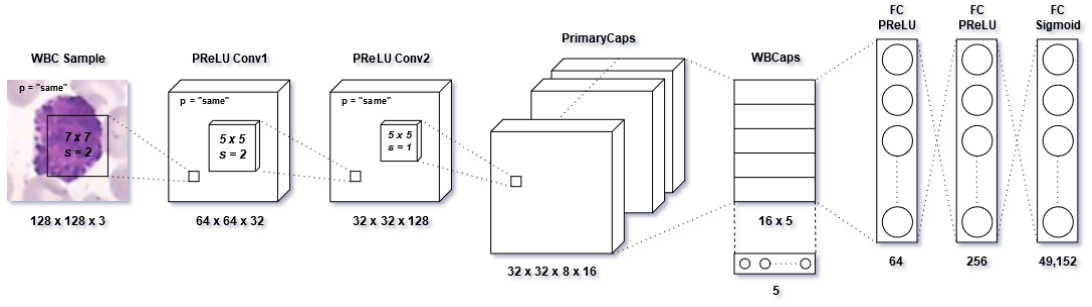
WBC alt-tipi	Bazofil	Eozinofil	Lenfosit	Monosit	Nötrofil
Örnek Görüntü					
Çözünürlük	128x128				
Kanal Sayısı	3				
Örnek Sayısı	58	42	59	48	56
Eğitim Seti	49	36	50	43	45
Test Seti	9	6	9	5	11

Veri setinden çıkarılan 263 örnekten, %85’i eğitim (223), %15’i test (40) olacak şekilde ikiye bölünmüştür. Eğitime başlamadan önce görüntülerin her bir kanalında bulunan piksel değerleri 255’e bölünerek [0,1] aralığında normalize edilmiştir. Böylece ağın daha hızlı bir şekilde *yakınsama (converge)* durumuna gelmesi amaçlanmıştır [92]. Diğer yandan görüntülerin doğru sınıflarına ait hedef vektörleri *one-hot encoding* yöntemine göre oluşturulmuştur. Daha sonra ise inşa edilen model ile eğitim aşamasına geçilmiştir.

4.5. ÖNERİLEN MODEL (WBCaps)

Kapsül Ağlarının CNN'e göre birçok avantajı bulunmasına rağmen, ahip olduğu en önemli dezavantaj eğitilecek parametre sayısının oldukça fazla olmasıdır. Orijinal makalede kullanılan mimaride giriş verisi olarak kullanılan görüntülerin boyutu oldukça küçük olmasına ve az katmanlı bir mimari kullanılmasına rağmen $\sim 8.2M$ eğitim parametresi barındırmaktadır. Bu parametrelerin yaklaşık $\sim 5M$ tanesi yeniden şekillendirme ve dinamik yönlendirmenin gerçekleştiği *PrimaryCaps* katmanından kaynaklanmaktadır. Dolayısıyla, bu katmanda işlenecek görüntünün boyutunun yüksek olması eğitilecek parametre sayısının da aynı derecede yüksek olmasına yol açacaktır.

Kapsül Ağlarının yeteneklerini smallNORB [93] veri seti üzerinde test eden Sabour vd. ve Kapsül Ağları ile beyin tümörü sınıflandırması gerçekleştiren Afshar vd. bu problemin üstesinden gelebilmek adına veri setlerinde yer alan görüntülerin boyutunu 32×32 çözünürlüğe düşürmüşlerdir [73,94]. Fakat daha önce de bahsedildiği üzere medikal görüntülerin her bir pikseli anlamlı veri taşıyabilmektedir. Bu yüzden görüntülerin boyutunun azaltılması, önemli piksellerin yok olmasına neden olabilmektedir. Bu nedenle çalışmamızda kullandığımız 128×128 çözünürlüğe sahip görüntülerin boyutunu azaltmak amacıyla modelin başına bir konvolüsyon katmanı daha eklenmiştir. Eklenen katmanda kullanılan filtrelerin *stride* değeri ise 2 olarak belirlenmiştir. Bu sayede veri kaybının önüne geçilerek eğitim parametresi sayısı azaltılmıştır. Diğer yandan, eğitim setinde yer alan görüntülerin sayısının az olması nedeniyle, analizi kolaylaştıracak, görüntülerin ayrımının daha kolay yapılabilmesini sağlayacak, bir nevi her bir görüntüyü nitelikli olarak tanımlayacak *parmak izleri* (aktivasyon haritaları) ile modelin çalışması sağlanmıştır. (Şekil 4.25). Böylece, *PrimaryCaps* katmanından önce verilerin boyutu 32×32 seviyesine indirgenebilmiştir. Ayrıca, *padding = same* seçilerek kenarlarda bulunan piksellerden de maksimum çıkarım yapılması amaçlanarak reseptif alan artırılmıştır.



Şekil 4.25. Önerilen mimari.

Diğer taraftan çalışmamızda kullandığımız eğitim verisinin az olması nedeniyle, modelimizde kullanılan konvolüsyon filtrelerinin sayısı daha anlamlı çıkarımlar sağlanabilmesi ve aktive olmayan konvolüsyon katmanı oluşmaması adına düşük tutulmuştur. Diğer yandan, ReLU fonksiyonunun *ölen nöron (dying neuron)* problemine yol açması nedeniyle konvolüsyon ve tam bağlı katmanlarda *PReLU* [95] aktivasyon fonksiyonu kullanılmıştır.

Son konvolüsyon katmanının derinliği 128 olduğundan ve 8B vektörler oluşturulduğundan dinamik yönlendirme prosedürüne geçen *PrimaryCaps* katmanı sayısı $128/8 = 16$ olmuştur. *PrimaryCaps* katmanında yer alan düşük seviyeli kapsüllerin sayısı ise $32 \times 32 \times 16 = 16,384$ 'dür. WBC alt-tiplerinin sınıflandırılması, karakter tanımaya göre daha komplike bir görev olduğundan daha fazla spesifik özelliğin tanımlanabilmesi için düşük seviye kapsül sayısı çalışmamızda bu denli yüksek tutulmuştur.

Sınıflandırmanın gerçekleştirildiği yüksek seviyeli kapsül katmanı (*WBCaps*) ise, sınıf sayısı (5) kadar 16B kapsülden oluşturulmuştur. Ağın çözücü kısmı ise sırasıyla 64 ve 256 nöron barındıran tam bağlı katmanlardan oluşmaktadır. Giriş verisi olarak verilen görüntünün yeniden inşasını sağlayacak son katman ise $128 \times 128 \times 3 = 49,152$ nöron barındırmaktadır.

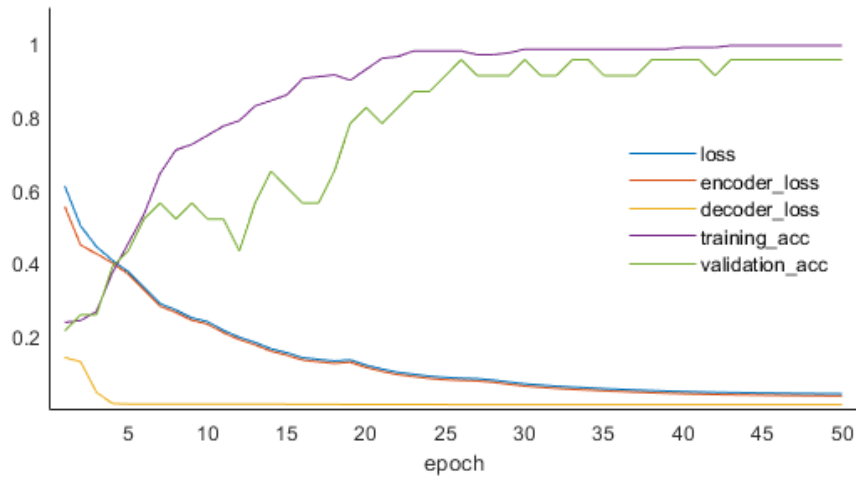
Eğitim sürecinde kullanılan önemli hiper-parametrelerden biri de geri yayılım sırasında gerçekleştirilecek güncellemelerin hesaplandığı optimizasyon algoritmasıdır. Burada, ağın sahip olduğu parametrelere göre *adaptif* bir güncelleme

sağlayarak *yerel minimum (local minimum)* probleminin oluşmasını engellemeye çalışan *ADAM* (Adaptive Momentum Estimation) [96] fonksiyonu tercih edilmiştir.

Ayrıca, modelimizin eğitim sürecinde *k-fold çapraz doğrulama* kullanılmıştır. *k* değeri 10 seçilerek modelin 223 görüntününün 200 tanesi ile eğitim, 23 tanesi ile doğrulama gerçekleştirilmesi sağlanmıştır. Daha sonra, her bir devirde eğitim ve doğrulama verileri karıştırılarak, eğitim verisinin ezberlenmesi engellenmiş ve kararlı bir sonuç üretilmesi amaçlanmıştır [97]. Toplamda $\sim 23.5M$ eğitim parametresine sahip olan model 50 devir (*epoch*) sayısı için eğitilmiştir.

4.6. DENEYSEL SONUÇLAR

Ağın her bir devri için hesaplanan *kayıp (loss)*, *kodlayıcı kaybı (encoder loss)*, *çözücü kaybı (decoder loss)*, *eğitim doğruluğu (trainin accuracy)* ve *tasdik doğruluğu (validation accuracy)* değerlerini içeren grafik Şekil 4.26’da görülebilir. Daha önce de bahsedildiği üzere, modelin sınıflandırmadan sorumlu olan bölümünün ağın toplam başarısında en yüksek role sahip olduğu ve çözücü kısmında hesaplanan kayıp değerinin düzenleyici bir etkisi olduğu açıkça görülmektedir. Diğer yandan ağın 27. devirde yakınsama durumuna geçtiği görülmüştür. Modelin ulaştığı en yüksek doğrulama başarısı ise $22/23 = 0.9565$ olarak ölçülmüştür. Eğitim sonlandırıldıktan sonra, modelin daha önce hiç görmediği test verileri üzerindeki sınıflandırma performansı ise $37/40 = 0.9250$ olarak gözlemlenmiştir.



Şekil 4.26. Ağın eğitim ve doğrulama değerleri.

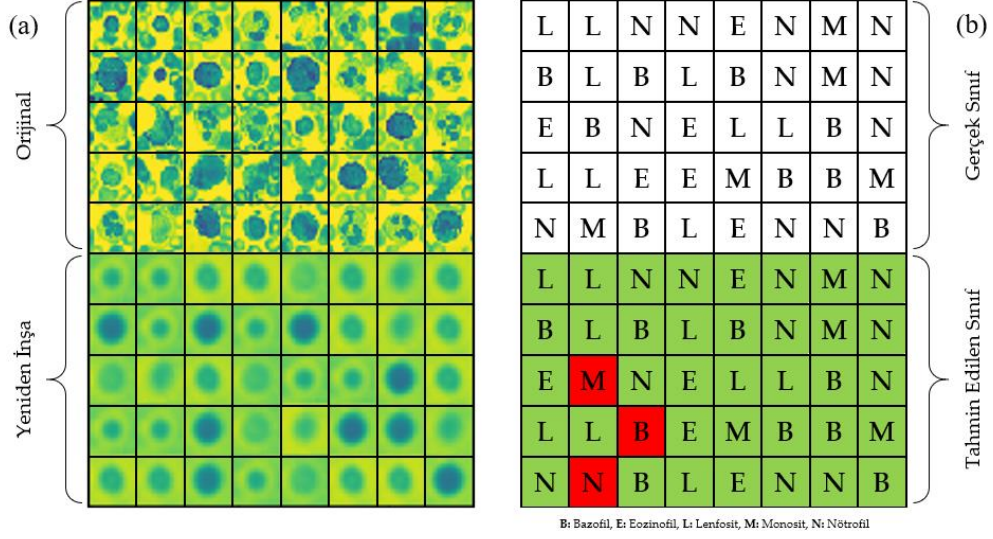
Kapsül Ağlarının sağladığı avantajlardan biri de yeniden inşa edilen görüntülerin görsel olarak incelenebilmesine, böylece, orijinal görüntülere ne kadar yaklaştığı ve eğitimin ne kadar başarılı olduğunun analiz edilebilmesine imkân tanınmasıdır. Yeniden inşa edilen görüntüler incelendiğinde, kullanılan veri setinin yetersiz sayıda olması nedeniyle test verilerine pek de yaklaşmadığı görülmüştür (Şekil 4.27). Diğer yandan, WBC alt-tiplerinin sınıflandırılması probleminin, orijinal makalede çözümlenmesi gerçekleştirilen karakter tanıma problemine nazaran daha kompleks olması, görsel manada yetersiz görüntüler oluşmasına neden olmuştur. Fakat burada dikkat edilmesi gereken detay, test edilen 40 görüntünün aynı sınıflara ait örneklerinin, benzer şekilde oluşturulmuş olmasıdır (Şekil 4.27-a).

Görüntülerdeki sarı-yeşil görünüm ise normalizasyon sürecinde yapılan veri tipi dönüşümlerinden kaynaklanmaktadır. Fakat, görüntüler bu dönüşümlere aynı oranda maruz kaldığından renk kanallarında herhangi bir veri kaybının yaşandığı ya da hücrelerin preferik yayma sırasındaki kullanılan boyama sürecine verdiği farklı tepkilerin çıkarımlarının yapılamamasının söz konusu olduğu söylenemez.

Şekil 4.27-b’de verilen sınıf haritası incelendiğinde ise, üç örnek dışında aynı sınıflara ait orijinal görüntülerin yeniden yapılandırılmış versiyonlarının benzer yapıda olduğu görülmüştür. Ayrıca, yeniden inşa edilen görüntüler, kapsüllerin WBC’lerin hangi özelliklerine yoğunlaştığı hakkında da fikir sahibi olmamızı sağlamıştır. Örneğin, görsel olarak hiç sitoplazması bulunmayan *bazofil*, büyük ve koyu dolgulu bir daire olarak resmedilmişken, bazofilden daha küçük boyutlarda olan ve az miktarda sitoplazması bulunan *lenfosit*, etrafında açık renkte bir halka bulunan küçük dolgulu bir daire olarak gösterilmiştir. Model *monosit* ve *nötrofil* hücrelerini açık renkte bir arka plana yerleştirmiş, nötrofili dairesel ama monositi düzensiz bir şekilde tasvir etmiştir. Son olarak, sitoplazması monosit ve nötrofilden az, ama bazofilden ve lenfositten fazla olan *eozinofil*, koyu renkte bir arka planda yer alan, fakat, nispeten açık renkte bir daire olarak görülmektedir.

Bu bilgiler ışığında, kapsüllerin WBC alt-tiplerinin sahip olduğu sitoplazma oranlarına ve şekil karakteristiklerine yoğunlaştığı sonucuna varmak mümkündür. Özetle, yeniden inşa süreci görsel manada başarısız olsa da eğitim sırasında kodlayıcı




bölümünün farklı sınıflara ait ayırt edici özellikleri doğru bir şekilde öğrenebildiği söylenebilir.



Şekil 4.27. a) Orijinal test verileri ve yeniden inşa edilmiş halleri, b) Görüntülerin gerçek ve tahmin edilen sınıfları.

Sonraki aşamada, doğru ve yanlış sınıflandırılan verilerin, modelin toplam başarısına olan etkisinin daha detaylı incelenebilmesi adına karmaşıklık matrisi oluşturulmuştur (Çizelge 4.3). Aynı zamanda, yanlış sınıflandırılan görüntüler daha detaylı incelenerek, modelin başarısını olumsuz etkileyen faktörler irdelenmiştir.

Çizelge 4.3. Karmaşıklık matrisi ve yanlış tahmin edilen WBC alt-tipleri.

Gerçek Sınıf	WBC alt-tipi	Tahmin Edilen				
		Bazofil	Eozinofil	Lenfosit	Monosit	Nötrofil
	Bazofil	8	0	0		0
	Eozinofil		5	0	0	0
	Lenfosit	0	0	9	0	0
	Monosit	0	0	0	4	
	Nötrofil	0	0	0	0	11

Yanlış sınıflandırılan bazofil ve eozinofil görüntüleri incelendiğinde, bu hücrelerin veri setinde yer alan diğer örneklere nazaran farklı ve deforme olmuş hücreler oldukları söylenebilir. Fakat, monosit hücresinde model açısından tam bir başarısızlık söz konusu olmuştur. Eozinofil ve monosit örneklerinin yanlış sınıflandırılmasına yol açan bir diğer önemli faktör ise ilgili iki sınıfın eğitim verilerinin diğer sınıflara nispeten daha az olması verilebilir.

Sonraki aşamada, oluşturulan karmaşıklık matrisi aracılığı ile modelin sınıf bazında ve genel olarak başarısını değerlendirebileceğimiz *duyarlılık (sensitivity)*, *özgüllük (specificity)*, *hassasiyet (precision)* ve *doğruluk (accuracy)* gibi istatistiksel ölçütler hesaplanmıştır (Çizelge 4.4).

Çizelge 4.4. Sınıflandırma raporu.

WBC alt-tipi	Duyarlılık (%)	Özgüllük (%)	Hassasiyet (%)	Doğruluk (%)
Bazofil	88.89	96.67	88.89	94.87
Eozinofil	83.33	100.00	100.00	97.37
Lenfosit	100.00	100.00	100.00	100.00
Monosit	80.00	97.06	80.00	94.87
Nötrofil	100.00	96.30	91.67	97.37
Ort./Total	92.50	98.01	92.50	96.86

Oluşturduğumuz modelin yüksek bir sınıflandırma doğruluğu (%96,86) elde ettiğini söylemek mümkündür. *Eozinofil* ve *monositin* duyarlılık değerlerinin ve yine, *monositin* hassasiyet değerinin düşük olmasının başlıca sebebi ise, kullanılan test verisindeki örnek sayısının az olmasıdır. Dolayısıyla, bir örneğin dahi yanlış sınıflandırılması sonuçlar üzerinde %20'ye varan olumsuz bir etkiye neden olmuştur.

4.6.1. Sonuçların Diğer Yöntemler ile Kıyaslanması

Tez çalışmasının bu bölümünde Kapsül Ağları kullanılarak elde edilen başarı oranları, *CNN* ve *Transfer Öğrenme* yöntemlerinden elde edilen sonuçlarla kıyaslanmıştır.

Daha önce de bahsedildiği üzere, DL yöntemlerinin başarısı veri setinin boyutuyla orantılı olarak artmaktadır. Fakat, DL literatüründe, veri setlerinin boyutunun küçük olduğu durumlarda da başarılı bir sınıflandırma gerçekleştirilebilmesini sağlayacak bazı metotlar mevcuttur. Bunlardan bir tanesi de *transfer öğrenme* yöntemidir.

Bu yöntemde, daha önce farklı bir sınıflandırma görevinde kullanılan bir model özellik çıkarıcı olarak kullanılarak, yeni veri setinin sınıflandırılması sağlanır. Bunun için, modelin özellik çıkarıcı katmanlarında yer alan ağırlık değerleri korunarak, yeni veri seti ile ileri besleme gerçekleştirilir ve özellik çıkarımları yapılmış vektörler elde edilir. Son aşamada ise elde edilen özellik vektörleri, sınıf sayısı kadar çıkışa sahip *softmax* veya *SVM* katmanına bağlanarak sınıflandırma gerçekleştirilir. Böylece, daha önce farklı bir veri seti için öğrenilmiş olan özelliklerin ve sınıflandırma yeteneklerinin, başka veri setlerinin de başarılı bir şekilde sınıflandırılabilmesi için kullanılması mümkün olmaktadır.

Bir diğer yöntem olan, *ince-ayarda* ise, modellerde yer alan bazı katmanların sahip olduğu parametreler korunurken, diğerlerinin (veya tüm katmanların) ağırlıklarının her bir devirde tekrar eğitilerek güncellenmesi sağlanır ve görüntülerin özellik çıkarımı gerçekleştirilir. Böylece, modelin ilk bölümlerinde temel özellik çıkarımları gerçekleştirilirken, sonraki katmanlarda ileri beslemesi gerçekleştirilen yeni görüntülerin daha karmaşık kısımlarına odaklanabilen bir model oluşturulması hedeflenmektedir. Dolayısıyla, yeni veri setine hızlıca adapte olabilen bir sistem elde edilir [98]. Bu yöntemlere ek olarak CNN ile *en-baştan-öğrenme (training from scratch)* ise, sıfırdan yeni bir CNN modeli oluşturularak bir veri seti ile eğitim ve test gerçekleştirilmesidir.

Sınıflandırılması gerçekleştirilecek veri seti üzerinde transfer öğrenme, ince ayar veya en-baştan-öğrenme metotlarından hangisinin tercih edileceği Şekil 4.28'de görülebileceği üzere iki kriter üzerinden değerlendirilebilir. Eğer elimizdeki veri seti boyutu büyük ve kullanılacak model daha önce benzer bir veri setinde kullanılmışsa, ince-ayar kullanmak eğitim verisinin ezberlenmesine yol açmayacak ve başarılı bir sınıflandırma gerçekleştirilebilecektir. Eğer veri seti boyutu büyük fakat, kullanılan veri setine benzer bir veri üzerinde daha önce eğitilmiş bir model yoksa, o zaman

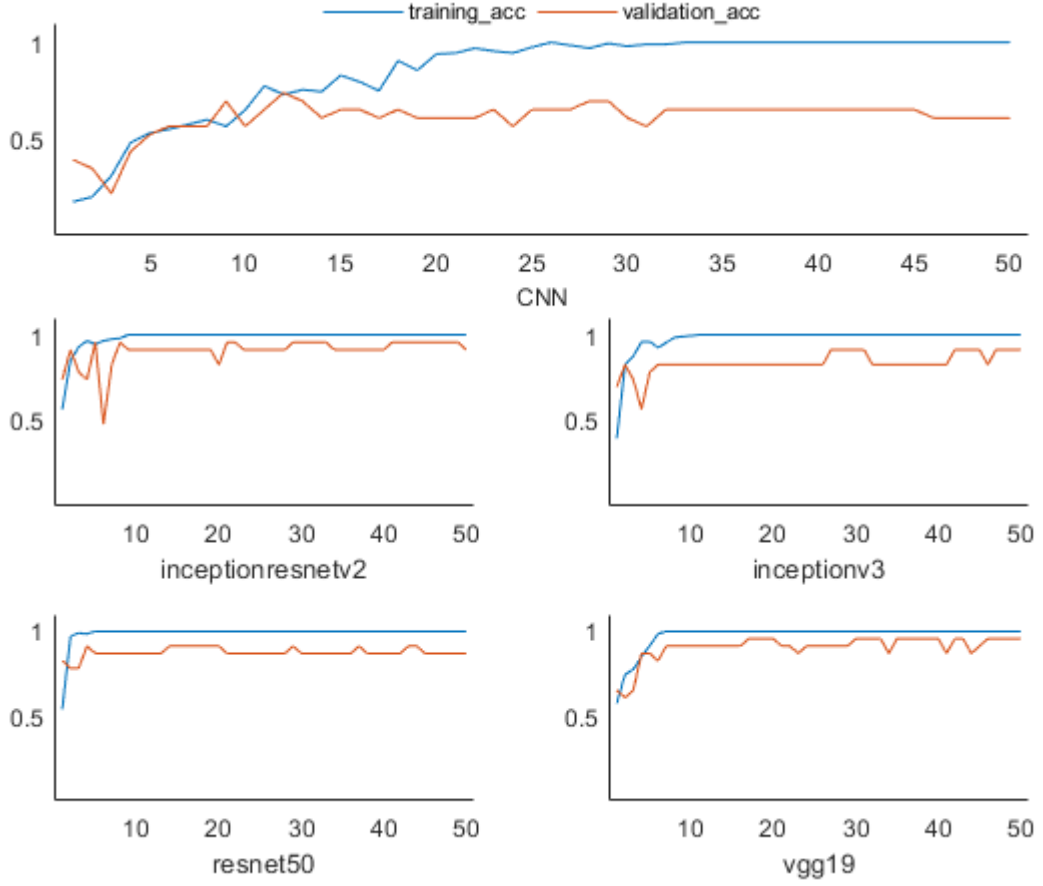
transfer öğrenme ve ince-ayar yöntemi yerine sıfırdan bir CNN modeli inşa edilerek eğitim gerçekleştirilmesi önerilebilir. Diğer yandan, veri setinin küçük ve orijinal veri setine benzer özellikte olduğu durumlarda, transfer öğrenme ile yeni verilerin özellik çıkarımlarının gerçekleştirilmesi ve sadece modelin sonunda bulunan sınıflandırıcı kısmının eğitilmesi ile modelin karmaşık özelliklerin çıkarımını yaptığı bölümdeki yeteneklerini kullanmak mantıklı olacaktır. Fakat, veri seti küçük ama orijinal veri setine benzer özelliklere sahip değilse, o zaman modelin ilk bölümünde yer alan bazı katmanlarındaki parametreler korunarak, son bölümde yer alan katmanların ve sınıflandırıcının eğitilmesi daha yüksek bir başarı elde edilebilmesi için mantıklı olacaktır. Aksi takdirde, küçük bir veri seti ile gerçekleştirilecek bir ince-ayar süreci, eğitim verilerinin ezberlenmesine yol açacaktır [99].

		Benzerlik	
		<i>Benzer</i>	<i>Benzer değil</i>
Veri seti boyutu	<i>Büyük</i>	İnce-ayar	Baştan öğren
	<i>Küçük</i>	Transfer Öğrenme (LS)	Transfer Öğrenme

Şekil 4.28. Veri setinin benzerliğine ve boyutuna göre yöntem seçimi.

WBC sınıflandırılmasında kullanılan veri setinin sayıca düşük olması nedeniyle, gerçekleştireceğimiz testlerde transfer öğrenme metodu kullanılmıştır. Diğer yandan, kullanılacak veri seti ile, modellerinin eğitiminin gerçekleştirildiği veri seti arasında benzerlik olmamasından dolayı, modellerin yalnızca sınıflandırıcısı değil, sınıflandırıcıdan önce yer alan iki katmanda bulunan parametrelerde eğitime dahil edilerek bir sınıflandırma gerçekleştirilmiştir. Modelin sonuna ise beş çıkışa sahip bir softmax katmanı eklenmiştir. Bu sayede modellerin ilk katmanlarında görüntülerin kenar, şekil vb. özelliklerine odaklanan, fakat, sonraki katmanlarda WBC'lerin daha karmaşık özelliklerine odaklanan mimariler elde edilmiştir. Transfer öğrenmenin yanı

sıra, LeNET [41] mimarisine benzer bir model oluşturularak, veri setinin sınıflandırılması gerçekleştirilmiştir. Böylece, CNN ve Kapsül Ağlarının sıfırdan yapılacak bir eğitimde yakalayacağı başarı oranlarını kıyaslama ortamı oluşmuştur.



Şekil 4.29. Baştan eğitilen CNN ve transfer öğrenme modellerinin eğitim ve doğruluk değerleri.

Kullanılacak modellerin ilk ağırlıklarının verilmesinde, 1000 farklı kategoriden görüntüler barındıran, böylece, farklı veri setlerinde yer alan görüntüleri de yorumlama yeteneği oldukça fazla olan ImageNET [100] veri seti tercih edilmiştir. Ayrıca, bu veri seti üzerinde yüksek performans elde eden modeller tercih edilerek maksimum seviyede çıkarım yapılabilmesi amaçlanmıştır. Dolayısıyla, test için kullandığımız modeller Inception-ResNETv2 [101], Inceptionv3 [64], ResNET50 [43] ve VGG19 [102] olarak belirlenmiştir. Bu modellerin tercih edilmesindeki bir diğer önemli neden ise CNN'in yanı sıra, ağ içinde ağ [87] ve Derin Artık Ağlar (DRN) yöntemlerinin sağladığı avantajları da kullanan yöntemler olmasıdır. Böylece diğer öğrenme

metotlarının da veri seti üzerindeki başarı oranları incelenebilmiştir. Tüm modeller 50 devir için eğitilmiş ve doğrulama aşamasında *k-fold çapraz doğrulama* yönteminden faydalanmıştır.

Şekil 4.29’da gözlemlenebileceği üzere, CNN ile baştan bir eğitim gerçekleştirildiğinde, eğitim verisinin düşük sayıda olmasından dolayı, beklendiği gibi, ağ 20. devirden itibaren eğitim verisini ezberlemeye başlamıştır. Diğer yandan, her bir devirde eğitim verileri karıştırıldığından, elde edilen doğruluk değerlerinin de oldukça düşük olduğu, CNN modelinin sınıflandırmayı neredeyse “yazı-tura” olasılığına indirgediği gözlemlenmiştir (%69,57). Modelin eğitime katılmayan görüntüler üzerindeki başarı oranı ise %62,50 olarak ölçülmüştür. Diğer yandan, daha önce eğitilen modeller üzerinde gerçekleştirilen testlerde, sınıflandırıcıdan önceki son iki katman da eğitime dahil edilerek bir sınıflandırma süreci sağlansa da verilerin ezberlenilmesinin önüne geçilememiştir. Fakat burada, baştan eğitilen CNN’den farklı olarak, modellerin WBC’lerin karmaşık özelliklerinin de öğrenimini gerçekleştirebildiği ve veri seti üzerinde başarılı bir kestirim gerçekleştirebildiğinden bahsedilebilir.

Dört model için ulaşılan en yüksek doğrulama değerleri sırasıyla; *Inception-ResNETv2*: %95,65, *Inceptionv3*: %91,30, *ResNET50*: %91,30, *VGG19*: %95,65 olarak ölçülmüştür. Fakat, test verisi üzerinde sınıflandırma gerçekleştirildiğinde ulaşılan başarı değerlerinin, doğrulama değerlerinin oldukça aşağısında kaldığı görülmüştür. Ölçülen *hassasiyet* değerleri sırasıyla *Inception-ResNETv2*: %82,50, *Inceptionv3*: %80,00, *ResNET50*: %80,00, *VGG19*: %77,50 olmuştur.

4.7. TARTIŞMA

Transfer öğrenme yöntemleri birçok sınıflandırma probleminin üzerinden başarı ile gelebilse de bu modellerin eğitim süreçleri makroskobik dünya görüntüleri ile gerçekleştirildiğinden, mikroskobik veriler üzerindeki başarılarının nispeten daha düşük olduğu görülmüştür. Diğer yandan, bu mimariler oldukça derin olmasından dolayı, çapraz doğrulama yöntemi ile eğitim ve doğrulama verileri sürekli yer değiştirilmesine ve modelin son katmanları eğitime dahil edilse dahi verilerin

ezberlenilmesinin önüne geçilememiştir. Doğrulama için kullanılan görüntüler de aslında eğitim verisinin içinde yer aldıklarından, yüksek doğrulama değerlerinin de kararlı bir ağın sonuçları olarak değerlendirilmesi yanlış olacaktır. Transfer öğrenme yönteminin bir diğer handikabı ise, bu modellerin sabit mimariler olmasıdır. Bu sebeple modelin giriş katmanında kullanılacak verilerin, daha önce eğitilmiş veri setinde kullanılan görüntülerle aynı boyutlarda olması gerekmektedir. Medikal görüntülerde gerçekleştirilecek boyut azaltma işlemi değerli piksellerin kaybına neden olduğu gibi, boyut artırma işleminin de görüntüye gürültüler ekleyerek doğru bir sınıflandırmanın önüne geçmesi son derece muhtemeldir.

Transfer öğrenmenin medikal görüntüler üzerinde sağlıklı çalışmasını olumsuz etkileyen bir diğer sebep ise, medikal görüntülerin her zaman renkli görüntüler olmamasıdır (ör; MR görüntüleri, ultrason vb.). Bu bakımdan, üç kanallı görüntüler üzerinde eğitimi gerçekleştirilen modellerin, direkt olarak siyah-beyaz görüntüler üzerinde kullanılması mümkün olmayacaktır. Bu da ön-işlem süreci, dolayısıyla zaman kaybı ve verinin orijinalliğinin bozulması anlamına gelmektedir.

Tezin bu aşamasına kadar elde edilen sonuçlar ve Kapsül Ağlarının diğer DL yöntemlerine nazaran artıları, gerçekleştirilen çalışma açısından değerlendirilirse;

- Kapsül Ağları sayesinde herhangi bir ön-işlem gerçekleştirilmeden, direkt olarak verilerin baştan sona eğitimi sağlanmıştır.
- Düşük sayıdaki bir veri setinde dahi kararlı sonuçlara ulaşmıştır.
- Model üzerinde kullanılan hiper-parametreler üzerinde esneklik sağlamıştır.
- Kullanılan veri seti üzerinde diğer yöntemlere göre daha yüksek başarılarla ulaşılmıştır.

Kapsül Ağları ile gerçekleştirilen analizlerde gözlemlenen en büyük olumsuzluk ise dinamik yönlendirme sürecinin oldukça fazla işlem barındırması ve bu sebepten eğitim sürecinin oldukça uzun sürmesidir. Dolayısıyla, büyük veri setleri üzerinde bu modelin ulaşacağı başarı oranlarının incelenmesi oldukça zor bir hale gelmektedir.

Önerilen modelin LISC veri seti üzerindeki başarısını sorgulamak iyi bir başlangıç olsa da modelin diğer WBC veri setlerindeki başarısının da sorgulanması gerekmektedir. Önerilen model sadece LISC veri setinin ihtiyaçları doğrultusunda modifiye edildiğinden, daha büyük veri setleri için kullanılacak bir model de önerilebilir. Diğer yandan Kapsül Ağlarının alan adaptasyonu ve transfer öğrenmedeki yetenekleri de sorgulanmaya muhtaçtır.

Gerçekleştirilen çalışmanın literatüre katkısı ise aşağıdaki gibidir:

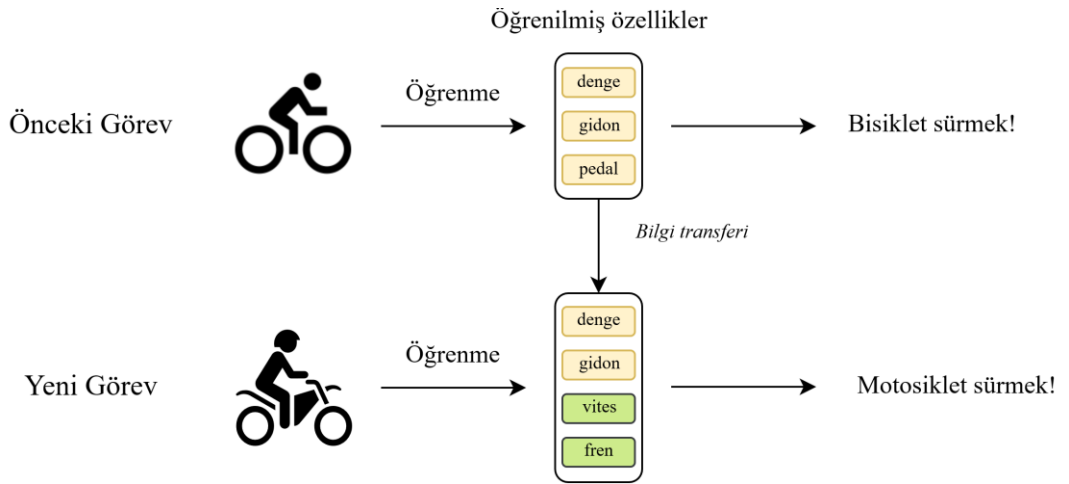
- Orijinal modele göre faydalı yapısal modifikasyonlar yapılmıştır.
- Literatürde medikal veri üzerinde gerçekleştirilen birkaç çalışmadan biridir.
- WBC kullanılarak gerçekleştirilen ilk çalışmadır.
- Kapsül Ağlarının üç-kanallı görüntüler üzerinde çalışmasını inceleyen ilk çalışmalardandır.
- WBC literatürüne nazaran oldukça düşük veri seti boyutlarında yüksek başarı elde edilmiştir.
- Diğer DL modellerine göre daha yüksek başarı oranlarına ulaşılmıştır.
- Kapsül Ağlarının çalışma prensibi detaylıca anlatılmıştır ¹.

¹ Tezin bu bölümüne kadar yapılan çalışmadan “Computerized Medical Imaging and Graphics” dergisinde “Classification of white blood cells using capsule networks” başlıklı bir makale çıkarılmıştır.

BÖLÜM 5

ALAN KAYMASI (DOMAIN SHIFT)

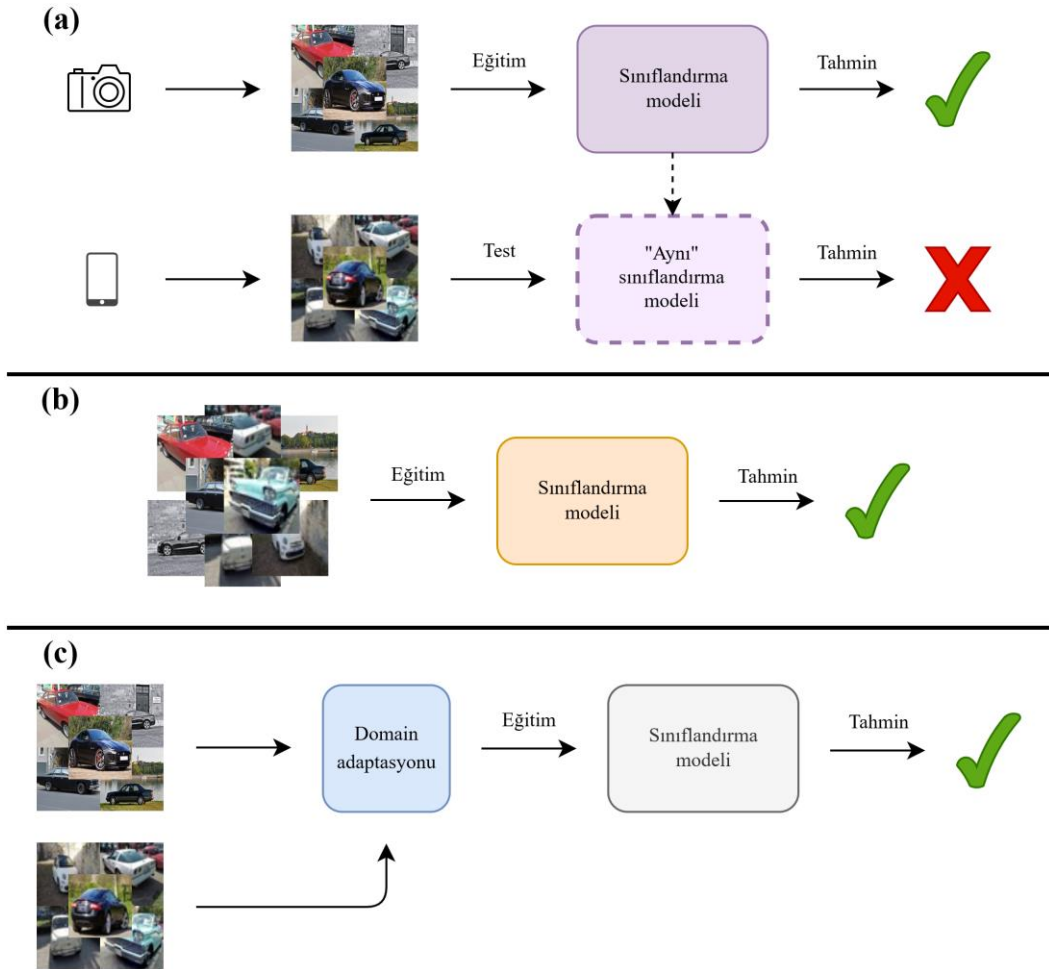
İnsanlarda öğrenme süreci genellikle geçmişteki görevlerden kazanılan deneyimlerin yeni bir problemde kullanılması ile gerçekleştirilmektedir [103]. Örneğin bisiklet kullanmayı bilen birisi, motosiklet kullanmayı daha önce hiç bisiklet kullanmamış birine nazaran daha kolay öğrenir (Şekil 5.1). Daha önce de bahsedildiği üzere ML yöntemleri sayesinde insanların sahip olduğu yeteneklerin bilgisayarlara aktarılabilmesi amaçlanır (örneğin; yapay sinir ağları) [21]. Bu açıdan transfer öğrenme yöntemleri de insanların bilgiyi öğrenme sürecinin modellenmesi olarak tanımlanabilir [104]. Bu süreç insanda otomatik olarak gerçekleşirken, bilgisayarla modelleme de *öğrenilmiş bilgilerin faydalı sunumlarının* yeni bir göreve transfer edilmesi ile sağlanmaktadır [105]. Tüm bilgilerin transferi yerine ise sadece *faydalı* olanları aktarmak olumsuz etkileri azaltacaktır. Örneğin, Şekil 5.1’de görüldüğü üzere *pedal* kullanmanın motosiklet kullanmaya faydası yoktur, dolayısıyla aktarılmamıştır.



Şekil 5.1. Geçmiş görevlerden öğrenilen bilgilerin yeni görevlere aktarılması.

Peki neden transfer öğrenmeye niçin ihtiyaç duyuyoruz? Bilgisayarlı görü problemleri açısından bu ihtiyacı bir gerçek dünya senaryosu ile açıklamaya çalışalım. Örneğin,

üniversitede öğrencisiniz ve bitirme projesi gerçekleştireceksiniz. Danışman hocanız da sizden *araba markası sınıflandırma* problemi üzerine bir model inşa etmenizi istedi. Yeni bir *kamera* satın aldınız ve sokaklarda oldukça fazla sayıda fotoğraf çektiniz. Daha sonra modelinizi oluşturarak eğitim gerçekleştirdiniz ve test verilerinde oldukça yüksek başarı oranlarına ulaştınız. Sonraki aşamada, danışman hocanız modelinizi değerlendirebilmek adına kendi *telefonunu* kullanarak okulun otoparkında fotoğraflar çektiğini ve modelinizi bu fotoğraflarla test etmenizi istediğini farz edelim. Fakat, size verdiği görüntüler eğitimde kullandığınız verilere kıyasla oldukça az sayıda. Modelinizi kullanarak hocanızın verdiği görüntülerde tahminler gerçekleştirdiniz, ne yazık ki, ulaştığınız skorlar hiçte tatmin edici değil (Şekil 5.2-a). Bu aşamada önünüzde iki adet seçenek belirecektir:



Şekil 5.2 Araba sınıflandırması senaryoları. a) eğitim gerçekleştirilen veri seti ile hedefte yer alan verilerin farklı dağılımlara sahip olması, b) kaynak ve hedef verilerin birleştirilmesi, c) kaynak veri setinden öğrenilen bilgiler kullanılarak hedef verinin öğrenilme sürecinin geliştirilmesi.

- Hocanızdan telefonunu aldınız ve yine sokaklarda oldukça fazla sayıda araba fotoğrafı çektiniz. Sonrasında ise, çekilen fotoğrafları etiketleyerek, kamera görüntülerinden oluşan veri setiyle birleştirdiniz. Birleştirerek oluşturduğunuz veri setinin sınıfları dengeliyse ve bu veri setini kullanarak modeli en baştan eğiterseniz, hocanızın size daha önce verdiği görüntüler üzerinde yüksek başarı oranlarına ulaşabilirsiniz (Şekil 5.2-b). Fakat bu süreç oldukça uzun ve maliyetlidir.
- Muhtemelen hocanız telefonunu vermeyeceğinden, yalnızca elinizdeki veri setini ve hocanızın verdiği az sayıdaki görüntüleri kullanarak, iki veri setinde de başarılı sonuçlara oluşacak bir model oluşturmaya çalışmak (Şekil 5.2-c). Bu süreç transfer öğrenme olarak tanımlanır ve kaynak veri setinden öğrenilen bilgiler kullanılarak hedef verinin öğrenilmesi sürecini geliştirmeyi amaçlar.

Makine öğrenmesi yöntemleri eğer eğitim-test verisi aynı kaynaktan geliyorsa ve *bağımsız ve aynı dağılıma (independent and identically distributed)* sahipse oldukça yüksek başarılarla ulaşabilmektedir [106]. Fakat eğitim gerçekleştirilen örnekler ile test gerçekleştirilecekler arasında yer alan farklılıklar, modelin başarısız sonuçlar üretmesine neden olmaktadır. Eğer, bilgisayarlı görü açısından bakarsak bu problemler genellikle verinin elde edilme sürecinde kullanılan farklı görüntüleme araçlarından kaynaklanmaktadır [11]. Görüntülerin sahip olduğu kendine has özelliklerin de (parlaklık, yakınlık-uzaklık, arka plan vd.) sınıflandırmaya olumsuz katkı yapması mümkündür [107,108]. Bu problem literatürde *domain değişimi (domain shift)* veya *veri seti değişimi (data set shift)* olarak tanımlanır [11,109]. Uzun yıllardır araştırmacıların üzerinde çalıştığı bu problemin üstesinden gelinebilmesi adına geçmişten günümüze oldukça fazla yöntem önerilmiştir [110]. Bu yöntemler literatürde *transfer öğrenme (TL)* ve *domain adaptasyonu (DA)* başlığı altında incelenir [111].

5.1. DOMAIN VE GÖREV

Bir domain, $D = \{X, P(X)\}$, bir veri setinde yer alan örneklerin sahip olduğu olasılık dağılımıdır [106,112] ve iki adet bileşenden oluşur: X : özellik uzayı ve $P(X)$:

marjinal dağılım, burada, $X=\{x_1, x_2, \dots, x_n\} \in X$ 'dir. Makalenin başında verilen “araba sınıflandırması” örneği üzerinden açıklamaya çalışırsak; x_i : her bir örnekten çıkarımı yapılmış spesifik bir özellik (marka logosu, lastik boyutu, vs.), n : X 'te bulunan özelliklerin sayısı, X : tüm *mümkün (possible)* özelliklerin yer aldığı uzay ve X : veri setinde yer alan herhangi bir örnektir [111,113].

Bir görev, $T = \{Y, \Phi\}$, ise veri seti üzerinde gerçekleştirilen öğrenme işlemidir (sınıflandırma, kümeleme, vs.). Görev de domain gibi iki adet bileşene sahiptir: Y : etiket uzayı ve Φ : karar fonksiyonu. Yine araba sınıflandırması örneğinden devam edersek, Y sınıf sayısı iken, Φ , X 'ten çıkarımı yapılmış özellikleri ve örneğe ait sınıf bilgisini kullanarak öğrenme gerçekleştirilmeyi ve eğitim sonunda minimum *kayıp değerine* sahip olmayı amaçlayan karar mekanizmasıdır [114].

Eğitim yapılan veri setinin özelliklerini öğrenen fonksiyonun yetenekleri test verisi üzerinde sınanacağına göre, iki farklı domainden bahsetmek mümkündür: kaynak domain $D_s = \{X_s, P(X_s)\}$ ve hedef domain $D_t = \{X_t, P(X_t)\}$. Aynı şekilde iki farklı görev: kaynak $T_s = \{Y_s, \Phi_s\}$ ve hedef $T_t = \{Y_t, \Phi_t\}$ mevcuttur [113]. Klasik ML 'de $D_s = D_t$ ve $T_s = T_t$ olarak farz edilir. Buradan hareketle, kaynak domain de yer alan veriler üzerinde öğrenme gerçekleştiren Φ_s fonksiyonunun ve hedef domainde yer alan örnekler üzerinde tahmin gerçekleştiren Φ_t fonksiyonuna eşit olduğu söylenebilir [115].

5.1.1. Domain ve Görev Eşitsizliği

Bir önceki bölümde verilen nütasyonlara göre, iki tane eşitsizlik durumunun ortaya çıkması muhtemeldir: Domain farkı $D_s \neq D_t$ ve/veya görev farkı $T_s \neq T_t$ [111].

Domain farkı ve görev farkı da iki eşitsizlikten kaynaklanabilir:

Eğer $D_s \neq D_t$ ise:

- $X_s \neq X_t$: Özellik uzayları eşit veya benzer değildir. Örneğin, kaynak domain sadece araba görüntülerinden oluşuyor iken hedef domainin kamyon veya motosiklet görüntülerini de barındırması gibi.
- $P(X_s) \neq P(X_t)$: Marjinal dağılımlar eşit veya benzer değildir. Örneğin, görüntülerin farklı görüntüleme cihazları tarafından elde edilmesi gibi.

Eğer $T_s \neq T_t$ ise:

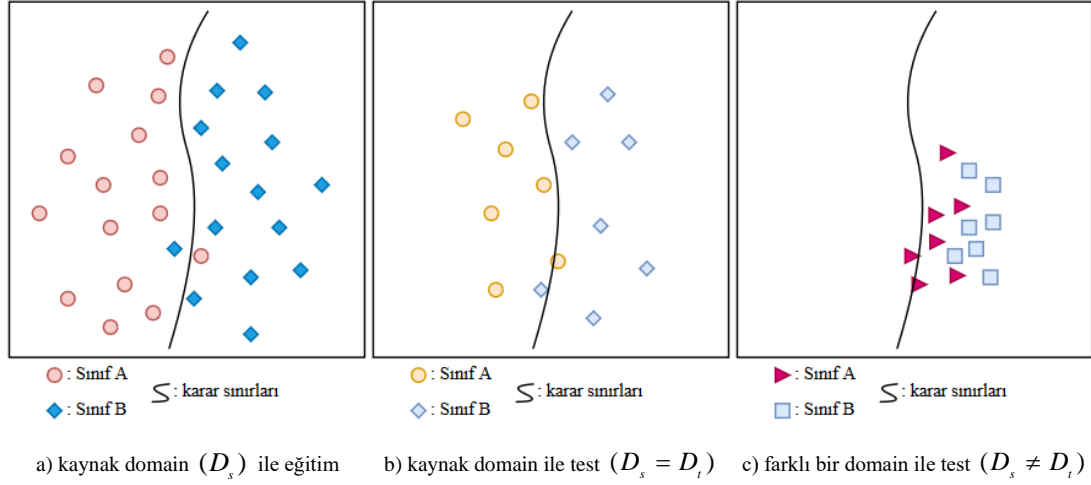
- $Y_s \neq Y_t$: Etiket uzayları eşit değildir. Kaynak domaindeki veri seti 4 sınıflı iken hedef domainde yer alan veri setinin 10 sınıflı olması gibi.
- $\Phi_s \neq \Phi_t$: Olasılık açısından bu durum koşul olasılık dağılımının farklılığı olarak da değerlendirilebilir, $P(Y_s | X_s) \neq P(Y_t | X_t)$. İki özelliğin farklı domainlerde farklı anlamlara gelmesi anlamına gelir. Örneğin, gerçek bir araba ile bu arabanın birebir aynı modellenmiş bir oyuncağı gibi [113].

Bu durumlardan bir tanesi dahi gerçekleştiğinde *veri seti değişimi (data set shift)* denilen problem ortaya çıkar. TL sayesinde, bu problemlerin üstesinden gelmek mümkün olabilmektedir [11].

5.2. TRANSFER ÖĞRENME

Kaynak veri seti ile hedef veri seti arasında domain uyumsuzluğuna neden olabilecek çeşitli etkenler mevcuttur (simple covariate shift, imbalanced data, domain shift, vs.) [11]. Bu gibi durumlarda kaynak domaindeki veriler kullanılarak bir modelleme gerçekleştirilebilse de, yani, eğitim aşamasında, öğrenici fonksiyon Φ_s , kaynak domain de yer alan örnekleri x_s ve bu örneklere ait sınıf bilgilerini içeren etiketleri y_s kullanarak kendini geliştirebilse de, test aşamasında farklı bir domainden gelen verilerle karşılaştığında, $\{x_t, y_t\}$ çiftleri üzerinde tahmin gerçekleştirecek Φ_t yetersiz kalacaktır (Şekil 5.3). İşte bu öğrenici ve tahmin edici fonksiyonlar arasında bir bağlantı kuran, tahmin edici fonksiyonu önceden öğrenilmiş bilgiler sayesinde

geliştiren ve öğrenici modele genelleştirme yeteneği kazandıran eğitim modeline TL adı verilmektedir [103]. ML en baştan bir modelleme gerektirirken, TL ise daha önce öğrenilmiş bilgiyi yeni problemi daha hızlı ve iyi çözmeye kullanılabilmektedir [111].



Şekil 5.3. Kaynak ve hedef domainler arasındaki dağılım farklılığına göre çizilen karar sınırları.

Pan & Yang'a göre TL inductive, transductive ve danışmansız olarak üç başlık altında incelenir. Yazarlar, bir problemin hangi başlık altında inceleneceğinin belirlenebilmesi için öncelikle kaynak ve hedef veriler arasında domain mi yoksa hedef farkı mı olduğuna, daha sonra ise veri setlerinin etiketli mi yoksa etiketsiz mi olduğuna bakılması gerektiğini belirtmişlerdir (Çizelge 5.1).

Çizelge 5.1. TL türleri [111].

TL Tipi	Benzer/İlgili Alan	Kaynak Domainde Etiketli Veri	Hedef Domainde Etiketli Veri	Görevler
Inductive	Multi-task Learning	Mevcut	Mevcut	Regresyon, Sınıflandırma
	Self-taught Learning	Yok	Mevcut	Regresyon, Sınıflandırma
Transductive	Domain Adaptasyonu, Sample Selection Bias, Co-variate Shift	Mevcut	Yok	Regresyon, Sınıflandırma
Danışmansız	-	Yok	Yok	Kümeleme, Boyut Azaltma

Problemin hangi sınıfta yer aldığından bağımsız olarak yukarıda belirtilen üç TL türünün de çözümünde kullanılacak yöntemleri *dört* sınıfta incelemek mümkündür:

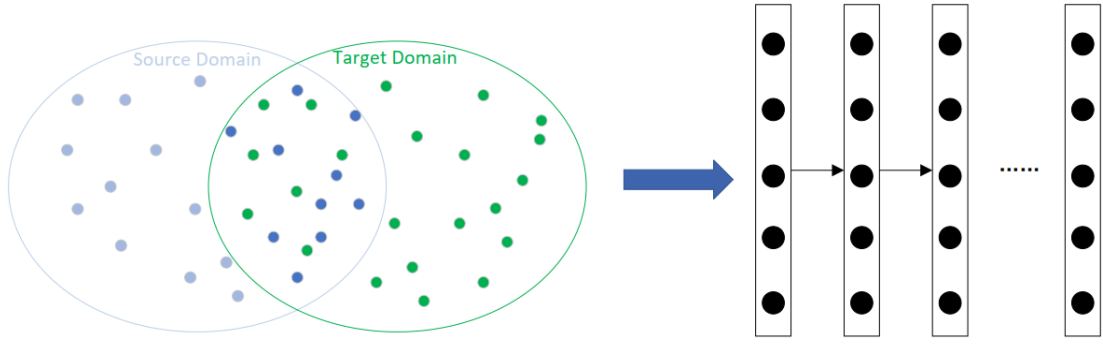
- Instance-based transfer (Örnek-tabanlı transfer): Kaynak domainde yer alan örneklerin bazı bölümleri *ağırlandırılarak* hedef domainde tekrardan kullanılabilir. Örnek ağırlıklandırma ve önem artırma bu yöntemde en sık kullanılan tekniklerdir.
- Feature-based transfer (Özellik-tabanlı transfer): Bu yöntemlerin arka planında yatan fikir hedef domain için *iyi bir sunum* oluşturmaktır. Bu yöntemde domainler arasında transfer edilecek bilgi *özellik sunumu* olarak oluşturulur. Bu sunumların da hedef domaindeki öğrenmeyi yüksek oranda artırması beklenir.
- Parameter-based transfer (Parametre-tabanlı transfer): Bu uygulamada kaynak ve hedef görevler için oluşturulmuş modeller arasında paylaşılan dağılımlar veya hiper-parametreler kullanılır. Transfer edilecek bilgi paylaşılan parametrelere kodlanır.
- Relational-knowledge transfer (İlişkili-bilgi transferi): Bu yöntem kaynak ve hedef domainde yer alan örnekler arasında bazı ilişkiler yer aldığını varsayar. Genellikle istatistiksel yöntemler içerir [111].

5.2.1. Derin Transfer Öğrenme

Her alanda olduğu gibi TL’de de DL yöntemleri sık kullanılmaya başlanmıştır. DL sayesinde verinin yüksek-seviye özelliklerinin çıkarımının sağlanması, bu tekniği ML’nin bir adım önüne taşımaktadır. Diğer yandan bu yöntem sayesinde, verinin özelliklerinin yarı-danışmanlı veya danışmansız yöntemler sayesinde otomatik olarak çıkarımı sağlanabilir [116].

TL alanında kullanılan DL tekniklerini *dört* kategori altında incelemek mümkündür:

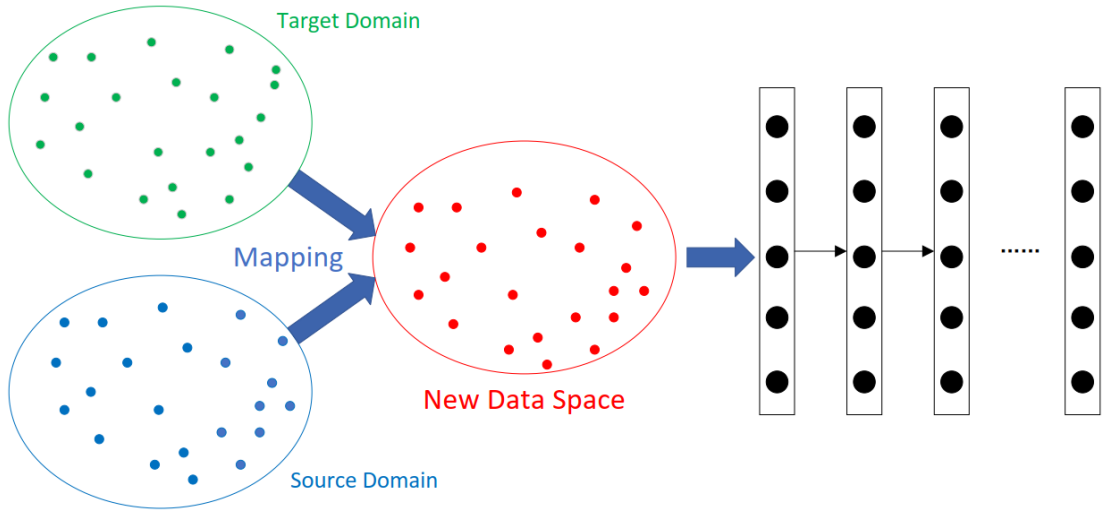
Instance-based (Örnek-tabanlı) Derin TL: Bu yöntemler kaynak domainde yer alan örneklerden bir kısmını alır, bunlara ağırlık ataması gerçekleştirir ve hedef domain de yeniden kullanır. Bu yöntemlerin yaptığı varsayım: “*İki domain arasında farklılıklar yer alsa da kaynak domaindeki belirli örnekler, hedef domain için uygun ağırlıklar ile kullanılabilir*” dir.



Şekil 5.4. Örnek-tabanlı derin TL [116].

Şekil 5.4’de açık mavi olanlar hedef domainle uzaktan yakından ilgisi olmayan örneklerdir ve hedef domaindeki eğitim veri setinde kullanılmamıştır. Koyu mavi olanlar ise hedef domain verilerine benzerdir ve uygun ağırlıklandırmalar yapılarak eğitim setine dahil edilmiştir.

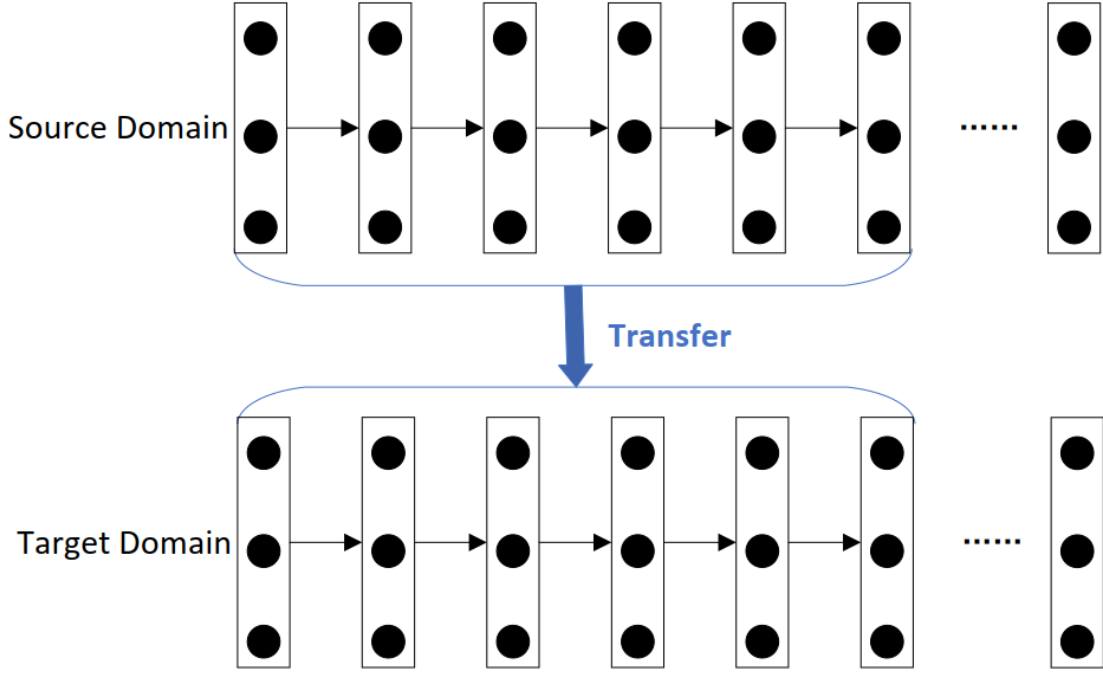
Mapping-based (Haritalama-tabanlı) Derin TL: Bu yöntemler “İki domain farklı olsa da yeni bir veri uzayında benzer özelliklere sahip olabilirler” varsayımı üzerine kurulmuştur.



Şekil 5.5. Haritalama-tabanlı derin TL [116].

Şekil 5.5’de görüldüğü üzere kaynak ve hedef domainde yer alan örnekler eş zamanlı olarak yeni bir veri uzayına taşınmıştır. Yeni veri setinde yer alan tüm örnekler sınıflandırıcı model için giriş verisi olarak kullanılabilir.

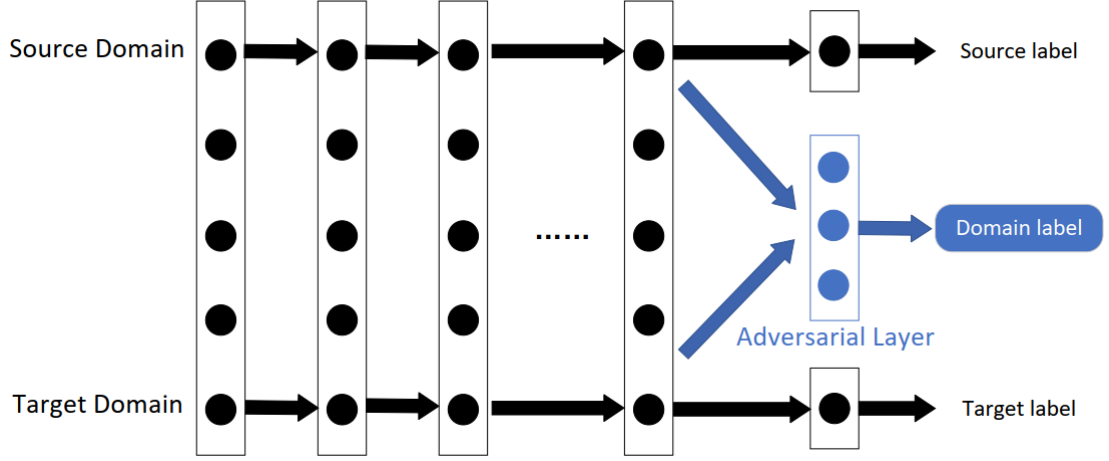
Network-based (Ağ-tabanlı) Derin TL: Bu yöntemde, kaynak domainde daha önce eğitilen bir ağın belli kısımları, ağ yapısı ve bağlantı parametreleri korunarak hedef domainde eğitilen ağda da kullanılır. Bu yöntemin varsayımı: “*Sinir ağları insan beyni gibi iteratif ve sürekli olarak özetleme yapar. O zaman ağın önündeki katmanlar özellik çıkarıcı olarak kullanılabilir*” şeklindedir.



Şekil 5.6. Ağ-tabanlı derin TL [116].

Şekil 5.6’da da görüldüğü üzere kaynak domainde yer alan ve oldukça fazla sayıda veri kullanılarak bir ağ eğitilmiştir. Daha sonra ise eğitilen ağın bir kısmı hedef domain için kullanılacak ağın bir kısmında kullanılmıştır. Transfer edilen bölümün *fine-tune* (*ince-ayar*) yapılması da mümkündür.

Adversarial-based (Çekişme-tabanlı) Derin TL: Bu yöntemin TL için gerçekleştirdiği varsayım ise: “*Faydalı bir transfer için, elde edilecek sunumlar ana öğrenme görevi için ayırt edici olurken, kaynak ve hedef domain için ayırt edilemez olmalıdır*” şeklindedir. Hem kaynak hem de hedef domainler için uygun olan transfer edilebilir sunumların elde edilebilmesi için özellikle Üretici Çekişmeli Ağlar (GAN) yönteminden faydalanılır.



Şekil 5.7. Çekişme-tabanlı derin TL [116].

Kaynak domainde yer alan geniş bir veri seti kullanılarak eğitim gerçekleştirilir (Şekil 5.7). Burada ağın ilk katmanları özellik çıkarıcı olarak kullanılır. İki domainden çıkarılan özellikler çekişme katmanlarına gönderilir. Çekişme katmanlar sayesinde ağ transfer edilebilir genel özellikleri öğrenmeye zorlanır.

5.2.2. Negatif Transfer

Herhangi bir transfer metodunun ne kadar etkili olduğu, kaynak ve hedef görevlerin birbiriyle olan ilişkisine göre şekillenir. Eğer ilişki kuvvetli ise ve transfer için kullanılacak yöntem bu ilişkiyi avantajına çevirebilirse, o zaman transferin performansı da artar. Fakat, eğer ilişki yoksa veya transfer yöntemi ile ilişki iyi bir şekilde ortaya çıkarılamamışsa o zaman hedef domain üzerindeki performansta azalacaktır. Negatif transfer durumunu engellemek adına genellikle üç yöntem kullanılır. Bunlar:

- Kötü bilgiyi reddetme: Hedef görevi olumsuz etkileyecek bilgilerin tespit edilerek transfer işlemine dahil edilmemesi.
- Kaynak görev seçimi: Eğer sadece bir kaynak domain veya görev yoksa, transfere en fazla katkıyı sağlayacak domain veya görev tercih edilebilir.
- Eş zamanlı görev modellemesi: Eğer birden fazla kaynak ve domain mevcutsa, bunların tümünü veya bir kısmını tercih ederek, transfere en fazla katkıyı sağlayacak model inşa edilebilir [103].

5.2.3. Domain Adaptasyonu

Pan ve Yang tarafından belirtilen karar mekanizmasına göre DA problemi *transductive TL* sınıfında yer alır [111]. Transductive TL’de görevler aynı iken ($T_s = T_t$), domainler arasında farklar mevcuttur ($D_s \neq D_t$). Bu bakımdan literatürde *domain adaptasyonu* olarak ta tanımlanır [104]. Eğer domainlerin özellik uzayları farklı ($X_s \neq X_t$) ise heterojen, aynı ise ($X_s = X_t$) homojen DA sınıfında yer alır [108]. Pan & Yang’a göre transductive TL ayarında, kaynaktaki veri *oldukça fazla ve etiketli* iken, hedef veriler ise *etiketsizdir ve eğitimde tamamı (veya bir kısmı)* kullanılır [111]. Fakat gerçek dünya problemlerinde az da olsa hedef domain de etiketli veri bulunabileceğinden, bu tanımlama biraz daha esnetilmiştir [117]. Böylece DA tıpkı makine öğrenmesi gibi veri tipine göre üç farklı sınıf altında incelenebilir hale gelmiştir:

- Danışmanlı DA: Kaynak etiketli veriden oluşurken, hedefte az sayıda etiketli veri ve etiketsiz veri mevcuttur. Etiketli veri eğitime dahil edilerek, etiketsiz veri üzerinde tahmin gerçekleştirilir.
- Yarı-danışmanlı DA: Kaynakta etiketli veri mevcutken, hedefte etiketli ve etiketsiz veri mevcuttur. Eğitime hem etiketli hem de etiketsiz veri dahil edilir.
- Danışmansız DA: Kaynakta etiketli veri mevcuttur, fakat hedefte ise etiketli veri yoktur. Etiketsiz verinin bir kısmı eğitime dahil edilerek, diğer etiketsiz veriler hakkında tahminlerde bulunulur.

DA çoğunlukla bir kaynak veriden bir hedef veriye uygulanmaktadır. *Çoklu-kaynak DA* da ise birden fazla kaynak domainden öğrenilen bilgiler ile hedefteki bir domain üzerinde adaptasyon sağlanır [114]. Tam aksine, eğer bir kaynak veriden çıkarılan özellikler hedefte birden fazla veriye adapte edilmek isteniyorsa, problem *çoklu-hedef DA* olarak adlandırılır [118]. Eğer eğitim sadece kaynak veri de gerçekleştiriliyor, fakat hedefteki veriden herhangi bir destek alınmadan, hedef veri üzerinde bir kestirim modeli oluşturmak isteniyorsa, o zaman *domain genelleştirme* konusu incelenebilir [112].

Genel olarak derin DA yöntemlerini üç kategori altında incelemek mümkündür [117]:

- **Discrepancy-based (Ayrım-tabanlı) yöntemler:** Bu yöntemler temel olarak oluşturulan modelin bir ince-ayar sürecinden geçirilmesini içerir. Bu süreç *dört* kritere göre gerçekleştirilir.
 - Sınıf kriterine göre: Domainler arasında bilgi transferi gerçekleştirilirken örneklere ait sınıf bilgileri kılavuz olarak kullanılır.
 - İstatistik kriterine göre: Kaynak ve hedef domainlerin istatistiksel dağılımlarını hizalamaya çalışır.
 - Mimari kriterine göre: Kullanılan derin ağın mimarisi ayarlanarak daha fazla transfer edilebilir özelliklerin öğrenmesi sağlanır.
 - Geometri kriterine göre: Kaynak ve hedef domainler arasında geometrik özelliklerine göre bir köprü oluşturur.
- **Adversarial-based (Çekişme-tabanlı) yöntemler:** Bu yöntemlerin hedefi ise kaynak ve hedef dağılımların arasındaki mesafelerin minimize edilmesidir. Bu amaçla, bir *domain ayırıcı* kullanılarak *domain karmaşıklığının*² artırılması sağlanır.
 - Generative (Üretici) modeller: Tipik GAN mimarisinde olduğu gibi ayırıcı modelin önünde bir üretici ağ bulunur.
 - Non-generative (Üretici olmayan) modeller: Yeni örnekler üretmek yerine, görüntülerin çıkarılmış özellikleri ve sınıf bilgileri kullanılarak örneklerin ayırt edici özellikleri öğrenilir. Böylece, *domain-değişmez (domain bağımsız)* sunumların ortaya çıkarılması sağlanır.
- **Reconstruction-based (Yeniden-inşa) yöntemler:** Bu yöntemler kaynak ve hedef domainlerde yer alan örneklerin tekrardan inşa edilmesinin DA performansına katkı sağlayacağını varsayar.
 - Encoder-decoder reconstruction (Kodlayıcı-çözücü ile yeniden inşa): AE modelleri kullanılarak örneklerin faydalı sunumlarını açığa çıkarmaya çalışır.
 - Adversarial reconstruction (Çekişmeli yeniden-inşa): Yeniden inşa sırasında üretilen hata oranları kullanılarak, orijinal ve yeniden inşa edilen görüntüler arasındaki farklar azaltılmaya çalışılır [117].

² Domain confusion (domain karmaşıklığı): Domainler arasındaki ayrımın minimum, modelin ana görevinin başarı oranının (ör: sınıflandırma) maksimum olma durumu [119].

BÖLÜM 6

BEYAZ KAN HÜCRELERİ VERİ SETLERİNDE GÖZLEMLENEN ALAN KAYMASI PROBLEMİNİN ÇÖZÜMÜNE YÖNELİK BİR ÇOKLU-HEDEF ALAN ADAPTASYONU MODELİNİN GELİŞTİRİLMESİ

Medikal görüntüler gerçek dünya görüntülerine nazaran daha karmaşık yapıdadır. Bu da medikal görüntüler için gerçekleştirilecek bir adaptasyon sürecini daha zor bir görev haline getirmektedir. Medikal görüntülerde karşılaşılan bir diğer problem de veri setlerinin genellikle farklı deneklerden elde edilmesi ile oluşturulmasıdır [12]. Bu sebepten deneklerin sahip olduğu farklılıkların da görüntülerde varyasyona neden olması beklenir [120]. Yani medikal görüntü analizinde sadece farklı cihazlardan gelen ölçümlerden değil denek farklılıklarından da etkilenmeyen bir model oluşturulmalıdır. Bu açıdan değerlendirildiğinde medikal görüntülerde DA standart DA problemlerine göre daha meşakkatli bir süreç olarak tanımlanabilir.

Fakat, klinik açıdan değerlendirildiğinde medikal veriler için domain farklılıklarından etkilenmeyecek bir model oluşturmak gerçek dünya problemlerine nazaran daha büyük bir öneme sahiptir. Bilindiği gibi, medikal verilerin elde edilmesi gerek etik gerekse maddi sebeplerden oldukça zor bir süreçtir [27]. Bu nedenle, örneğin, Magnetic Resonance Imaging (MRI) görüntülerinden beyin segmentasyonu gerçekleştiren bir araştırmacı, inşa ettiği modeli, farklı bir cihazdan gelen görüntüleri incelemek için kullandığında başarısızlıkla karşılaşacaktır [121,122]. Test aşamasında kullanılan görüntülerin elde edildiği cihazdan yeni bir veri seti oluşturmak ve bunları ön-işleme sürecinden geçirerek modele uygun hale getirmenin sebep olacağı maliyet ve zaman kaybı düşünüldüğünde, DA'nın Computer-aided Diagnosis (CAD) için ne derece mühim olduğu anlaşılabilir. Sahip olduğu yüksek önemden dolayı da, literatürde MRI, Computed Tomography (CT), Göğüs X-ray ve retinal görüntülerinde DA çalışmalarını gözlemlemek mümkündür [121–127].

DL yöntemleri ML yöntemlerine göre birçok problemde başarı oranını artırsa da eğitim için yüksek veri sayılarına ihtiyaç duyması belki de sahip olduğu en büyük handikapıdır [8]. Bu da daha önce de bahsedildiği üzere medikal veri analizi gerçekleştirmek isteyen araştırmacılar için önemli bir sorundur. Diğer bir problem ise, her ne kadar DL yöntemleri eğitim gerçekleştirilen verilerin özetlenmesinde ve faydalı özelliklerinin çıkarılmasında başarılı olsa da yine de *domain farklılıklarından* olumsuz etkilenmektedir [117]. Bu sebepten kullanılan yöntemden bağımsız olarak farklı kaynaklardan gelen verileri iyi bir şekilde değerlendirebilecek bir model ihtiyacı bulunmaktadır.

Doktora çalışmasının bu aşamasında DA problemi bir adım daha ileriye taşınmış ve çoklu-hedef DA problemi [128] WBC'lerin sınıflandırılması özelinde incelenmiştir. Motivasyonumuz, *“piyasada çok çeşitli özelliklerde mikroskoplar bulunduğundan, modeli her seferinde gelebilecek yeni veri setleri ile baştan eğitmek yerine, sadece bir veri setinden çıkarılacak ‘domain-değişmez’ özellikleri diğer veri setlerinin sınıflandırılmasında da kullanabileceğimiz bir model oluşturmak mümkün müdür?”* sorusuna cevap aramak üzerine kurulmuştur.

6.1. VERİ SETLERİ

Daha öncede belirtildiği üzere, WBC'lerin kandaki yoğunluklarının farklı olması dengeli bir veri seti oluşturmanın önündeki en büyük engeldir. Bu sebeple yalnızca bir denekten alınacak örnekler tüm WBC alt-tiplerinin eşit sayıda elde edilmesini sağlayamaz. Bu bakımdan WBC veri seti oluşturan araştırmacılar farklı deneklerden örnekler alarak periferik yayma testi gerçekleştirmekte ve nispeten dengeli bir veri seti oluşturmaya çalışmaktadır [51]. Fakat, her bir deneğin sahip olacağı biyolojik varyantlar da her bir deneğin farklı bir domain olarak değerlendirilmesine neden olur. Dolayısıyla, tüm denekler için genelleştirme yapabilecek bir model oluşturulması ihtiyacını doğurur [12].














































WBC'ler mikroskop altında incelenmeden önce lamel üzerindeki kan örneği bir boyama prosedüründen geçirilmektedir [2]. Bu aşamada kullanılan boyama tekniğindeki farklılıklar, oluşturulan solüsyonların fazla asidik veya alkalın olması,

kullanılan lamelin inceliği/kalınlığı gibi etkenler incelenen WBC'lerin ve kandaki diğer bileşenlerin (kırmızı kan hücreleri, trombositler, vs.) renk olarak farklı tepkiler vermesine yol açabilmektedirler [14,129]. Bir diğer farklılıkta kullanılan mikroskoplara takılan görüntüleme cihazından kaynaklanmaktadır. Cihazların sahip olduğu farklı lens yapıları ve görüntülerin elde edilmesinde tercih edilen yakınlaştırma/uzaklaştırma değerlerinin de marjinal olasılık dağılımlarında değişime yol açacağı bilinen bir gerçektir [107]. Bu farklılıkların her biri veya tamamı, kullanılan yöntemden bağımsız olarak (ML veya DL) domainler arasındaki geçişe olumsuz etki yaparak sınıflandırma başarılarını oldukça aşağıya çekecektir [130]. Her bir domain için yeni bir model oluşturmak ve verileri etiketleyerek en baştan bir eğitim sürecine sokmanın da oldukça maliyetli olacağı unutulmamalıdır.

Çalışmamızın motivasyonunu ve amacını oluştururken, WBC alt-tiplerinin sınıflandırılmasının önemi ve farklı domainlerden gelen görüntüleri de yüksek başarı ile sınıflandırabilecek bir model ihtiyacı göz önünde bulundurularak, bir *çoklu-hedef DA* modeli önerilmiştir. Modelin oluşturulması sırasında yapılan varsayımlar ve kullanılan tekniklerden bahsetmeden önce, kullanılan veri setleri detaylı analizleriyle beraber tanıtılacaktır. Bu sayede çözülmesi amaçlanan problemin niteliği hakkında okuyucunun bir fikir sahibi olması hedeflenmektedir.

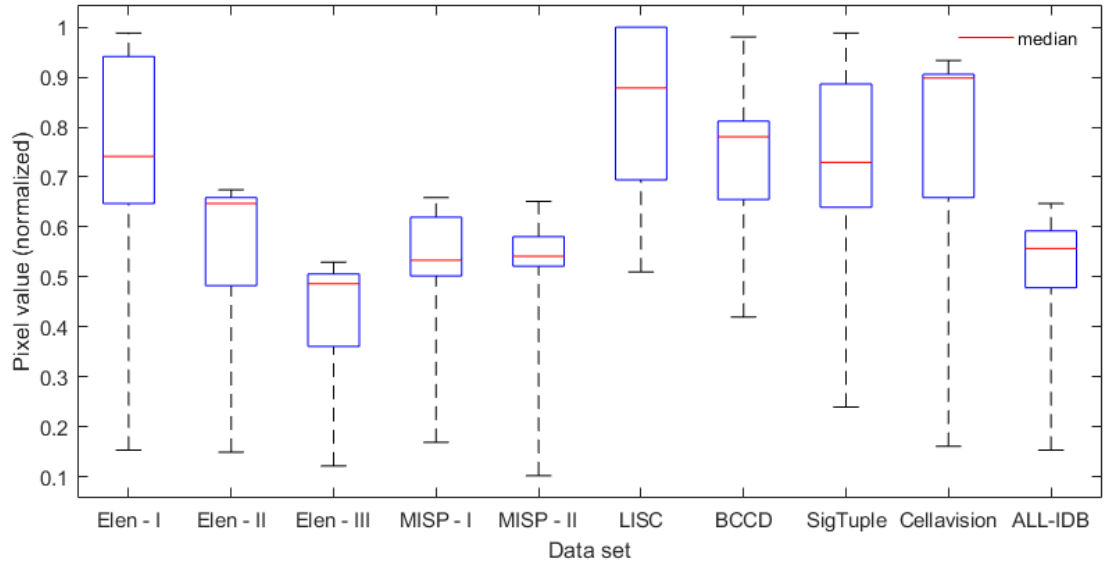
Çalışmamızda 10 farklı veri setinden faydalanmıştır. Bunlardan bir tanesi kaynak domain, diğerleri ise hedef domain olarak değerlendirilmiştir. Kaynak olarak kullanılacak veri setini belirlerken, yukarıda bahsettiğimiz dengesiz veri seti problemi göz önünde bulundurularak diğer veri setlerine nazaran sınıflarındaki veri sayısı birbirine yakın olan LISC [51] veri seti seçilmiştir. Tüm veri setlerinde yer alan görüntüler her bir örnek bir WBC alt-tipi içerecek şekilde 128x128 çözünürlüğünde kesitler alınarak kaydedilmiştir. Veri setlerinden örnek görüntüleri ve her bir veri setinin sahip olduğu örnek sayılarını ve veri setlerinin diğer özelliklerini içeren çizelge aşağıdaki gibidir (Çizelge 6.1). ALL-IDB [131] veri seti hali hazırda Akut Lenfoblastik Lösemi (ALL) örnekleri de barındırdığından çalışmamızda yalnızca sağlıklı hücrelerle çalışılmıştır. Diğer veri setlerinin ise orijinal etiketlerine sadık kalınmıştır. Eğer sınıflandırılmamış örnekler var ise bunlar da bir eksper yardımı ile doğru sınıflarına yerleştirilmiştir.

Çizelge 6.1. Kaynak ve hedef domainler.

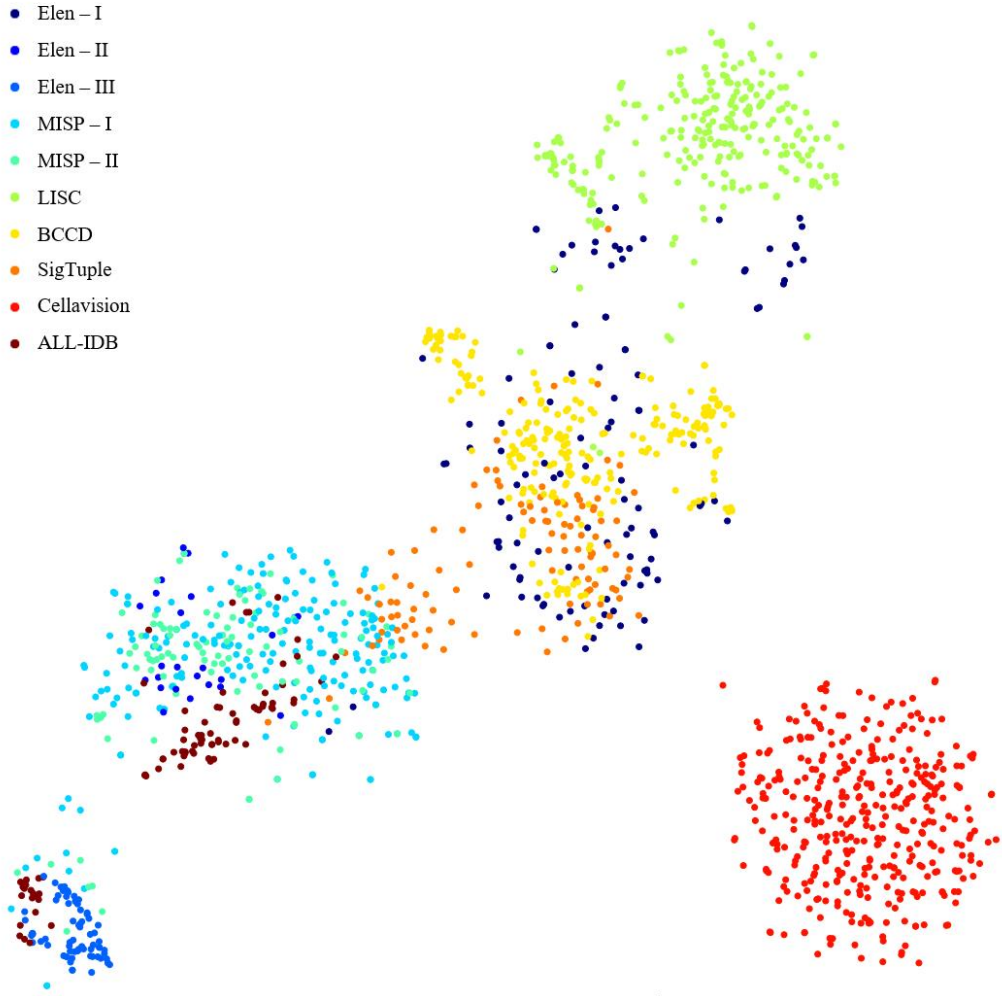
Domain Tipi	Veri seti	Boyama Tekniği	Görüntüleme Cihazı / Zum Seviyesi	Görüntü Formatı	Özellikler	Sınıflar					Toplam
						Bazofil	Eozinofil	Lenfosit	Monosit	Nötrofil	
Kaynak	LISC [51]	Gismo-Right	Sony SSC-DC50AP / 100X	720 x 576 BMP	Örnek Görüntü						266
					Örnek Sayısı	54	42	59	55	56	
	Elen – I [132]	Giemsa	Panasonic MN34120 / 100X	4608 x 3456 TIF	Örnek Görüntü						112
					Örnek Sayısı	1	1	38	4	68	
	Elen – II [132]	Giemsa	Panasonic MN34120 / 100X	2304 x 1728 TIF	Örnek Görüntü						23
					Örnek Sayısı	-	-	2	2	19	
	Elen – III [132]	Giemsa	Panasonic MN34120 / 100X	2304 x 1728 TIF	Örnek Görüntü						67
					Örnek Sayısı	-	-	14	-	53	
	MISP – I [133]	N/A	Nikon V1 / 100X	3246 x 2448 JPEG	Örnek Görüntü						202
					Örnek Sayısı	1	5	54	10	132	
Hedef	MISP – II [134]	N/A	Nikon V1 / 100X	3246 x 2448 JPEG	Örnek Görüntü						104
					Örnek Sayısı	4	3	27	12	58	
	BCCD [135]	Gismo-Right	CCD color camera / 100X	640 x 480 JPEG	Örnek Görüntü						241
					Örnek Sayısı	2	42	38	16	143	
	SigTuple [60]	Variable	iPhone-6s / 400X	Variable JPEG	Örnek Görüntü						107
					Örnek Sayısı	8	5	39	6	49	
	Cellavision [136]	Romanowsky	CellaVision DM96 / 1000X	360 x 363 JPEG	Örnek Görüntü						376
					Örnek Sayısı	2	5	94	35	240	
	ALL-IDB [131]	N/A	Canon PowerShot G5 / 300X – 500X	2592 x 1944 JPEG	Örnek Görüntü						84
					Örnek Sayısı	1	2	60	3	18	

Veri setleri incelendiğinde ilk göze çarpan farklılık, örnek sayılarında gözlemlenmiştir. Daha önce bahsedildiği üzere, WBC'lerin kandaki farklı yoğunlukları, oluşturulan veri setlerinin dengesiz olmasına yol açmıştır. Yine hepsinin farklı parlaklık seviyelerine sahip olduğu ve aynı sınıfa ait hücrelerin dahi boyama prosedürlerine farklı tepkiler verebildiği gözlemlenmiştir. Görüntülerin arka planlarında oluşan farklılıklar da kolaylıkla sezilebilmektedir. Veri setlerinin elde edilme süreçlerinde kullanılan görüntüleme cihazlarının farklı özelliklere sahip olması, diğer yandan farklı yakınlaştırma seviyelerinde görüntülerin alınması, farklı çözünürlüklerde ve farklı görüntü formatlarında kaydedilmesi veri setleri arasında domain farklılığına yol açan etkenlerden olmuştur. Bu etkenlerin piksel şiddetlerini nasıl etkilediğini gözlemlemek için Şekil 6.1 incelenebilir.

Şekilde de gözlemlendiği üzere veri setlerinde yer alan örnekler farklı piksel şiddeti aralıklarında yayılmaktadır. Dolayısıyla piksel değerleri üzerinden çıkarım yapan modeller, aynı sınıfa ait görüntünün sınıfını kestirim yapmakta zorlanması beklenir.



Şekil 6.1. Veri setlerinde yer alan lenfosit örneklerinin piksel şiddet dağılım grafiği.



Şekil 6.2. Veri setlerinin t-SNE sunumları.

Veri setlerinin histogram grafikleri görüntülerin sahip olduğu temel karakteristikleri (parlaklık, kontrast vs.) incelemede faydalı olsa da görüntülerin ve veri setlerinin birbirine olan mesafelerini çıkarma konusunda yetersizdir. Bu nedenle, bir sonraki aşamada veri setleri, bir boyut azaltma tekniği olan t-Distributed Stochastic Neighbor Embedding (t-SNE) kullanılarak incelenmiştir. Bu sayede veri setlerinin kendi aralarında ve diğer veri setleri ile olan mesafelerini iki boyutlu uzay düzleminde görmek mümkün olabilmektedir [137]. Şekil 6.2’de sınıf bilgileri göz ardı edilerek veri setleri incelendiğinde, her veri setinin kendine ait özel bir yayılıma sahip olduğu ve Cellavision veri seti hariç her veri setinin bir noktadan birbiriyle *ilişkili* olduğu görülmüştür. MISP – I ve MISP – II veri setleri birbirinden farklı amaçlarla oluşturulmasına rağmen aynı koşullarda elde edilmesinden dolayı beraber hareket etmişlerdir.

Eğer iki domain arasında sahip olduğu özellik uzayları açısından bir benzerlik var ise o zaman bu domainler “ilişkilidir” denilebilir [111]. DA sayesinde bu ilişkiler ortaya çıkarılarak, domainler arasında geçiş yapılması sağlanmaktadır. Bir sonraki bölümde bu geçişi sağlayacak yöntem tanıtılacaktır.

6.2. ÖNERİLEN MODEL

Pan & Yang iyi bir TL'nin üç kriterin doğru bir şekilde belirlenmesi sayesinde gerçekleştirebileceğini belirtmişlerdir. Bu kriterler TL sürecini sağlayacak prosedüre üç adet soru yönelir: “Ne” sorusu hangi özelliklerin kullanılacağı ile ilgilenirken, “ne zaman” ile transferin öğrenmeye katkı sağlayıp sağlamayacağı sorgulanır. “Nasıl” sorusunun cevabı ise problemi çözen yöntemdir [111].

WBC görüntülerinde domain farkına yol açan faktörlerin genel olarak görüntülerin elde edilmesindeki süreçten kaynaklandığını belirtmiştik. Fakat, görüntülerin parlaklık, kontrast, renk, yakınlık-uzaklık, arka plan gibi özellikleri değişkenlik gösterse de hücrelerin temel karakteristikleri korunmaktadır. Örneğin, bir bazofil hücresi çok granüllü, sitoplazması görülemeyecek kadar az, çapı $12-15\mu m$ seviyesinde ve mor-siyah yapıdadır [1]. Bu özellikler tüm bazofil hücrelerinde çok az farklılıkla da olsa mevcuttur. Dolayısıyla modelimizi inşa ederken, domainler arasındaki geçişi sağlayacak ortak özellikler, yani “ne transfer edilecek” sorusunun cevabı, WBC'lerin sahip olduğu spesifik özelliklerdir. Peki, değişen özellikler ile nasıl mücadele edeceğiz. Modelimiz, hücrelerin temel özelliklerini o kadar iyi öğrenmeli ki, görüntüler de bu ortak özellikler ile karşılaştığında gürültülerden etkilenmeyerek başarılı bir kestirim yapabilsin. Bu da modelin tüm görüntüler için genelleştirme yeteneğinin artırılması ile sağlanabilir. O zaman şu soruyu sorabiliriz:

Elimizde bir veri seti var ve tüm veri setlerine cevap verecek bir model inşa edilmek amaçlanıyor. Eğer, veri setini farklı veri dağılımlarına sahip olacak şekilde genişletebilirsek domain değişikliğinden etkilenmeyen özelliklerinin çıkarımını sağlayabilir miyiz?

Veri/özellik temelli yöntemler DA’da sıklıkla kullanılır ve kullanılan modelin genelleştirme yeteneğini arttırabilmeyi amaçlar. Bu yöntemlerin genel amacı hedef domain için “iyi bir sunum” öğrenmektir. Bunun için de domainler arasında ilişkili olan özellikler kullanılarak domainler arasındaki mesafe azaltılmaya çalışılır. Eğer özellik arttırımı yapılacak ise üretilecek örnekler zaten sahip olduğu orijinal özelliklerin yanı sıra domainler arasında geçiş yapmayı sağlayacak ortak özellikleri de barındırmalıdır. Bu sayede eğer model, *domain değişmez (domain invariant)* özellikleri başarılı bir şekilde öğrenebilirse, karşılaşıacağı yeni görüntülerde bu özellikleri *domain bağımlı (domain dependent)* özelliklerden ayırıştırarak doğru bir sınıflandırma sağlayabilecektir [111].

Fakat, iyi bir sunum bulabilmek için öncelikli adım oldukça fazla sayıda kaynak veriye sahip olmaktır. Bu sayede farklı örneklerin *altında yatan gizli bilgilerin (underlying latent information)* çıkarımı mümkün olur [104]. Elimizde bulunan kaynak veri setinin örnek sayısı bu tip bir çıkarım sağlayacak fazlalıkta ve farklı özellikleri de barındıracak varyasyona sahip olmadığından öncelikli amacımız veri sayısını yeterli seviyeye çıkarmak olmuştur.

6.2.1. Veri Arttırma (Data Augmentation)

Veri çoğaltma DL de sıklıkla kullanılan, oldukça basit ama efektif bir yöntemdir. Bu yöntem sayesinde dengesiz sınıf probleminin üstesinden gelinebilmesinin yanı sıra, sınıflandırıcı modele genelleştirme yeteneği kazandırılması da mümkün olabilmektedir [10,138]. Böylece eğitim verisinin ezberlenmesinin de önüne geçilebilmektedir. Diğer yandan veri çoğaltma sayesinde domain değişmez özelliklerin model tarafından öğrenilmesi mümkün olmakta, böylece, DA’da başarı oranları yüksek noktalara taşınabilmektedir [139–141].

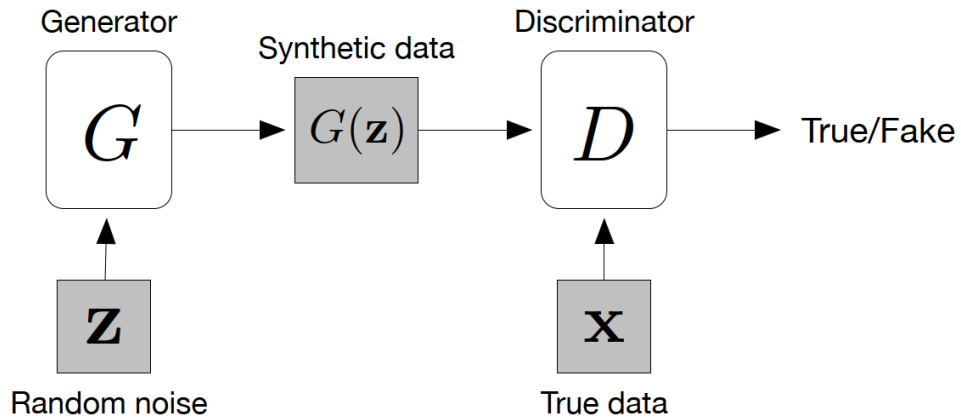
Veri çoğaltma oldukça faydalı olmasına rağmen kullanılacak verinin sahip olduğu özelliklere göre farklı stratejiler geliştirmek gerekmektedir. Örneğin, rakam tanıma gerçekleştirilecek ise, rakamların yatay düzlemde ters çevrilmesi sınıflandırmaya olumlu değil olumsuz katkı sağlayacaktır. Araştırmacılar da modellerini geliştirmek

ve *negatif transfer* fenomeninin oluşmasından kaçınmayı sağlayacak yöntemler geliştirmeye yönelmiştir [10,141,142].

Çubuk vd. tarafından önerilen AutoAugment [142], optimizasyon sürecinde pekiştirmeli öğrenme (reinforcement learning) kullanan, bir yandan veriyi çoğaltırken diğer yandan inşa edilen bir sınıflandırıcı modelin performansını da ölçen, böylece, doğru veri çoğaltma poliçelerini bulmaya çalışan bir yöntemdir. Çalışmamızda bu yöntemi kullanılarak hem kaynak domainde yer alan etiketli veri sayısının artırılması hem de sınıflandırıcıya genelleştirme yeteneği kazandıracak örneklerin elde edilmesi amaçlanmıştır.

6.2.2. Üretici Çekişmeli Ağlar (GAN)

Üretici Çekişmeli Ağlar (GAN) 2014 yılından itibaren DL’de çığır açan ve birçok problemin çözümünde başarılı bir şekilde kullanılan yöntemlerden biri olmuştur [143]. Özellikle veri sayısının yetersiz olduğu medikal görüntü analizi gibi problemlerde gerçekçi görüntüler üretilmesine imkân tanınmasıyla sınıflandırma ve segmentasyon gibi problemlerin çözülmesine yüksek katkılar sağlamışlardır [144,145]. Diğer yandan verilerin temel özelliklerini maksimum seviyede öğrenebilmesi, görüntüyü sınıflandıracak modeli kandırabilecek nitelikte görüntüler *inşa edebilmesi (construct)* ve *domain ayrışımını (divergence)* minimize edecek *ara (intermediate) formları* üretebilmesi sayesinde TL ve DA görevlerinde sıklıkla kullanılmaktadır [139,146].



Şekil 6.3. GAN mimarisi [147].

GAN mimarileri temelde iki kısımdan oluşmaktadır; *üretici* ve *ayırıcı*. Eğitim sırasında bu iki kısım arasında sürekli bir yarış hali mevcuttur. Ayırıcı eğitim verilerini öğrenerek, üretilen görüntülerin gerçek mi sahte mi olduğunu tanımlamak ile görevliyken, üretici ise ayırıcıdan gelen geri bildirimine göre daha gerçekçi görüntüler oluşturmaya çalışır. Eğitim ilerledikçe ayırıcı görüntüler ile alakalı daha fazla bilgiye sahip olacağından daha nitelikli ayrımlar yapabilecekken, üretici ise ayırıcıyı kandırabilmek için görüntülerin veri dağılımını en iyi şekilde yansıtacak görüntüler oluşturmaya zorlanacaktır. Standart bir GAN mimarisi Şekil 6.3'te görülebilir.

Şekilde görülen üretici, z gürültü vektöründen anlamlı veriler oluşturmaya çalışırken, ayırıcı üretilen görüntü $G(z)$ 'yi, gerçek görüntüler x 'ten öğrendiği bilgiler ışığında yargılayacaktır [147].

Hem üretici hem de ayırıcı birer sinir ağıdır. Dolayısıyla eğitimin her bir devrinde iki kısım için de maliyet değerleri hesaplanır ve eğitim başarısının güncellenmesi sağlanır (Eşitlik (6.1))

$$\begin{aligned} \min_G \max_D L(D, G) &= \mathbb{E}_{x \sim p_r(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \\ &= \mathbb{E}_{x \sim p_r(x)} [\log D(x)] + \mathbb{E}_{x \sim p_g(x)} [\log(1 - D(x))] \end{aligned} \quad (6.1)$$

Veri çoğaltma akabinde elde edilen görüntülerin ortak ve ayırt edici özelliklerine ulaşabilmek adına çalışmamızda GAN'ın geliştirilmiş versiyonlarından biri olan Wasserstein Generative Adversarial Networks – Gradient Penalty (WGAN-GP) yöntemi kullanılmıştır [148]. Wasserstein Generative Adversarial Networks (WGAN) [149] modelinin kayıp fonksiyonuna ceza uygulanarak elde edilen bu model sayesinde, GAN'ın ve WGAN'ın da sahip olduğu *mod çökmesi* (*mode collapse*) probleminin üstesinden gelmek mümkün olabilmektedir. Çalışmamızda WGAN-GP modelinin üretici ve ayırıcı kısmında Deep Convolutional Generative Adversarial Networks (DCGAN) [150] mimarisi kullanılmış, böylece, görüntülerden daha nitelikli çıkarımlar yapılması sağlanmıştır [149].

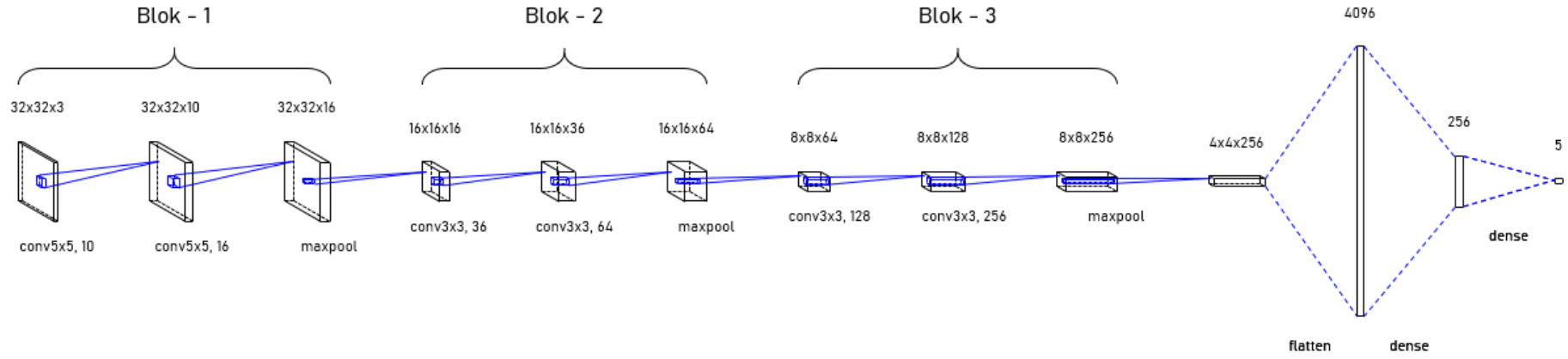
6.2.3. İnce-ayar (Fine-tuning)

Konvolüsyonel filtreler aracılığıyla verilerden özellik çıkarımı gerçekleştiren model (ör: CNN), öğrendiği özellikleri hiyerarşik olarak sunabilme yeteneğine sahiptir [9]. Bu sayede test aşamasında model tarafından daha önce görülmemeyen görüntülerde benzer bir sunumla karşılaşıldığında başarılı bir tanımlama yapılabilmektedir.

Bir konvolüsyonel ağ mimarisinin ilk katmanlarında verilerin genel özelliklerinin çıkarımı yapılırken, ileriki katmanlarda görüntünün daha spesifik özelliklerine odaklanılmaktadır [151]. Sahip olduğu bu yetenek sayesinde bir görevde öğrenilen bilgiler, diğer görevlere de başarılı bir şekilde aktarılabilir. Bu aktarım genellikle iki süreç ile gerçekleşir: *Şişe başı özelliklerin çıkarımı (bottleneck features) ve/veya ince-ayar süreci*.

Şişe başı özellik çıkarımında, daha önce başka bir veri seti ile eğitilen mimarinin özellik çıkarıcı katmanları ağırlık değerlerini korurken yeni görüntülerle ileri besleme gerçekleştirilir. Son konvolüsyon katmanının akabinde elde edilen değerlerle oluşturulan *vektör* bir sınıflandırıcıya (ör: MLP) bağlanır ve sadece sınıflandırıcı bölümünün eğitimi gerçekleştirilir. İnce ayar sürecinde ise ağın belli katmanlarının ağırlıkları korunurken, diğer katmanlar (veya tamamı) eğitime dahil edilerek modelin yeni görüntülere göre kendini güncellemesi sağlanır [98].

Çalışmamızda kullandığımız CNN modeli *iki konvolüsyon – bir havuzlamadan* oluşan ve birbirini takip eden üç ana bloktan oluşmaktadır (Şekil 6.4). Modelin hiperparametreleri *deneme/yanılma (babysitting)* [152] ile optimize edilirken, aynı zamanda model *PReLU* [95] aktivasyon fonksiyonu, *network-in-network* [87], *dropout* [153] ve *batch normalization* [101] teknikleri ile güçlendirilmiştir. İnce-ayar sürecinde ilk blok dondurulmuş, ikinci ve üçüncü blok eğitime dahil edilmiştir.

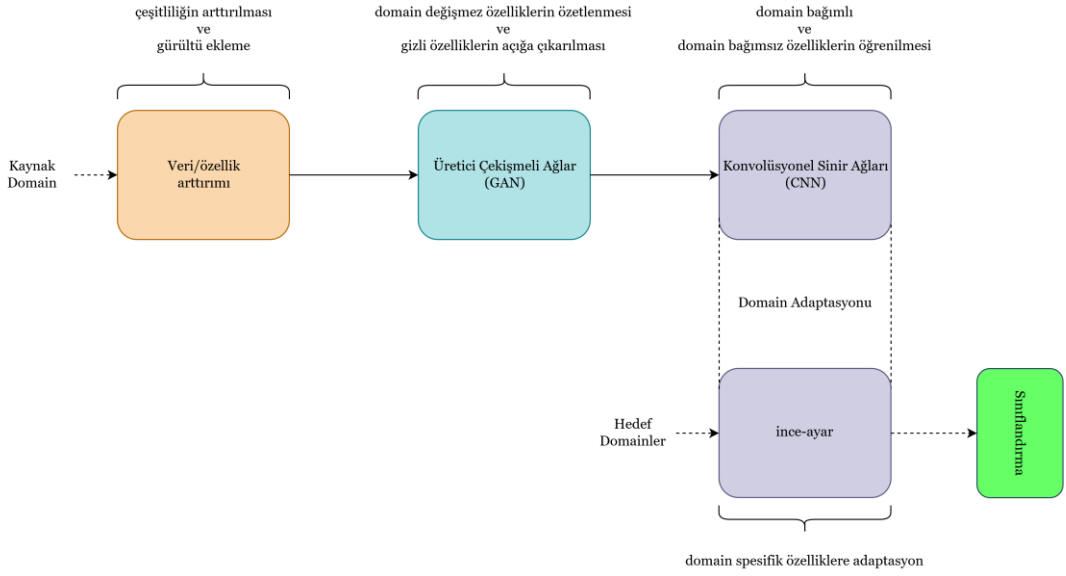


Şekil 6.4. Eğitim ve ince-ayar sürecinde kullanılan CNN modeli.

6.2.4. Özet

Önerdiğimiz model ve modelin oluşturulma sürecindeki varsayımlar özetlenirse;

- Çoklu hedef DA sürecinde kullanılacak veri setinde yer alan örnek sayısının yetersiz olması, ayrıca, diğer veri setlerinde gözlemlenen ve verinin elde edilme sürecinden kaynaklanan farklılıkları yansıtmadığı bilindiğinden, ilk aşamada veri arttırımı süreci kullanılarak, verinin farklı varyasyonlarda kopyaları üretilmiştir.
- Bu süreç sırasında AutoAugment yöntemi tercih edilmiştir. Sebebi ise, verilerin sayısı arttırılırken, diğer yandan transfer edilecek bilgilerin kaybını engellemektir. Diğer yandan, *veri seti bias* arttırılarak, eğitilecek modelin görüntülerde varyasyonlara neden olan faktörlere karşı kararlı olması hedeflenmiştir.
- Sonraki aşamada, farklı gürültülere sahip görüntülerden, sınıflar arasında ayrımı sağlayacak ve tüm domainlerde *ortak olan/paylaşılan gizli sunumların ortaya çıkarılması* hedeflenmiştir. Bu da farklı dağılımlara sahip görüntülerin bir *ara uzaya* taşınması sayesinde mümkün olabilir [154].
- Bu aşamada tercih edilen yöntem *GAN* olmuştur. Bu yöntemin sahip olduğu verilerin özelliklerini *özetleyebilme* ve verilerin *altında yatan faktörleri* ortaya çıkarabilme yeteneği sayesinde, farklı dağılımlara sahip verilerden *domain bağımsız* özelliklere sahip örnekler üretilirken, domain farkına yol açan gürültülerin de temizlenmesi hedeflenmiştir.
- Sürecin son aşamasında ise *GAN* ile elde edilen çıktılar ile bir CNN mimarisi eğitilmiştir. Eğitim sonrasında modelin ilk bloğu (*genel özelliklerinin çıkarımını sağlayan*) dondurulmuş, *spesifik özelliklerin* çıkarımını yapan bloklar ise eğitime dahil edilerek bir ince-ayar süreci gerçekleştirilmiştir.



Şekil 6.5. Önerilen modelin diyagramı.

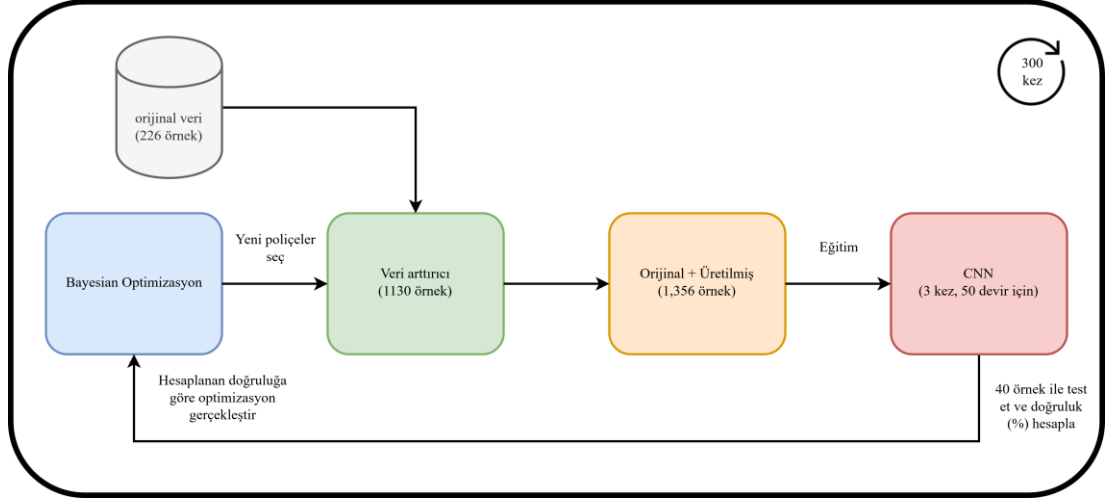
Şekil 6.5’te görüldüğü üzere kaynak domainde yer alan veriler öncelikle veri çoğaltma sürecinden geçirilerek veri sayısı yüksek noktalara taşınmış aynı zamanda verilere *domain spesifik* özellikler kazandırılmıştır. GAN sayesinde ise farklı domainlere yayılan görüntüler domain spesifik özelliklerden arındırılarak domainler arasında ortak olan özellikleri içeren görüntüler oluşturulmuştur (ara uzay). Son aşamada ise domain değişmez özellikleri öğrenen modelin, hedef domainde yer alan verilere ince-ayar ile adaptasyonu sağlanarak sınıflandırma gerçekleştirilmiştir.

6.3. DENEYSEL SONUÇLAR

Gerek AutoAugment gerekse GAN oldukça maliyetli işlem süreçlerine sahiptir. Dolayısıyla bu süreci hızlandırmak adına görüntüler öncelikle 32×32 seviyesine çekilmiştir. Diğer yandan pekiştirmeli öğrenme sürecinin de yavaş çalışması nedeniyle Bayesian optimizasyon kullanılmıştır [155]. Veri çoğaltma sağlayacak modelin çalışma süreci Şekil 6.6’da görülebilir.

İlk aşamada rastgele veri çoğaltıcılar seçilerek görüntü üretimi gerçekleştirilir ve orijinal verilerle birleştirilerek bir CNN eğitilmektedir. Veri çoğaltmanın sınıflandırma başarısına yaptığı katkı, ödül fonksiyonu olarak hesaplanarak optimizasyon sürecine geçilir. Kullanıcının belirlediği iterasyon sayısı kadar, yeni veri çoğaltma poliçeleri

seçilerek optimizasyon süreci devam eder. İşlemlerin sonunda sınıflandırma başarısını en fazla arttıran veri çoğaltma poliçeleri belirlenmiş olur.



Şekil 6.6. Veri çoğaltma optimizasyon süreci.

Kaynak domain de yer alan 266 görüntünün 226 tanesi eğitim, 40 tanesi ise test aşamasında kullanılacak şekilde optimizasyon süreci başlatılmıştır. CNN eğitiminde *10-fold çapraz doğrulama* süreci uygulanarak stabil sonuçlar elde edilmeye çalışılmıştır [97]. Öncelikle modelin *veri arttırıcı (augmenter)* modülü tarafından orijinal verilere toplam *beş* farklı veri çoğaltma çifti (örn: vertical flip & translate-y) uygulanarak, 5 adet veri seti elde edilmiştir ($226 \times 5 = 1,130$). Orijinal görüntülerin de eğitime dahil edilmesiyle $1,130 + 226 = 1,356$ görüntüyle CNN eğitilmiş, 40 görüntüde ise test gerçekleştirilmiştir. Bu şekilde üç kez ve 50 devir için eğitilen CNN mimarisinin ürettiği sonuçların ortalaması alınarak, seçilmiş poliçelerin sınıflandırmaya yaptığı katkı hesaplanmıştır.

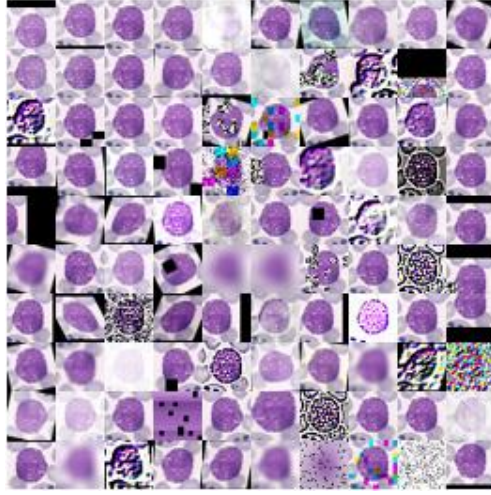
Tüm bu süreç 300 kez tekrar ettirilerek optimizasyon süreci tamamlanmıştır. Optimizasyon sonrasında sınıflandırmaya en fazla katkı sağlayan 20 poliçe belirlenmiş ve bu poliçeler kullanılarak toplam da $20 \times 5 \times 266 = 26,600$ görüntü üretilmiştir. Bu görüntülerin farklı sınıflara ait seçilmiş örnekleri Şekil 6.7’de görülebilir. Sınıflandırmaya en çok katkı sağlayan poliçeler ise Çizelge 6.2’de verilmiştir.

Çizelge 6.2. Veri arttırımı sırasında sınıflandırmaya en fazla katkı sağlayan 20 poliçe.

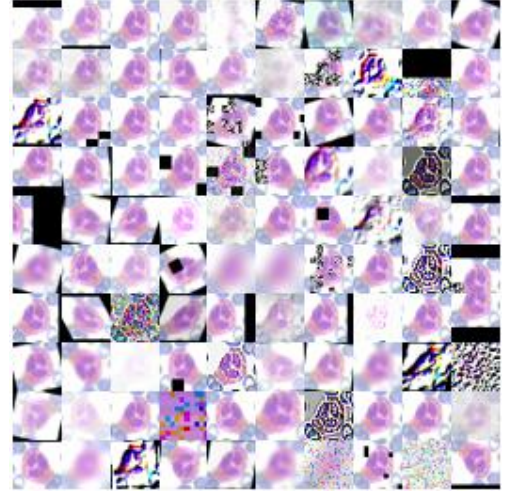
Deneme No	Veri seti - 1		Veri seti - 2		Veri seti - 3		Veri seti - 4		Veri seti - 5		Doğruluk
	1. Teknik	2. Teknik	1. Teknik	2. Teknik	1. Teknik	2. Teknik	1. Teknik	2. Teknik	1. Teknik	2. Teknik	
174	vertical-flip	translate-y	horizontal-flip	additive-gaussian-noise	brighten	rotate	emboss	clouds	translate-y	gaussian-blur	0.9640
212	clouds	emboss	horizontal-flip	coarse-dropout	horizontal-flip	vertical-flip	brighten	clouds	shear	horizontal-flip	0.9557
284	emboss	crop	horizontal-flip	coarse-salt-pepper	coarse-dropout	additive-gaussian-noise	coarse-dropout	crop	shear	vertical-flip	0.9527
276	gaussian-blur	translate-y	horizontal-flip	shear	additive-gaussian-noise	emboss	shear	vertical-flip	translate-y	rotate	0.9527
242	coarse-dropout	horizontal-flip	rotate	coarse-salt-pepper	gamma-contrast	brighten	vertical-flip	translate-y	emboss	translate-y	0.9500
80	rotate	brighten	dropout	translate-y	horizontal-flip	emboss	emboss	coarse-salt-pepper	fog	horizontal-flip	0.9500
240	translate-y	gamma-contrast	horizontal-flip	rotate	invert	add-to-hue-and-saturation	additive-gaussian-noise	clouds	rotate	rotate	0.9497
194	gamma-contrast	additive-gaussian-noise	crop	clouds	brighten	shear	rotate	additive-gaussian-noise	fog	brighten	0.9470
146	gamma-contrast	brighten	horizontal-flip	vertical-flip	gamma-contrast	translate-y	translate-y	gamma-contrast	gaussian-blur	coarse-dropout	0.9443
270	sharpen	rotate	translate-x	brighten	coarse-dropout	additive-gaussian-noise	clouds	horizontal-flip	gamma-contrast	additive-gaussian-noise	0.9417
285	gaussian-blur	dropout	gaussian-blur	vertical-flip	shear	brighten	shear	vertical-flip	shear	emboss	0.9390
154	gaussian-blur	brighten	horizontal-flip	translate-y	clouds	clouds	translate-x	crop	sharpen	sharpen	0.9387
127	crop	invert	horizontal-flip	vertical-flip	crop	coarse-dropout	rotate	translate-x	gamma-contrast	horizontal-flip	0.9363

Çizelge 6.2. (devam ediyor).

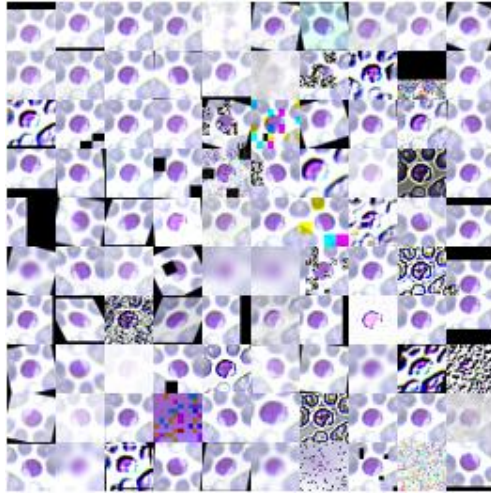
Deneme No	Veri seti - 1		Veri seti - 2		Veri seti - 3		Veri seti - 4		Veri seti - 5		Doğruluk
	1. Teknik	2. Teknik	1. Teknik	2. Teknik	1. Teknik	2. Teknik	1. Teknik	2. Teknik	1. Teknik	2. Teknik	
55	rotate	gamma-contrast	additive-gaussian-noise	emboss	emboss	shear	gaussian-blur	vertical-flip	rotate	gamma-contrast	0.9360
299	translate-y	sharpen	horizontal-flip	coarse-dropout	additive-gaussian-noise	gamma-contrast	vertical-flip	gaussian-blur	emboss	rotate	0.9360
157	invert	translate-y	horizontal-flip	vertical-flip	sharpen	additive-gaussian-noise	translate-x	vertical-flip	translate-y	translate-x	0.9337
252	brighten	shear	sharpen	additive-gaussian-noise	gamma-contrast	shear	translate-x	vertical-flip	coarse-salt-pepper	translate-y	0.9333
200	sharpen	sharpen	horizontal-flip	translate-y	coarse-salt-pepper	brighten	gaussian-blur	horizontal-flip	coarse-dropout	rotate	0.9333
14	gamma-contrast	gamma-contrast	horizontal-flip	additive-gaussian-noise	gaussian-blur	rotate	translate-y	horizontal-flip	clouds	vertical-flip	0.9333
144	rotate	gaussian-blur	translate-x	add-to-hue-and-saturation	gamma-contrast	horizontal-flip	coarse-dropout	sharpen	brighten	gamma-contrast	0.9330



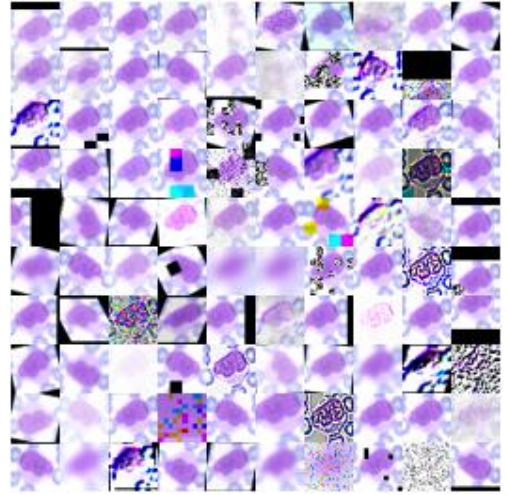
a) Bazofil



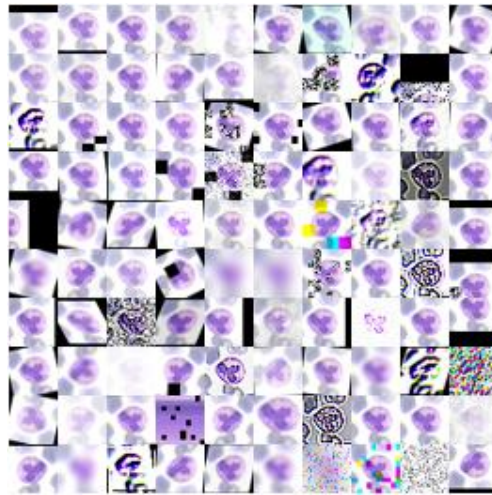
b) Eozinofil



c) Lenfosit



d) Monosit



e) Nötrofil

Şekil 6.7. Veri çoğaltma sonrası elde edilen görüntü örnekleri (her bir sınıfa ait).

Poliçeler incelendiğinde optimizasyon sürecinde bazı afin transformasyon (translate, flip, shear, etc.) işlemlerinin yanı sıra cropping ve coarse-dropout gibi, eğitilen modelin verileri ezberlemesinin önüne geçmeye çalışan veri çoğaltma tekniklerinin tercih edildiği gözlemlenmiştir. Fakat bu poliçelerden DA sürecine en fazla katkının, görüntülere gürültüler ekleyen (gaussian noise, blur, salt-pepper, etc), parlaklık veya kontrastı üzerinde değişiklikler gerçekleştiren (brighten, gamma-contrast) veya görüntülerin renk uzaylarında değişiklik yapan (hue-saturation) gibi teknikler sayesinde elde edileceği tahmin edilmektedir.

WBC veri setlerindeki domain değişiminin, özellikle, görüntülerin histogram karakteristikleri farklarından meydana geldiğini daha önce belirtmiştik. Dolayısıyla optimizasyon gerçekleştiren modelin sınıflandırıcının kan hücrelerinin ayırt edici özelliklerine odaklanabilmesini sağlamak amacıyla adına görüntülere gürültüler eklediği ve parlaklık-kontrast gibi özelliklerini değiştirmeye yöneldiği gözlemlenmiştir. Diğer bir deyişle, sınıflandırıcı modelin en yüksek başarıları *yerel değişikliklerden (domain spesifik özellikler)* etkilenmeden, hücrelerin *değişmeyen özelliklerini (domain invariant)* öğrenebilmesini sağlayan örnekler sayesinde elde edebildiği görülmüştür [112]. Bunu da görüntülerin bir kısmını eğitim verisini ezberlemeyi engelleyecek basit transformasyon yöntemleri ile oluşturmuş iken, diğer görüntülere gürültüler ekleyerek verilerin karmaşıklığını arttırarak, böylece sınıflandırıcıyı domain değişmez özellikleri öğrenmeye zorlayarak gerçekleştirmiştir.

Bu aşamada üretilen görüntüleri kombin ederek bir kaynaktan gelen veriymiş gibi değerlendirmek ve bunlar ile bir sınıflandırıcı eğitmek, daha sonra sınıflandırıcı model ile hedef domainler üzerinde test gerçekleştirerek iyi sonuçlar elde etmeyi beklemek fazla iyimser bir bakış açısı olacaktır. Çünkü bu haliyle sınıflandırıcının kaynak veriler arasındaki domain farklılıklarını göz ardı etmesi muhtemeldir [156]. Bunun yerine her bir kaynak domainin sahip olduğu spesifik özellikleri ortaya çıkararak bir çoklu kaynak DA gerçekleştirmek daha faydalı olacaktır [108]. Çünkü, üretilen görüntüler sayesinde yalnızca WBC'lerin temel özelliklerine odaklanılması sağlansa da hedef domain de yer alan örnekler ile üretilen görüntüler arasında temel özellikler açısından da farklılıkların bulunabileceği unutulmamalıdır. Aslında üretilen görüntüler

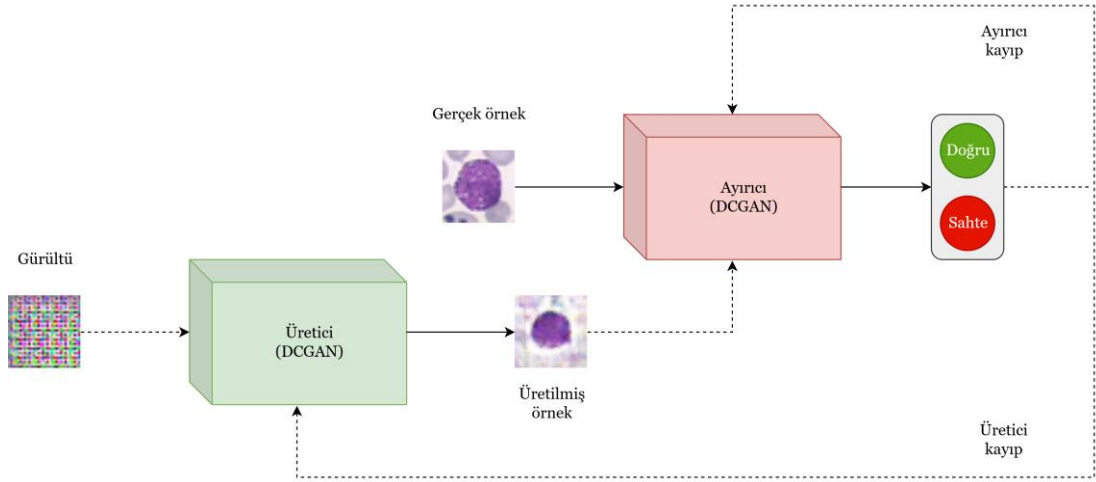
sayesinde problemin perspektifi genişletilmiş ve problem *bir kaynak- çoklu hedeften çoklu kaynak-çoklu hedef DA* haline döndürülmüştür.

Çoklu kaynak DA en temel haliyle, birden fazla kaynak domainden gelen verilerin altında yatan gizli faktörlerin kullanılarak hedef domainde iyi çalışan bir model ortaya koymak olarak tanımlanır [157]. Bu da ancak oldukça fazla sayıda örnek ile sağlanabilir. Bu sayede farklı dağılımlardan gelen örneklerde dağılım farkına yol açan faktörler çıkarılabilir ve hedef domain üzerinde tahmin performansını yükseltecek bir model ortaya koymak mümkün olabilmektedir [154].

Bu aşamadaki amacımız tam olarak farklı domainlerden gelen örnekler arasında *ortak olan ve paylaşılan özellikleri* bulmak, kaynak ve hedef veriler birbiriyle *bağıntılı* olduğundan, bu özellikleri iki domain arasındaki mesafeyi yakınlaştıracak bir köprü olarak kullanmaktır. Bu nedenle, domainler arasındaki farkları göz ardı edecek, ama, transfer edilebilir özellikleri de barındıran bir ara domain oluşturulmalıdır. Bu ara domain kaynak domainler de yer alan *yüksek seviye sunumları* yansıtarak *domain ayrımını minimize* ederken, *domain karmaşıklığının* artırılmasına da katkı sağlamalıdır. Bu sayede sınıflandırıcı modelin *verinin hangi kaynaktan geldiğini anlayamaması* amaçlanır [158].

Bu ara uzayı oluşturacak model olarak GAN kullanılabilir. Bu yöntem sayesinde farklı domainlerde yer alan bilgiler aynı özellik uzayına taşınabilmektedir [154]. Ortak ve domain spesifik özellikleri öğrenmede de oldukça yeteneklidir [108]. Ayrıca, DA işlemini direkt görüntünün üzerinde gerçekleştirebilme yeteneğine sahiptirler [105]. Bu sayede hem kaynak hem de hedef domainin birbirinden ayıramayacak şekilde benzer olmasını sağlayabilirler [116].

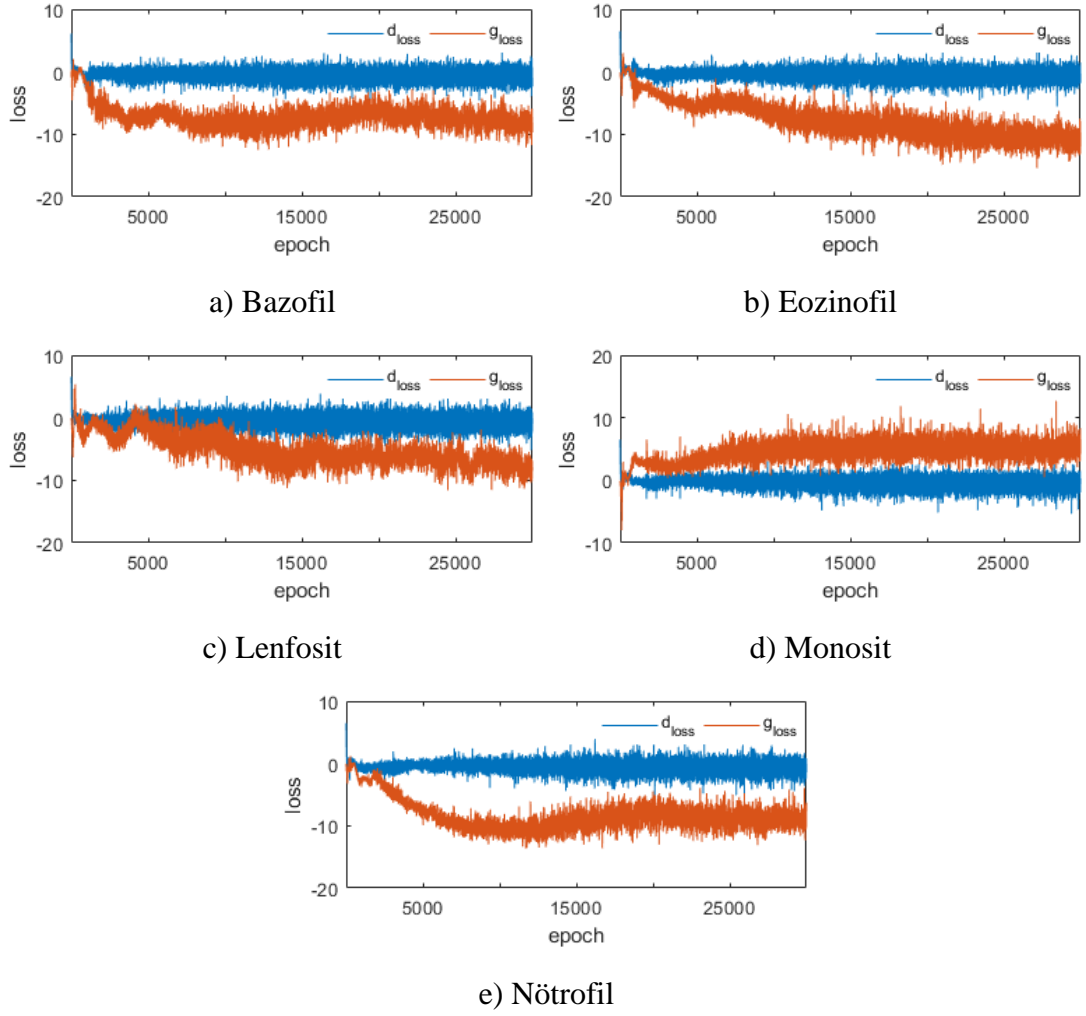
GAN modelleri genellikle iki ana bölümden meydana gelmektedir, üretici ve ayırıcı (Şekil 6.8). Üretici bölümü rastgele bir gürültüden anlamlı bir imaj oluşturmaya çalışırken, ayırıcı bölümü ise oluşturulan görüntülerin orijinal görüntülere benzerliğini kontrol eden bir mekanizmasıdır. Bu süreç teorikte ayırıcının üretilen görüntüleri gerçek olarak değerlendirene kadar devam eder.



Şekil 6.8. WGAN-GP modeli.

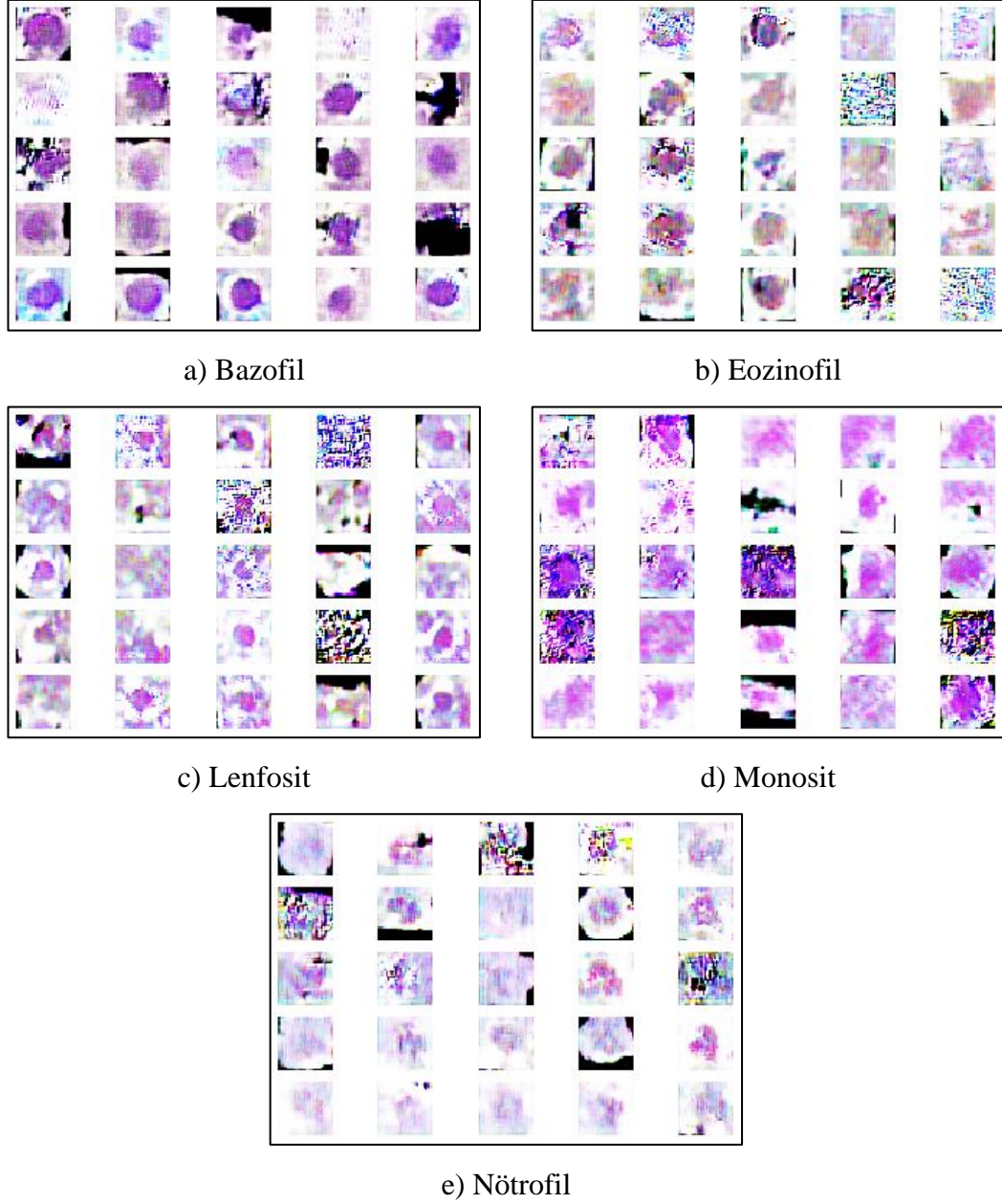
WGAN çoğu GAN modelinin aksine iki dağılım arasındaki benzerliği hesaplarken *Jensen–Shannon (JS)* veya *Kullback-Leibler (KL) divergence* yerine *Wassertein* (veya *Earth Mover*) mesafesini kullanarak, bu iki mesafe ölçütünün neden olduğu *türevlerin patlaması (vanishing gradient)* ve *mod çökmesi (mode collapse)* probleminin üstesinden gelmeye çalışmaktadır. Bu sayede, GAN eğitimi sürecinin hızlıca yakınsamasını sağlayarak, optimizasyon sürecini *daha stabil* hale getirmektedir. Fakat bu modelin amaç fonksiyonunda kullanılan *ağırlık kırpmasının (weight clipping)* gene de yavaş yakınsamaya neden olması ve patlayan türevler probleminde yol açması, ayrıca modelin zayıf örnekler üretmesinden dolayı, Gulrajani vd. [148] tarafından ağırlık kırpması yerine uygulanan *türev cezası (gradient penalty)* ile geliştirilmiştir. Bu sayede orijinal WGAN'a göre *daha kararlı* bir model elde edilebilmiştir.

Veri çoğaltma sonunda üretilmiş 5400 *Bazofil*, 4200 *Eozinofil*, 5900 *Lenfosit*, 5500 *Monosit* ve 5600 *Nötrofil* görüntüsü sınıf bazında ayrıştırılarak, *danışmansız* bir eğitim sürecine sokulmuş (*model, görüntülerden daha fazla anlamlı çıkarımı her bir sınıf için ayrı ayrı ve kendi başına öğrenebilsin diye*) ve her sınıf için 30000 devir kadar eğitilmiştir. Eğitim sonunda elde edilen üretici ve ayırıcı için kayıp değerlerini içeren grafikler Şekil 6.9'da görülebilir.



Şekil 6.9. Üretici modelin üretici ve ayırıcı bölümleri için her bir devirde hesaplanan kayıp değerleri.

WGAN modelinin sağladığı avantajlardan bir tanesi de *ayırıcı kayıp fonksiyonunun*, üretilen görüntülerin *kalitesinin* ve orijinal verilere olan *yakınlığının* analiz edilmesine imkan tanınmasıdır [149]. Grafiklerde de gözlemlenebileceği üzere, ayırıcının kayıp fonksiyonu '0' bölgesine oldukça yaklaşmıştır, yani üretici tarafından oluşturulan görüntüler ayırıcı tarafından orijinal görüntülere oldukça benzer olarak değerlendirilmiştir. Üretilen görüntülerin farklı sınıflara ait örnekleri Şekil 6.10'da verilmiştir.

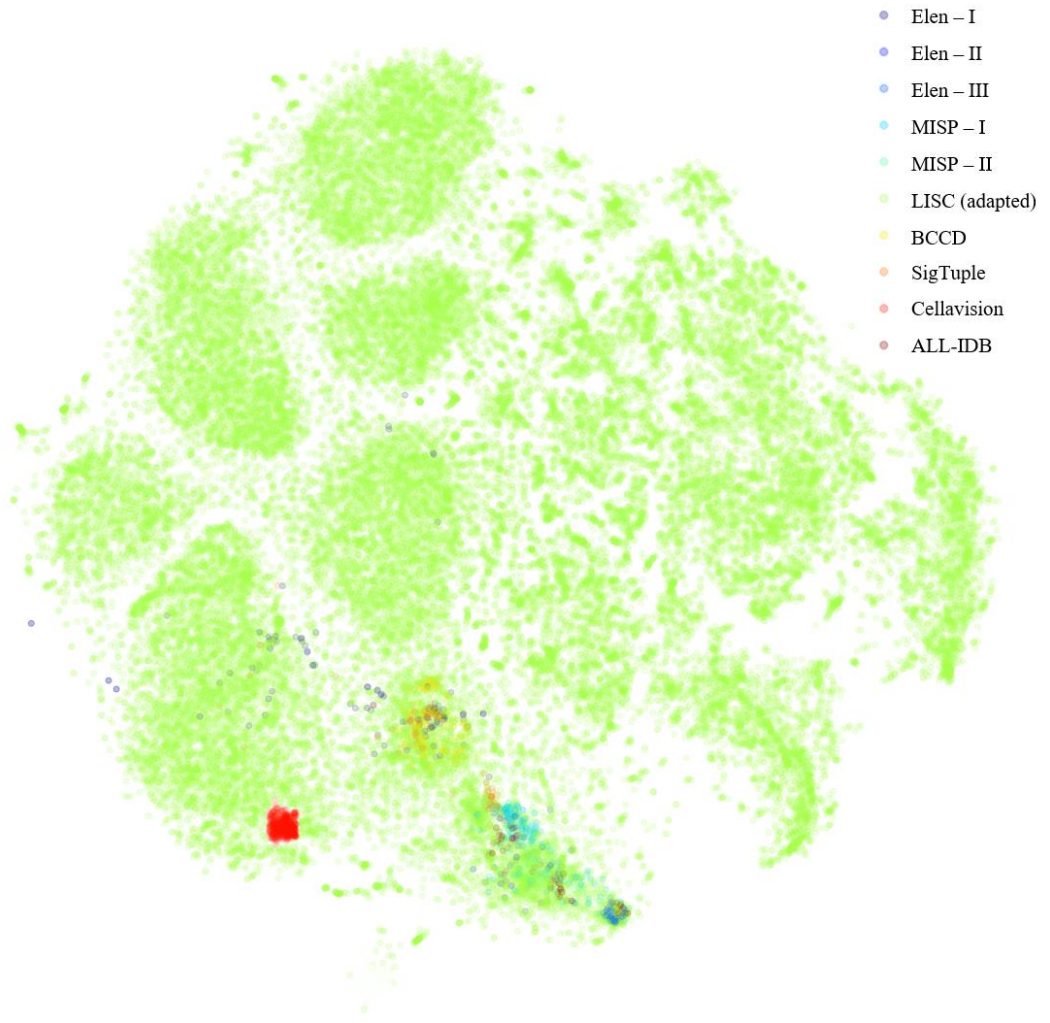


Şekil 6.10. Üretici model tarafından üretilen görüntüler.

Üretilen görüntüler incelendiğinde, örneklerin sınıf özelliklerini yansıtmada başarılı oldukları söylenebilir. Ayrıca, kaynak olarak kullanılacak verilerin sınıf ayrımı da artırılarak gizli bilgilerin açığa çıkarılması ve farklı varyasyonlara karşı stabil olması da sağlanmıştır [107]. Arka plandaki özelliklerin önemini yitirdiği ve modelin hücrenin kendisine odaklandığı gözlemlenmiştir. Görüntülerde yer alan siyah kısımlar veri çoğaltma optimizasyonu sürecinde uygulanan cropout tan kaynaklanmaktadır [159]. Bu teknik eğitilen model için iyi bir *düzenleyici (regularizer)* olarak görev yaptığından bu örnekler korunmuştur. Diğer yandan, bazı görüntülerin görsel olarak

başarılı olmadığı ve gürültülü bir izlenim verdiği görülmektedir. Fakat, Ilyas vd. bu gürültü veya bug olarak beliren şeylerin aslında üretici model tarafından öğrenilen özellikler olduğu ispatlanmıştır [160]. Dolayısıyla bu görüntülerin de kaynak domain olarak kullanılmasına karar verilmiştir.

Her bir sınıf için üretimi gerçekleştirilen 10000 görüntüden toplam 50000 görüntü ile CNN modelinin eğitimine başlanmadan önce üretilen görüntülerin, hedef domain de yer alan görüntülerin özelliklerini yansıtip yansıtmadığını görsel olarak incelemek adına bu görüntüleri de barındıran bir t-SNE sunumu yapılmıştır. Bu sayede önerilen modelin kaynak domaindeki veriler ile hedef domain de yer alan görüntüler arasındaki benzerliği yakalayıp yakalamadığı sorgulanacaktır (Şekil 6.11).



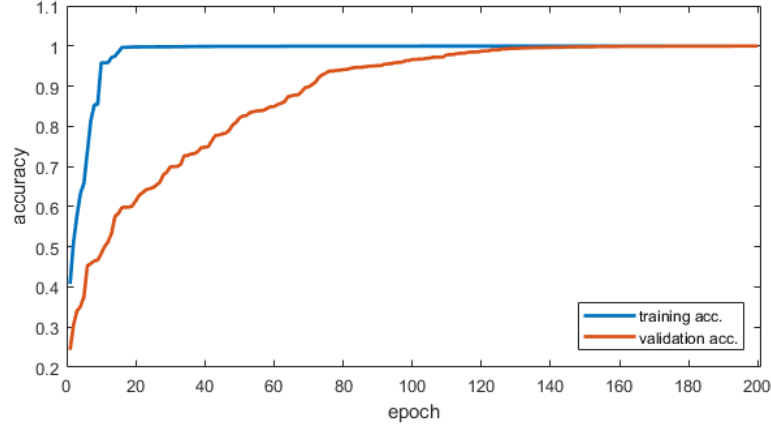
Şekil 6.11. Kaynak ve hedef domainleri içeren t-SNE sunumu.

Şekilde de görülebileceği üzere yapılan varsayımlar doğrultusunda kaynak domain de yer alan örneklerin hedef domainlerde yer alan örnekleri kapsayacak nitelikte oluşturulabildiği görülmüştür. Üretilen görüntülerin yoğunluğu nedeniyle hedef görüntüler zar zor seçilebilmektedir. Bu aşamadan sonra, kaynak domainde yer alan verilerin sahip olduğu ve hedef domaindeki verilerde de yer alan ortak özelliklerin sınıflandırıcı bir model tarafından öğrenilerek, hedefte yer alan tüm örneklere cevap verecek bir sunum oluşturmak mümkün olabilecektir [107]. Fakat, hedefteki örneklerin sahip olduğu domain spesifik özelliklere sınıflandırıcı model hâkim olmadığından, bu özellikleri sınıflandırıcı modele ince-ayar yöntemi ile tanıtarak sınıflandırıcının bu özelliklere de adaptasyonu sağlanmalıdır.

Bir DL yöntemi olan CNN özellik çıkarıcı olarak oldukça güçlü bir metottur [151]. Eğitim başında sahip olduğu rastgele ağırlık değerlerini, konvolüsyonel filtreler aracılığı ile görüntüyü tarayarak ve görüntünün sınıfını doğru tanımlayacak bu değerleri bulmaya zorlayarak güncellerler. Böylece, görüntülerin anlamlı özelliklerini, yani giriş ve sınıf arasındaki ilişkiyi kuracak bir *haritalamayı (mapping)* öğrenerek, kararlı sunumlar ortaya koyabilirler [108]. Farklı katmanlarda öğrendikleri bilgileri hiyerarşik olarak sunabildiklerinden, belli bölümlerin transfer edilebilmesi de mümkündür.

Çalışmamızın bu aşamasında kullanılan CNN modeli (Şekil 6.4) 200 devir için eğitilmiştir. 50000 görüntü, %80 eğitim, %10 doğrulama ve %10 test olarak üç parçaya bölünmüştür. Modelin her bir devirde 5000 görüntü ile doğrulama yapması sağlanarak, modelin eğitim verisini ezberlemesi engellenmeye çalışılmış ve daha kararlı sonuçlar elde edilmesi amaçlanmıştır. Eğitim sonunda elde edilen eğitim ve doğrulama değerlerini içeren grafik Şekil 6.12’de görülebilir.

Eğitim sonunda elde edilen en yüksek doğrulama değeri 0.9998 olarak ölçülmüştür. Test verisi üzerindeki tahmin skoru ise 0.9972 olmuştur. Bu da 5000 görüntünün yalnızca 13 tanesinin yanlış sınıflandırıldığı anlamına gelmektedir. Dolayısıyla CNN modelinin üretilen verileri (kaynak domain) yüksek başarıyla öğrenebildiği söylenebilir.



Şekil 6.12. CNN modelinin eğitim sonunda elde ettiği eğitim-doğrulama eğrileri.

Eğer eğittiğimiz sınıflandırıcı verilerde yer alan ayırt edici özellikleri *doğru bir şekilde öğrenmeyi başarabildiyse*, test işlemi gerçekleştirilecek veriler ile arasında *ortak özellikler mevcutsa (benzer ise)*, kaynak veri sayısı oldukça *geniş* ve hedef veri *yetersiz* ise, o zaman yeni veriler ile yapılacak bir TL/fine-tuning sürecinin ezberlemeye neden olmayacağı ve hedef domain de yer alan görüntülerde ki sınıflandırma performansını üst noktalara taşıyacağı varsayılabilir [98,99]. Diğer yandan, DL'nin verilerin *hiyerarşik olarak yüksek seviye özelliklerini* öğrenebilmesi sayesinde, kaynak veride yer alan *farklı varyasyonlardaki gizli faktörler* açığa çıkarılarak *transfer edilebilir sunumları* öğrenebilmesi sağlanırsa, hiyerarşik olarak domain değişmez özellikleri *benzerliklerine göre* gruplandırabilir, bu sayede, hedef domain üzerinde de yüksek başarıya ulaşılabilir [116,117].

Sığ DA yöntemlerinde, özellikler önceden belirlenerek sabitlenir. Derin modellerde ise transfer edilebilir özellikler modelin kendisi tarafından öğrenilerek hedef veriye adaptasyonu gerçekleştirilir. Bu da derin modellerin başarısını arttıran en önemli etkenlerdendir [112]. Daha önce eğitilmiş olan CNN mimarisinin alt katmanları eğitildiği verilerin genel özelliklerine yoğunlaşırken, üst katmanlar verinin daha spesifik özelliklerine odaklanırlar. Bu sebepten eğitilen modelin temel çıkarımlar yaptığı ilk bloktaki ağırlık değerleri dondurulurken (2 konvolüsyon katmanı, 1 pooling), kalan iki bloğun ağırlık değerleri güncellenecek şekilde bir ince-ayar sürecine başlanmıştır. Bu sayede modelin *sınıf bozulması (label corruption)*, *sınıf dengesizliği (class imbalance)* ve *düşman ataklarına (adversarial)* karşı dayanıklı olması da sağlanmıştır [161].

Eğer hedef veride etiketli veri sayısı fazla ise DA gerçekleştirmek gereksizdir. Bunun yerine etiketli veriler ile danışmanlı bir eğitim yapmak, daha sonra ise etiketsiz verilerde test gerçekleştirmek mantıklı olacaktır. Bu nedenle, hedef domainde yer alan verilerin sadece %15'ini eğitim %85'ini test olarak ayırarak eğitim sürecine başlanmıştır (Çizelge 6.3). İnce-ayar sürecinde de *10-fold çapraz doğrulama* kullanılmış ve 200 devir için eğitilmiştir.

Çizelge 6.3. Eğitim ve test setleri.

Veri seti	Özellikler	Sınıflar					Toplam
		Bazofil	Eozinofil	Lenfosit	Monosit	Nötrofil	
Elen – I [132]	Eğitim	1	1	6	1	10	19
	Test	-	-	32	3	58	93
Elen – II [132]	Eğitim	-	-	1	1	3	5
	Test	-	-	1	1	16	18
Elen – III [132]	Eğitim	-	-	2	-	8	10
	Test	-	-	12	-	45	57
MISP – I [133]	Eğitim	1	1	8	2	20	32
	Test	-	4	46	8	112	170
MISP – II [134]	Eğitim	1	1	4	2	9	17
	Test	3	2	23	10	49	87
BCCD [135]	Eğitim	1	6	6	2	21	36
	Test	1	36	32	14	122	205
SigTuple [60]	Eğitim	1	1	6	1	7	16
	Test	7	4	33	5	42	91
Cellavision [136]	Eğitim	1	1	14	5	36	57
	Test	1	4	80	30	204	319
ALL-IDB [131]	Eğitim	1	1	9	1	3	15
	Test	-	1	51	2	15	69

Gerçekleştirilen ilk deneyde modelin her bir veri seti ile ayrı ayrı performansı ölçülmüştür (Şekil 6.13). Bu süreç *bir kaynak bir hedef DA* olarak da değerlendirilebilir. Karmaşıklık matrisleri incelendiğinde modelin test için ayrılan görüntüleri yüksek başarı ile sınıflandırabildiği görülmüştür. Tahmin etme değerleri en az %83,33, en fazla ise %98,25 olarak ölçülmüştür. Modeli olumsuz etkileyen en büyük etken ise ince-ayar sürecinde kullanılan eğitim verisinin dengesizliği olmuştur. Buradan, sınıfları dengeleyecek işlemlerin akabinde modelin başarısının daha yüksek noktalara taşınabileceği çıkarılabilir.

True Basophil	0	0	0	0	0	0	0.00%	0.00%
True Eosinophil	0	0	0	0	0	0	0.00%	0.00%
True Lymphocyte	0	0	27 29.03%	3 3.23%	2 2.15%	32	84.38%	15.43%
True Monocyte	0	0	2 2.15%	1 1.08%	0	3	33.33%	66.67%
True Neutrophil	0	3 3.23%	0	2 2.15%	53 56.99%	58	91.38%	8.62%
Precision	0 0.00%	3 100.00%	29 83.10%	6 16.67%	55 96.36%	93	87.10%	12.90%
	Basophil	Eosinophil	Lymphocyte	Monocyte	Neutrophil	Sensitivity		
	Predicted							

a) Elen-I

True Basophil	0	0	0	0	0	0	0.00%	0.00%
True Eosinophil	0	0	0	0	0	0	0.00%	0.00%
True Lymphocyte	0	0	1 5.56%	0	0	1	100%	0.00%
True Monocyte	0	0	1 5.56%	0	0	1	0.00%	100.00%
True Neutrophil	0	0	1 5.56%	1 5.56%	14 77.78%	16	87.50%	12.50%
Precision	0 0.00%	0 0.00%	3 33.33%	1 100.00%	14 100%	18	83.33%	16.67%
	Basophil	Eosinophil	Lymphocyte	Monocyte	Neutrophil	Sensitivity		
	Predicted							

b) Elen-II

True Basophil	0	0	0	0	0	0	0.00%	0.00%
True Eosinophil	0	0	0	0	0	0	0.00%	0.00%
True Lymphocyte	0	0	11 19.50%	0	1 1.75%	12	91.67%	8.33%
True Monocyte	0	0	0	0	0	0	0.00%	0.00%
True Neutrophil	0	0	0	0	45 78.95%	45	100%	0.00%
Precision	0 0.00%	0 0.00%	11 100%	0 0.00%	46 97.92%	57	86.14%	13.86%
	Basophil	Eosinophil	Lymphocyte	Monocyte	Neutrophil	Sensitivity		
	Predicted							

c) Elen-III

True Basophil	0	0	0	0	0	0	0.00%	0.00%
True Eosinophil	0	3 1.76%	1 0.59%	0	0	4	75.00%	25.00%
True Lymphocyte	0	0	41 24.12%	0	5 2.94%	46	89.13%	10.87%
True Monocyte	0	0	1 0.59%	6 3.53%	1 0.59%	8	75.00%	25.00%
True Neutrophil	0	1 0.59%	8 4.71%	1 0.59%	102 60.00%	112	91.07%	8.93%
Precision	0 0.00%	4 75.00%	51 80.39%	7 88.71%	108 94.44%	170	87.33%	12.67%
	Basophil	Eosinophil	Lymphocyte	Monocyte	Neutrophil	Sensitivity		
	Predicted							

d) MISP-I

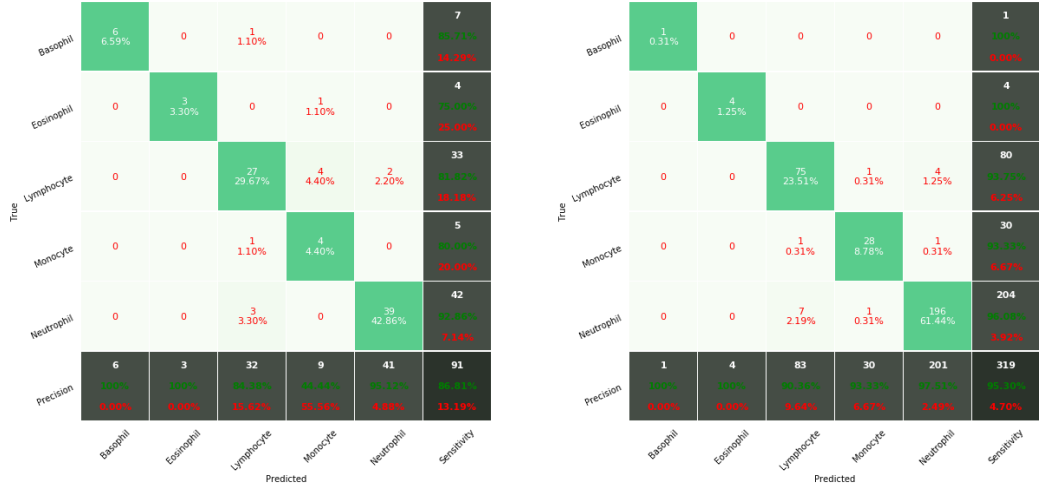
True Basophil	3 3.45%	0	0	0	0	3	100%	0.00%
True Eosinophil	0	1 1.15%	0	1 1.15%	0	2	50.00%	50.00%
True Lymphocyte	1 1.15%	0	20 22.99%	0	2 2.30%	23	86.96%	13.04%
True Monocyte	0	0	2 2.30%	6 6.90%	2 2.30%	10	60.00%	40.00%
True Neutrophil	0	0	3 3.45%	2 2.30%	44 50.37%	49	88.00%	12.00%
Precision	4 75.00%	1 100%	25 80.00%	9 86.67%	48 91.67%	87	88.00%	12.00%
	Basophil	Eosinophil	Lymphocyte	Monocyte	Neutrophil	Sensitivity		
	Predicted							

e) MISP-II

True Basophil	1 0.49%	0	0	0	0	1	100%	0.00%
True Eosinophil	0	39 14.63%	2 0.98%	1 0.49%	3 1.46%	36	83.33%	16.67%
True Lymphocyte	0	1 0.49%	27 13.17%	3 1.46%	1 0.49%	32	84.38%	15.62%
True Monocyte	0	0	2 0.98%	12 5.85%	0	14	85.71%	14.29%
True Neutrophil	0	4 1.95%	9 4.39%	2 0.98%	107 52.20%	122	87.70%	12.30%
Precision	1 100%	35 85.71%	40 87.50%	18 88.67%	111 96.44%	205	88.30%	11.70%
	Basophil	Eosinophil	Lymphocyte	Monocyte	Neutrophil	Sensitivity		
	Predicted							

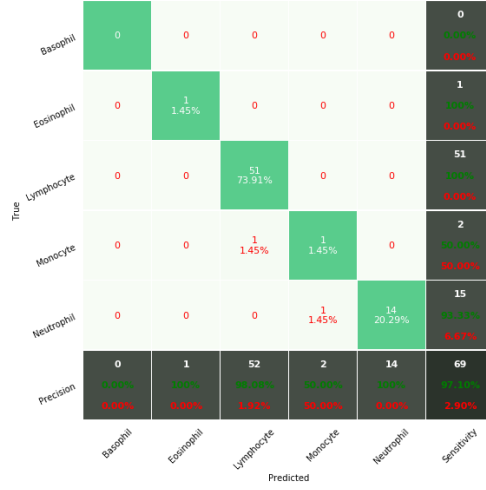
f) BCCD

Şekil 6.13. Sınıflandırıcı modelin hedef domainler üzerinde ulaştığı başarıları gösteren karmaşıklık matrisleri.



g) SigTuple

h) Cellavision



i) ALL-IDB

Şekil 6.13. (devam ediyor).

İkinci deneyde ise modelin tüm hedef domainler üzerindeki genel başarısı bir çoklu hedef DA süreci oluşturularak incelenmiştir. Bu amaçla, hedef domainde yer alan tüm veriler bir veri seti altında birleştirilerek bir ince-ayar süreci ve test gerçekleştirilmiştir (Çizelge 6.4). Veri seti %15'i eğitim ve %85'i test olacak şekilde ikiye ayrılmış ve yine 200 devir için eğitilmiştir.

Çizelge 6.4. Oluşturulan genel veri seti.

Veri seti	Özellikler	Sınıflar					Toplam
		Bazofil	Eozinofil	Lenfosit	Monosit	Nötrofil	
Tüm	Eğitim	3	5	53	13	123	197
	Test	16	58	313	75	657	1,119
	Toplam	19	63	366	88	780	1,316

Eğitim sonunda test verisi ile gerçekleştirilen sınıflandırmada, modelin tahmin başarısı %95,35 olarak ölçülmüştür (Şekil 6.14). Daha sonra karmaşıklık matrisi kullanılarak sınıf bazında ve genel olarak modelin ulaştığı başarı, istatistiksel metrikler hesaplanarak incelenmiştir (Çizelge 6.5).

True	Basophil	13 1.16%	0	1 0.09%	0	2 0.18%	16 81.25% 18.75%
	Eosinophil	0	48 4.29%	3 0.27%	2 0.18%	5 0.45%	58 82.76% 17.24%
	Lymphocyte	1 0.09%	0	296 26.45%	4 0.36%	12 1.07%	313 94.57% 5.43%
	Monocyte	0	0	8 0.71%	62 5.54%	5 0.45%	75 82.67% 17.33%
	Neutrophil	3 0.27%	4 0.36%	2 0.18%	0	648 57.91%	657 98.63% 1.37%
	Precision	17 76.47% 23.53%	52 92.31% 7.69%	310 95.43% 4.12%	68 91.18% 8.82%	672 96.43% 1.57%	1119 98.35% 4.65%
		Basophil	Eosinophil	Lymphocyte	Monocyte	Neutrophil	Sensitivity
		Predicted					

Şekil 6.14. Tüm veri setleri birleştirerek yapılan sınıflandırma sonuçlarını içeren karmaşıklık matrisi (çoklu-hedef DA).

Çizelge 6.5. Tüm verilerle yapılan deneyin sınıflandırma raporu.

WBC alt-tipi	Duyarlılık (%)	Özgüllük (%)	Hassasiyet (%)	Doğruluk (%)
Bazofil	81.25	99.62	76.47	99.35
Eozinofil	82.76	99.61	92.31	98.70
Lenfosit	94.57	98.22	95.48	97.18
Monosit	82.67	99.41	91.18	98.25
Nötrofil	98.63	94.58	96.43	97.00
Ort./Toplam	95.35	98.80	95.35	98.09

Sonuçlar incelendiğinde genel olarak model için *domain farkının ortadan kalktığı*, modelin kaynak verilerde yer alan ve hedef domain de de *ortak olan özellikleri* başarılı bir şekilde öğrenebildiği, ince-ayar süreci sırasında hedef verilerdeki *domain spesifik verilere adapte* olabildiği ve sonuçta %98,09 gibi oldukça yüksek bir doğruluk sonucuna ulaştığı gözlemlenmiştir. Sonuçlar sınıf bazında incelendiğinde ise en düşük duyarlılık (sens.) ve hassasiyet (prec.) değerleri, veri setlerinde en az sayıda örnek içeren sırasıyla Bazofil, Eozinofil ve Monosit hücreleri için gözlemlenmiştir. Sınıflandırıcı model dengesiz veri setleriyle iyi bir şekilde başa çıkabilse de yine de bu durumdan olumsuz etkilendiği söylenebilir. Fakat ilk teste göre değerlendirildiğinde, veri sayılarındaki artışın sonuçlara olumlu yansıdığı görülmüştür. Dolayısıyla, sınıflar da yer alan örnek sayılarında yapılacak dengelemeler sayesinde (yukarı örnekleme veya aşağı örnekleme) (up-sampling veya down-sampling) başarının artacağı öngörülebilir.

6.4. TARTIŞMA

Bir uygulama tasarlarken öncelikle problem belirlenmelidir [130]. Her problem de kendi çözümünü doğurur, dolayısıyla uygulanan bir yöntemin her problem üzerinde başarılı olacağı garanti değildir [112]. WBC veri setlerinde gözlemlenen en büyük problemler veri sayısı yetersizliği ve etiketleme işleminin maliyetli olmasıdır. Diğer yandan, domainlerde yer alan veriler temelde benzer olsa da verilerin görüntülenme sürecindeki değişkenlerin (arka plan, parlaklık vs.) domain farkının oluşmasına yol açtığı görülmüştür.

Bu çalışmanın da genel amacı da bu değişikliklerden etkilenmeyen bir model ortaya koymaktır. Bunu sağlayabilmenin yolu da WBC görüntüleri üzerinde sınıflandırma gerçekleştirecek bu modelin *iyi bir genelleştirme yeteneğine* sahip olmasından geçer. *Domain-değişmez özellikleri kaynak veri setinde iyi öğrenecek bir model, hedef domain üzerinde (eğer benzer özelliklere sahipse) benzetme yapabileceğinden, modelin başarılı olacağı varsayılabilir.* Modelimizi oluştururken öncelikli motivasyonumuz da bu karakteri modele kazandırabilmek olmuştur.

Teorikte eğer kaynak ve hedef veri ilişkili ise bilgi transferi tek seferde yapılabilir. Fakat pratikte çoğu zaman istenilen sonuçlar elde edilemez. Bu gibi durumlarda kaynak ve hedef veriyi yakınlaştıracak bir ara domain hesaplanmalıdır [117]. Çalışmamızın ilk aşamasında gerçekleştirilen veri çoğaltma ve üretici model ara formu hesaplamak için kullanılmıştır. Ara formun domainler arasındaki mesafeyi yakınlaştırması sayesinde DA gerçekleştirecek modelin transfer edilebilir özellikleri öğrenerek, aradaki bağlantıyı oluşturması mümkün olabilmektedir.

Çoklu-hedef DA literatürde yeni çalışılmaya başlanmış bir DA türüdür. Literatürde konuyla ilgili yapılan çalışmalar incelendiğinde [128,162,163] yazarların bizim çalışmamızdakine benzer hipotezler üreterek, kaynak ve hedef arasında adaptasyon gerçekleştirecek modeller önerdikleri görülmüştür. Zaten, bu çalışmada kullanılan yöntemler de TL ve DA literatüründe sıklıkla karşılaşılan ve başarısını ispatlamış yöntemleri (veri çoğaltma/üretici ağlar/ince-ayar) barındırmaktadır. Fakat bu çalışmayı özel kılan nokta bu yöntemlerin bir *boru hattı (pipeline)* halinde, gerçek bir problemde nasıl efektif olarak kullanılabileceğini göstermesidir. Özellikle medikal veri setlerinin kompleks yapılarda olan mikroskobik veriler olması, TL ve DA literatüründe test aşamasında kullanılan veri setleri (Office, Caltech, vb.) üzerinde başarı yakalayan yöntemlerin medikal veri setlerinde ulaşacağı başarı oranlarını sorgular hale getirmektedir. Bu sebeple çalışmamızda, mevcut problem detaylıca analiz edilmiş, ilgili varsayımlar sonrasında, problemle en iyi şekilde başa çıkabilecek yöntemler seçilerek, test aşamasında yüksek başarı oranları yakalanabilmiştir.

Tezin bir önceki TARTIŞMA bölümünde de belirtildiği üzere, eğer elimizdeki veri sayısı yeterince fazla değilse fakat DL yöntemlerinden faydalanılmak isteniyorsa, daha önce başka bir veri seti ile eğitilmiş bir model özellik çıkarıcı olarak kullanılarak, veri setinin sınıflandırmasını gerçekleştirmek mümkündür. Şekil 4.28’de de verildiği üzere TL ve ince-ayar yöntemlerinden hangisinin kullanılacağı sınıflandırma yapılacak veri setinin, daha önce eğitilmiş mimaride kullanılan veri setine benzerliği ve genişliğine göre belirlenmektedir.

Gene bir önceki çalışmanın sonuçlarında görüldüğü üzere veri setlerinin mikroskobik veriler olması ve veri sayılarının yetersiz olması ince-ayar yapılsa dahi verilerin model

tarafından ezberlenmesine yol açmış ve tahmin değerlerin oldukça düşük olmasına yol açmıştır. Dolayısıyla çalışmanın ikinci aşamasında kullanılan veri setlerinin de standart TL ve ince-ayar yöntemleri kullanıldığında benzer sonuçlar vereceği öngörülmektedir.

Eğer veri arttırma teknikleri kullanılarak veri seti boyutu arttırılırsa TL/ince-ayar yöntemleriyle yüksek başarı oranlarının yakalanabileceği düşünülebilir. Fakat veri sayısı yüksek ve daha önce eğitilmiş modeldeki veri seti ile benzer olmadığında TL yöntemleriyle bir sınıflandırma yerine en-baştan bir eğitim öngörülebilir. Fakat gene de [98,164] referanslı çalışmalarda da belirtildiği üzere, sınıflandırıcı modelin sahip olduğu ağırlıkların benzer olmasa dahi bir veri seti ile ilk atamalarının yapılması modelin başarısına yüksek katkı vermektedir.

Fakat WBC veri setlerinin sahip olduğu yetersiz ve dengesiz veri problemi TL/ince-ayar yöntemleri kullanılsa dahi problemin şeklini değiştirmekten ileri gitmemektedir. Bazı veri setlerinin oldukça az sayıda veri içermesi veri arttırma teknikleri için bile yetersizdir. Üretici modeller de veriye aç bir yapıda olması bu kadar az veri ile tatmin edici bir sayıda veri elde edilmesini sağlayamaz.

Veri sayıları ve sınıfların dengeli olması problemlerinin bir veri seti için çözülmüş olduğunu kabul edelim. O zaman bu veri setinin %75-%80 gibi bir kısmını eğitim, geri kalan kısmını ise tahmin için ayırarak verilerin başarılı bir şekilde sınıflandırılması sağlanabilir. Fakat, bir başka veri seti için aynı sınıflandırmanın gerçekleştirilmesi için de tüm bu süreç tekrarlanmalıdır. Her veri seti için yetersiz veri ve dengesiz sınıf probleminin yanı sıra süre ve maliyet gibi problemlerde ortaya çıkmaktadır.

Gerçekleştirilen çalışmada ise önerilen model sayesinde yetersiz veri ve dengesiz sınıflar problemiyle oldukça başarılı bir şekilde başa çıkılabilmektedir. Sahip olunan görüntülerin ufak bir kısmı modelin verilerin spesifik kısımlarını öğrenmesi için kullanılmış, bu sayede verinin büyük bölümünün sınıflandırılması sağlanmıştır. Model bir kez oluşturulduktan sonra, veri setleri değişse dahi süreci tekrarlanmak yerine sadece ince ayar ile çok hızlı bir şekilde adaptasyon süreci sağlanarak test verileri üzerinde tahminler gerçekleştirilebilmiştir.

Çalışmamız sırasında karşılaştığımız en büyük handikap gerek veri çoğaltmadaki optimizasyon sürecinin gerekse üretici modelin işlem açısından oldukça maliyetli olmasıdır. Verilerin çözünürlüğü 32×32 seviyesine çekilmiş olsa dahi oldukça uzun bir eğitim süreciyle karşılaşmıştır. Dolayısıyla veri/özellik üretimi sürecinde verilerin faydalı özelliklerini öğrenecek tekniğin hızlandırılması oldukça önemlidir. Buna yönelik yapılacak çalışmaların başarıya giden yolu daha da kısaltacağı öngörülmektedir³.

³ Tezin bu bölümüne kadar yapılan çalışmadan “Computer Methods and Programs in Biomedicine” dergisinde “Learn from one data set to classify all – A multi-target domain adaption approach for white blood cell classification” başlıklı bir makale çıkarılmıştır.

BÖLÜM 7

GENEL SONUÇLAR

Kan testi sonuçları doktor tarafından incelendiğinde ve bağışıklık sistemini ilgilendiren değerlerde bir problem gözlemlendiğinde, hastadan periferik yayma testi yapması istenir. Bu test maliyeti oldukça düşük olmasına rağmen, kişinin bağışıklık sisteminin kondisyonunu veya sahip olduğu hastalıkların teşhisi/tedavisi için oldukça önemli bilgileri barındırmaktadır. Hematolojist tarafından analizi gerçekleştirilen görüntülerde, hücrelerin sayısı/yoğunluğu gibi niceliklerin yanı sıra WBC'lerin alt-tipleri de incelenmektedir. Bu sayede hastanın bağışıklık sistemi ile ilgili doktor tarafından bir yorum yapılabilmesi mümkün olur. Fakat, bu analizler hematolojist tarafından çoğunlukla mikroskop altında manuel olarak gerçekleştirilmektedir. Bu sebeple oldukça hataya yatkındır ve sürecin uzamasına neden olabilmektedir. Bu probleminin farkında olan araştırmacılar da uzun yıllardır, hem görüntüler üzerinde yüksek doğrulukla tespit ve sınıflandırma yapacak yöntemler önermekte, hem de süreci daha hızlı hale getirecek tam otomatik CAD sistemleri üretmektedirler.

Medikal görüntülerin kompleks yapılarda olması, bu görüntüler ile gerçekleştirilecek bir analizi, en zor görüntü işleme görevlerinden birisi yapmaktadır. Medikal görüntülerin belli bir standartta olmamasından dolayı (farklı renk kanalı sayısı, farklı çözünürlükler, farklı odaklanmalar vs.), segmentasyon ve/veya sınıflandırma gerçekleştirmek isteyen araştırmacılar tarafından geçmişten günümüze çeşitli yöntemler önerilmiştir. Eğer bu yöntemler analiz sonrasında elde edilecek özellik vektörleri ile klasik makine öğrenmesi kullanmayı gerektiriyorsa, görüntülerin uzun bir ön-işlem sürecinden geçirilmesi gerekmektedir. Diğer yandan, elde edilen anlamlı bilgilerle yapılacak hesaplamaların başarısı ön-işleme sürecindeki başarı oranı ile direkt irtilidir. Bu yüzden araştırmacılar her geçen gün daha hızlı, daha başarılı ve daha global bir yöntem bulmaya yönelik çalışmalar ortaya koymaktadırlar.

DL, geçtiğimiz birkaç yılda oldukça popüler olmuş makine öğrenmesi yöntemlerindedir. Bu popülerliğin en önemli nedenlerinden bir tanesi, bu yöntem sayesinde baştan sona bir eğitim gerçekleştirilerek, ön-işleme sürecinin yol açtığı olumsuzlukların önüne geçilebilmesidir. Fakat her yöntem de olduğu gibi bu yöntemde bazı handikapları mevcuttur. Bunlardan bir tanesi de veri sayısına bağımlı bir yapıda olmasıdır. Veri arttırma teknikleri uygulanarak mevcut verilerin farklı varyasyonlarından maksimum çıkarımların sağlanması mümkün olsa da eğer veri sayısı az ise eğitim verisinin ezberlenmesinin önüne geçilemeyecek ve test verisi üzerinde eğitim sürecinde ki başarılar yakalanamayacaktır. Bu nedenle, az sayıdaki veri üzerinde maksimum çıkarımları yapabilecek ve baştan sona eğitim sürecinde, hiper-parametreler üzerinde gerekli esnekliği sağlayarak, başarılı bir modellemenin önünü açan bir yöntem gerekmektedir.

Kapsül Ağları DL yöntemlerinin sahip olduğu bu gibi problemlerin çözümünde kullanılabilecek bir metot olarak karşımıza çıkmaktadır. İnsan beyninin görüntüleri hiyerarşik bir düzende sunmasını ve işlemesini taklit eden yapısı ile, görüntülerin poz ve çeviri değişikliklerinden olumsuz etkilenmeyerek doğru bir analiz gerçekleştirme imkânı sağlar. Ayrıca, küçük veri sayılarında dahi kararlı bir modelleme yapabilme yeteneğine sahiptir.

Bu tez çalışmasının ilk bölümünde, insan morfolojisi hakkında önemli bilgiler sağlayan periferik yayma testi görüntüleri üzerinde, WBC'lerin analizi ve sınıflandırılması Kapsül Ağları ile gerçekleştirilmiştir. Çalışmanın ilk aşamasında, orijinal modele göre bazı modifikasyonlarla geliştirilen ağ ile, WBC'ler sıfırdan ve baştan sona bir eğitim süreci ile incelenmiştir. Çalışma düşük sayıdaki bir veri seti ile gerçekleştirilmesine rağmen oldukça yüksek başarı değerlerine ulaşılmıştır (%96,86).

Daha sonra ise, Kapsül Ağlarının yetenekleri diğer DL yöntemleri ile kıyaslanmıştır. Burada, Kapsül Ağlarının diğer yöntemlere göre sağladığı avantajlar tartışılarak aynı veri seti üzerinde elde edilen başarı oranları değerlendirilmiştir. Böylece, Kapsül Ağlarının, herhangi bir veri arttırma tekniği kullanmadan ve ön-işleme süreci gerçekleştirilmeden, küçük veri setleri üzerinde diğer modellere nazaran daha yüksek başarı oranları yakalayabildiği gösterilmiştir.

Buraya kadar yapılan çalışmanın sonuçları felsefik açıdan değerlendirilirse, analizi gerçekleştirilen medikal görüntüler hakkında daha önce herhangi bir bilgiye sahip olmayan makinenin, sınırlı sayıdaki görüntü üzerinde neredeyse bir hematolojist başarısına ulaşmış olması yapay zekanın günümüzde geldiği noktayı göstermesi açısından oldukça önemlidir.

Ne yazık ki, periferik yayma testi sırasında uygulanan prosedür farklılıkları ve görüntüleme cihazlarının sahip olduğu özellikler, WBC alt-tiplerini sınıflandıracak model için başka bir problem ortaya çıkarır: *domain shift*. Bu problemin yol açtığı olumsuzluklar nedeniyle belli bir veri seti üzerinde yüksek başarıyla çalışan model, diğer veri setleri üzerinde beklenen başarının çok altında kalır. Bu nedenle araştırmacılar bu problemle başa çıkabilmek adına, ya her veri seti için ayrı bir model önermek zorunda kalmakta, ya da TL ve DA yöntemleri ile veri setleri arasındaki ilişkileri açığa çıkararak, bir veri setinde öğrenilen bilgileri diğer veri setlerinde de efektif olarak kullanmaya çalışırlar. Fakat, TL ve DA sayesinde kaynak bir domainden hedef domaine başarılı bir şekilde geçmek mümkün olsa da yeni veri setleri ile karşılaşıldığında oluşturulan modelin yine bir başarısızlıkla karşılaşması olasıdır. Bu da her seferinde yeni bir adaptasyon süreci, yani zaman ve maliyet anlamına gelmektedir.

WBC veri setleri incelenerek yapılan çıkarımlar doğrultusunda, tez çalışmasının ikinci bölümünde, yalnızca bir veri seti kullanılmış, bu veri setinin sahip olduğu *temel özellikleri* öğrenirken, arka planda yer alan *değişikliklerden olumsuz etkilenmeyecek* bir model elde etmek hedeflenmiştir. Bu doğrultuda *AutoAugment* süreci kullanılarak, eğitilecek modele *genelleştirme yeteneği* kazandıracak, ama aynı zamanda *negatif transfer* fenomeninden de kaçınmayı sağlayacak örnekler oluşturulmuştur. Bu sayede bir domainden birden fazla domain üretilerek, *çoklu kaynak DA'nın* sunduğu avantajlarından biri olan *her bir domainden doğru bilgilerin çıkarımı* sağlanmıştır. Sonrasında ise *WGAN-GP* kullanılarak domainler arasındaki *transfer edilebilir özellikleri* barındıran ve *domainler arasındaki mesafeleri azaltacak* örnekler üretilmiştir. Bu sayede *domain karmaşıklığı* artırılarak, öğrenmenin *maksimum seviyede* olması sağlanmıştır. Daha sonrasında ise *CNN* ve *ince-ayar* sayesinde, öğrenilmiş bilgilerin hedef domainlere *adaptasyonu* sağlanmıştır. WBC'lerin alt-

tiplerinin *ayırt edici özelliklerini* öğrenen modelin, farklı domainlerde yer alan *domain spesifik özelliklere* adaptasyonu sağlanarak, modelin, hedef domainde yer alan veriler ile bir nevi “danışmanlı” ve “baştan-sona” bir eğitim süreci geçirmiş gibi davranması sağlanmıştır.

Adaptasyon sürecinde kullanılan *CNN* modeli üretilen *40000* görüntü ile eğitilmiş, eğitim sonunda en yüksek doğrulama değeri *0.9998* olarak ölçülmüştür. Daha sonra kaydedilen modelin, görüntülerin genel özelliklerini çıkaran ilk üç katmanında yer alan ağırlık değerleri korunurken, daha spesifik özelliklere odaklanan kalan altı katmanı eğitime dahil edilerek *ince-ayar* gerçekleştirilmiştir. Gerçekleştirilen deneylerin ilkinde modelin ayrı ayrı hedef domainlere adaptasyon süreci incelenmiştir. Bu amaçla, hedef domainde yer alan dokuz veri setinin ayrı ayrı *%15*'i ile *ince-ayar*, *%85*'i ile test gerçekleştirilmiştir. Test sonunda *%83,33 – %98,25* arasında değişen tahmin değerlerine ulaşılmıştır. Modelin hedef domainde yer alan görüntülere yüksek doğrulukla cevap verebildiği, fakat dengesiz veri seti probleminden olumsuz etkilendiği görülmüştür. Deneylerin ikinci aşamasında ise, modelin *çoklu-hedef DA* performansını ölçmek adına, hedef domainde yer alan tüm veriler aynı çatı altında toplanarak birleştirilmiş ve yine *%15*'i ile *ince-ayar* yapılarak kalan görüntülerde test yapılmıştır. Bu test sonunda *%95,35* tahmin başarısına ulaşılrken, modelin tüm sınıflar üzerinde toplam *%98,09* doğruluğa ulaştığı görülmüştür. Böylece model için artık domain farkının problem olmaktan çıktığı, dengesiz veri seti probleminin çözümüne yönelik yapılacak işlemler sayesinde başarının daha da arttırabileceği gösterilmiştir.

Modelin oluşturulma süreci uzun ve maliyetli olması bir handikap olsa da modelin yeni veri setlerine *adaptasyon süreci (ince-ayar)* oldukça hızlı bir şekilde gerçekleştirilebilmektedir. Bu tip bir modelleme sayesinde gelecekte hematolojist tarafından etiketlenen az sayıda veri kullanılarak oldukça fazla sayıda hücrenin hızlı ve doğru bir şekilde tanınması/sınıflandırılması sağlanabilecektir. Diğer yandan makine öğrenmesinin doğası gereği, *ince-ayar* gerçekleştirilen modelin güncellenen ağırlıklarının kaydedilerek, her yeni gelen veri seti ile model kendini daha da güçlü hale getirilebilecektir. Bu şekilde kendini sürekli geliştiren modelin, ileri de danışmansız bir şekilde yeni verileri sınıflandırabilmesi mümkündür.

KAYNAKLAR

1. Alberts, B., Johnson, A. D., Lewis, J., Morgan, D., Raff, M., Roberts, K., and Walter, P., "Molecular Biology of the Cell", 6. Ed., *W. W. Norton & Company*, New York, NY, US, (2014).
2. Adewoyin, A. and Nwogoh, B., "Peripheral blood film - A review", *Annals Of Ibadan Postgraduate Medicine*, 12 (2): 71–79 (2014).
3. Kumar, V., Abbas, A. K., and Aster, J. C., "Robbins Basic Pathology", 10. Ed., *Saunders*, Philadelphia, PA, 928 (2012).
4. Habibzadeh, M. and Krzyz, A., "Comparative study of shape, intensity and texture features and support vector machine for white blood cell classification", *Journal Of Theoretical And Applied Computer Science*, 7 (1): 20–35 (2013).
5. Dorini, L. B., Minetto, R., and Leite, N. J., "Semi-automatic white blood cell segmentation based on multiscale analysis", *IEEE Journal Of Biomedical And Health Informatics*, 17 (1): 250–256 (2013).
6. Rawat, J., Singh, A., Bhadauria, H. S., and Virmani, J., "Computer aided diagnostic system for detection of leukemia using microscopic images", *Procedia Computer Science*, 70 (2015): 748–756 (2015).
7. Suzuki, K., "Overview of deep learning in medical imaging", *Radiological Physics And Technology*, 10 (3): 257–273 (2017).
8. Shen, D., Wu, G., and Suk, H.-I., "Deep learning in medical image analysis", *Annual Review Of Biomedical Engineering*, 19 (1): 221–248 (2017).
9. Lee, J.-G., Jun, S., Cho, Y.-W., Lee, H., Kim, G. B., Seo, J. B., and Kim, N., "Deep learning in medical imaging: General overview", *Korean Journal Of Radiology*, 18 (4): 570–584 (2017).
10. Perez, L. and Wang, J., "The effectiveness of data augmentation in image classification using deep learning", *ArXiv:1712.04621 [Cs]*, (2017).
11. Storkey, A., Corfield, D., Hein, M., Hansen, L. K., Ben-David, S., Kanamori, T., Shimodaira, H., Rubens, N., Müller, K.-R., Gretton, A., Smola, A. J., Huang, J., Schmittfull, M., Borgwardt, K., Schölkopf, B., Bickel, S., Brückner, M., Scheffer, T., Globerson, A., Teo, C. H., and Roweis, S., "Dataset Shift in Machine Learning", *The MIT Press*, Cambridge, MA, US, 229 (2008).
12. Blanchard, G., Lee, G., and Scott, C., "Generalizing from several related classification tasks to a new unlabeled sample", *Advances in Neural Information Processing Systems 24*, Granada, Spain, (2011).

13. Elen, A., "Periferik yayma sonuçlarının otomatik analizi için zeki denetimli sistem tasarımı", Doktora Tezi, *Karabük Üniversitesi Fen Bilimleri Enstitüsü*, Karabük, Türkiye, (2018).
14. Bain, B. J., Bates, I., and Laffan, M. A., "Dacie and Lewis Practical Haematology", 11. Ed., *Churchill Livingstone*, London, UK, (2012).
15. Internet: Eldridge, L., "Types and Function of White Blood Cells (WBCs)", <https://www.verywellhealth.com/understanding-white-blood-cells-and-counts-2249217> (2019).
16. Internet: scikit-learn, "Choosing the Right Estimator", http://scikit-learn.org/stable/tutorial/machine_learning_map/ (2018).
17. Deng, L. and Yu, D., "Deep learning: Methods and applications", *Foundations And Trends® In Signal Processing*, 7 (3–4): 197–387 (2014).
18. Shen, D., "Deep learning in medical imaging analysis", *19th International Conference on Medical Image Computing & Computer Assisted Intervention*, Athens, Greece, (2016).
19. Bengio, Y., "Learning deep architectures for AI", *Foundations And Trends® In Machine Learning*, 2 (1): 1–127 (2009).
20. LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning", *Nature*, 521 (7553): 436–444 (2015).
21. Arel, I., Rose, D. C., and Karnowski, T. P., "Deep machine learning - A new frontier in artificial intelligence research", *IEEE Computational Intelligence Magazine*, 5 (4): 13–18 (2010).
22. Bengio, Y., Courville, A., and Vincent, P., "Representation learning: A review and new perspectives", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 35 (8): 1798–1828 (2013).
23. Bottou, L., "From machine learning to machine reasoning: An essay", *Machine Learning*, 94 (2): 133–149 (2014).
24. Hinton, G. E., Osindero, S., and Teh, Y.-W., "A fast learning algorithm for deep belief nets", *Neural Computation*, 18 (7): 1527–1554 (2006).
25. Långkvist, M., Karlsson, L., and Loutfi, A., "A review of unsupervised feature learning and deep learning for time-series modeling", *Pattern Recognition Letters*, 42 (1): 11–24 (2014).
26. Schmidhuber, J., "Deep learning in neural networks: An overview", *Neural Networks*, 61 (2015): 85–117 (2015).
27. Zhou, S. K., Greenspan, H., and Shen, D., "Deep Learning for Medical Image Analysis", 1. Ed., *Academic Press*, London; San Diego, 458 (2017).

28. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., and Sánchez, C. I., "A survey on deep learning in medical image analysis", *Medical Image Analysis*, 42 (Supplement C): 60–88 (2017).
29. Internet: Geitgey, A., "Machine Learning Is Fun! Part 3: Deep Learning and Convolutional Neural Networks", <https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721> (2018).
30. Internet: ujjwalkarn, "An Intuitive Explanation of Convolutional Neural Networks", <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/> (2018).
31. Fergus, R., Yu, K., Ranzato, M. A., Salakhutdinov, R., and Taylor, G., "Deep learning & feature learning methods for vision (CVPR 2012 Tutorial)", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, New York University, NY, US, (2012).
32. Internet: Machine Learning Guru, "Understanding Convolutional Layers in Convolutional Neural Networks (CNNs)", http://machinelearningguru.com/computer_vision/basics/convolution/convolution_layer.html (2018).
33. Nair, V. and Hinton, G. E., "Rectified linear units improve restricted boltzmann machines", *27th International Conference on Machine Learning (ICML '10)*, Haifa, Israel, (2010).
34. Fergus, R., "Neural Networks", *The Machine Learning Summer School (2015)*, Max Planck Institute For Intelligent Systems, Tübingen, Germany, (2015).
35. Internet: DeepLearning.Net, "Convolutional Neural Networks (LeNet)", <http://deeplearning.net/tutorial/lenet.html> (2019).
36. Internet: Karpathy, A., "Convolutional Neural Networks: Architectures, Convolution / Pooling Layers", <http://cs231n.github.io/convolutional-networks/> (2018).
37. Internet: Gwardys, G., "Convolutional Neural Networks Backpropagation: From Intuition to Derivation", <https://grzegorzwardys.wordpress.com/2016/04/22/8/> (2018).
38. Internet: Jefkine, "Backpropagation in Convolutional Neural Networks", <http://www.jefkine.com//general/2016/09/05/backpropagation-in-convolutional-neural-networks/> (2018).
39. Internet: Otuyama, J. M., "Introduction to Convolutional Neural Networks", <https://www.ime.usp.br/~otuyama/academic/cnn/index.html> (2018).
40. Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y., "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations",

26th Annual International Conference on Machine Learning (ICML '09), Montreal, Quebec, Canada, (2009).

41. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition", *Proceedings Of The IEEE*, 86 (11): 2278–2324 (1998).
42. Internet: Harley, A., "3D Visualization of a Convolutional Neural Network", <http://www.cs.cmu.edu/~aharley/vis/conv/> (2018).
43. He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, Las Vegas, NV, US, (2016).
44. Piuri, V. and Scotti, F., "Morphological classification of blood leucocytes by microscope images", *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, Boston, MA, US, (2004).
45. Ramoser, H., Laurain, V., Bischof, H., and Ecker, R., "Leukocyte segmentation and classification in blood-smear images", *IEEE Engineering in Medicine and Biology 27th Annual Conference*, Shanghai, China, (2005).
46. Theera-Umpon, N., "White blood cell segmentation and classification in microscopic bone marrow images", *Second International Conference Fuzzy Systems and Knowledge Discovery*, Changsha, China, (2005).
47. Mrčić, S. and Jorgovanović, N., "Application of neural network for automatic classification of leukocytes", *8th Seminar on Neural Network Applications in Electrical Engineering*, Belgrade, Serbia & Montenegro, Serbia, (2006).
48. Ramirez-Cortes, J. M., Gomez-Gil, P., Alarcon-Aquino, V., Gonzalez-Bernal, J., and Garcia-Pedrero, A., "Neural networks and SVM-based classification of leukocytes using the morphological pattern spectrum", *Soft Computing for Recognition Based on Biometrics*, Springer, Berlin, Heidelberg, 19–35 (2010).
49. Xie, E., McGinnity, T. M., and Wu, Q., "Automatic extraction of shape features for classification of leukocytes", *International Conference on Artificial Intelligence and Computational Intelligence*, Sanya, China, (2010).
50. Ko, B. C., Gim, J. W., and Nam, J. Y., "Cell image classification based on ensemble features and random forest", *Electronics Letters*, 47 (11): 638–639 (2011).
51. Rezatofighi, S. H. and Soltanian-Zadeh, H., "Automatic recognition of five types of white blood cells in peripheral blood", *Computerized Medical Imaging And Graphics*, 35 (4): 333–343 (2011).
52. Ramesh, N., Dangott, B., Salama, M. E., and Tasdizen, T., "Isolation and two-step classification of normal white blood cells in peripheral blood smears", *Journal Of Pathology Informatics*, 3 (13): (2012).

53. Stadelmann, J. V. and Spiridonov, I. N., "Automated classification of leukocytes in blood smear images", *Biomedical Engineering*, 46 (2): 83–87 (2012).
54. Joshi, M. D., Karode, A. H., and Suralkar, S. R., "White blood cells segmentation and classification to detect acute leukemia", *International Journal Of Emerging Trends & Technology In Computer Science (IJETTCS)*, 2 (3): 147–151 (2013).
55. Sarrafzadeh, O., Rabbani, H., Talebi, A., and Banaem, H. U., "Selection of the best features for leukocytes classification in blood smear microscopic images", *SPIE Medical Imaging*, San Diego, California, US, (2014).
56. Nazlibilek, S., Karacor, D., Ercan, T., Sazli, M. H., Kalender, O., and Ege, Y., "Automatic segmentation, counting, size determination and classification of white blood cells", *Measurement*, 55: 58–65 (2014).
57. Su, M.-C., Cheng, C.-Y., and Wang, P.-C., "A neural-network-based approach to white blood cell classification", *The Scientific World Journal*, 2014: (2014).
58. Habibzadeh, M., Krzyżak, A., and Fevens, T., "White blood cell differential counts using convolutional neural networks for low resolution images", *12th International Conference, ICAISC 2013*, Zakopane, Poland, (2013).
59. Li, X., Li, W., Xu, X., and Hu, W., "Cell classification using convolutional neural networks in medical hyperspectral imagery", *2nd International Conference on Image, Vision and Computing (ICIVC)*, Chengdu, China, (2017).
60. Mundhra, D., Cheluvraju, B., Rampure, J., and Rai Dastidar, T., "Analyzing microscopic images of peripheral blood smear using deep learning", *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Québec City, QC, Canada, (2017).
61. Yu, W., Chang, J., Yang, C., Zhang, L., Shen, H., Xia, Y., and Sha, J., "Automatic classification of leukocytes using deep neural network", *IEEE 12th International Conference on ASIC (ASICON)*, Guiyang, China, (2017).
62. Shahin, A. I., Guo, Y., Amin, K. M., and Sharawi, A. A., "White blood cells identification system based on convolutional deep neural learning networks", *Computer Methods And Programs In Biomedicine*, 168: 69–80 (2017).
63. Zhao, J., Zhang, M., Zhou, Z., Chu, J., and Cao, F., "Automatic detection and classification of leukocytes using convolutional neural networks", *Medical & Biological Engineering & Computing*, 55 (8): 1287–1301 (2017).
64. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z., "Rethinking the Inception architecture for computer vision", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, Las Vegas, NV, US, (2016).
65. Habibzadeh, M., Jannesari, M., Rezaei, Z., Baharvand, H., and Totonchi, M., "Automatic white blood cell classification using pre-trained deep learning

- models: ResNet and Inception", *Tenth International Conference on Machine Vision (ICMV 2017)*, Vienna, Austria, (2018).
66. Jiang, M., Cheng, L., Qin, F., Du, L., and Zhang, M., "White blood cells classification with deep convolutional neural networks", *International Journal Of Pattern Recognition And Artificial Intelligence*, 32 (9): 1857006 (2018).
 67. Liang, G., Hong, H., Xie, W., and Zheng, L., "Combining convolutional neural network with recursive neural network for blood cell image classification", *IEEE*, 6: 36188–36197 (2018).
 68. Qin, F., Gao, N., Peng, Y., Wu, Z., Shen, S., and Grudtsin, A., "Fine-grained leukocyte classification with deep residual learning for microscopic images", *Computer Methods And Programs In Biomedicine*, 162 (2018): 243–252 (2018).
 69. Thanh, T. T. P., Vununu, C., Atoev, S., Lee, S.-H., and Kwon, K.-R., "Leukemia blood cell image classification using convolutional neural network", *International Journal Of Computer Theory And Engineering*, 10 (2): 54–58 (2018).
 70. Tiwari, P., Qian, J., Li, Q., Wang, B., Gupta, D., Khanna, A., Rodrigues, J. J. P. C., and de Albuquerque, V. H. C., "Detection of subtype blood cells using deep learning", *Cognitive Systems Research*, 52: 1036–1044 (2018).
 71. Vogado, L. H. S., Veras, R. M. S., Araujo, Flavio. H. D., Silva, R. R. V., and Aires, K. R. T., "Leukemia diagnosis in blood slides using transfer learning in CNNs and SVM for classification", *Engineering Applications Of Artificial Intelligence*, 72 (2018): 415–422 (2018).
 72. Choi, J. W., Ku, Y., Yoo, B. W., Kim, J.-A., Lee, D. S., Chai, Y. J., Kong, H.-J., and Kim, H. C., "White blood cell differential count of maturation stages in bone marrow smear using dual-stage convolutional neural networks", *PLoS ONE*, 12 (12): e0189259 (2017).
 73. Sabour, S., Frosst, N., and Hinton, G. E., "Dynamic routing between capsules", *ArXiv:1710.09829 [Cs]*, (2017).
 74. Internet: Kwiatkowski, S. E., "Capsule Networks: A Glossary", <http://www.aisummary.com/blog/capsule-networks-glossary/> (2018).
 75. Internet: Kothari, T., "Uncovering the Intuition Behind Capsule Networks and Inverse Graphics", <https://hackernoon.com/uncovering-the-intuition-behind-capsule-networks-and-inverse-graphics-part-i-7412d121798d> (2018).
 76. Internet: Hinton, G., "R/MachineLearning - AMA Geoffrey Hinton", https://www.reddit.com/r/MachineLearning/comments/2lmo0l/ama_geoffrey_hinton/clyj4jv (2019).

77. Internet: Bourdakos, N., "Understanding Capsule Networks — AI's Alluring New Architecture", <https://medium.freecodecamp.org/understanding-capsule-networks-ais-alluring-new-architecture-bdb228173ddc> (2018).
78. Riesenhuber, M. and Poggio, T., "Hierarchical models of object recognition in cortex", *Nature Neuroscience*, 2 (11): 1019–1025 (1999).
79. Serre, T., "Hierarchical models of the visual system", Encyclopedia of Computational Neuroscience, *Springer New York*, New York, NY, US, 1–12 (2013).
80. Hubel, D. H. and Wiesel, T. N., "Brain mechanisms of vision", *Scientific American*, 241 (3): 150–162 (1979).
81. Hinton, G., Sabour, S., and Frosst, N., "Matrix capsules with EM routing", *6th International Conference on Learning Representations*, Vancouver, BC, Canada, (2018).
82. Hinton, G. E., "A parallel computation that assigns canonical object-based frames of reference", *Seventh International Joint Conference on Artificial Intelligence*, Vancouver, BC, Canada, (1981).
83. Hinton, G. E., Krizhevsky, A., and Wang, S. D., "Transforming auto-encoders", *21st International Conference on Artificial Neural Networks*, Espoo, Finland, (2011).
84. Internet: Mahmood, Z., "Beginner's Guide to Capsule Networks", <https://www.kaggle.com/fizzbuzz/beginner-s-guide-to-capsule-networks> (2018).
85. Internet: Hui, J., "Understanding Dynamic Routing Between Capsules (Capsule Networks)", <https://jhui.github.io/2017/11/03/Dynamic-Routing-Between-Capsules/> (2018).
86. Internet: Tan, K., "Capsule Networks Explained", https://kndrck.co/posts/capsule_networks_explained/ (2018).
87. Lin, M., Chen, Q., and Yan, S., "Network in network", *ArXiv:1312.4400 [Cs]*, (2013).
88. Internet: Pechyonkin, M., "Understanding Hinton's Capsule Networks. Part III: Dynamic Routing Between Capsules", <https://medium.com/ai%C2%B3-theory-practice-business/understanding-hintons-capsule-networks-part-iii-dynamic-routing-between-capsules-349f6d30418> (2018).
89. Internet: Pechyonkin, M., "Understanding Hinton's Capsule Networks. Part IV: CapsNet Architecture", <https://medium.com/@pechyonkin/part-iv-capsnet-architecture-6a64422f7dce> (2018).

90. Internet: Géron, A., "Capsule Networks (CapsNets) – Tutorial", <https://www.youtube.com/watch?v=pPN8d0E3900&feature=youtu.be> (2019).
91. He, H. and Garcia, E. A., "Learning from imbalanced data", *IEEE Transactions On Knowledge And Data Engineering*, 21 (9): 1263–1284 (2009).
92. Internet: Chollet, F., "Building Powerful Image Classification Models Using Very Little Data", <https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html> (2018).
93. LeCun, Y., Huang, F. J., and Bottou, L., "Learning methods for generic object recognition with invariance to pose and lighting", *2004 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, Washington, DC, US, (2004).
94. Afshar, P., Mohammadi, A., and Plataniotis, K. N., "Brain tumor type classification via capsule networks", *25th IEEE International Conference on Image Processing (ICIP)*, Athens, Greece, (2018).
95. He, K., Zhang, X., Ren, S., and Sun, J., "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification", *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, (2015).
96. Kingma, D. P. and Ba, J., "Adam: A method for stochastic optimization", *3rd International Conference on Learning Representations*, San Diego, US, (2014).
97. Refaeilzadeh, P., Tang, L., and Liu, H., "Cross-validation", *Encyclopedia of Database Systems*, Springer, Boston, MA, US, 532–538 (2009).
98. Yosinski, J., Clune, J., Bengio, Y., and Lipson, H., "How transferable are features in deep neural networks?", *Advances in Neural Information Processing Systems 27*, Montreal, QC, Canada, (2014).
99. Internet: Karpathy, A., "Transfer Learning", <http://cs231n.github.io/transfer-learning/> (2018).
100. Krizhevsky, A., Sutskever, I., and Hinton, G. E., "ImageNet classification with deep convolutional neural networks", *Advances in Neural Information Processing Systems 26*, Stateline, NV, US, (2013).
101. Ioffe, S. and Szegedy, C., "Batch normalization: Accelerating deep network training by reducing internal covariate shift", *32nd International Conference on Machine Learning*, Lille, France, (2015).
102. Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition", *ArXiv:1409.1556 [Cs]*, (2014).
103. Torrey, L. and Shavlik, J., "Transfer learning", *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and*

- Techniques, *Information Science Reference - IGI Global*, Hershey, PA, US, 1–22 (2009).
104. Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., and Zhang, G., "Transfer learning using computational intelligence: A survey", *Knowledge-Based Systems*, 80: 14–23 (2015).
 105. Venkateswara, H., Chakraborty, S., and Panchanathan, S., "Deep-learning systems for domain adaptation in computer vision: Learning transferable feature representations", *IEEE Signal Processing Magazine*, 34 (6): 117–129 (2017).
 106. Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F., "Analysis of representations for domain adaptation", *Advances in Neural Information Processing Systems 19*, Vancouver, BC, Canada, (2006).
 107. Gong, B., Grauman, K., and Sha, F., "Reshaping visual datasets for domain adaptation", *Advances in Neural Information Processing Systems 26*, Las Vegas, NV, US, (2013).
 108. Csurka, G., "A comprehensive survey on domain adaptation for visual applications", *Domain Adaptation in Computer Vision Applications*, Springer International Publishing, Cham, 1–35 (2017).
 109. Patel, V. M., Gopalan, R., Li, R., and Chellappa, R., "Visual domain adaptation: A survey of recent advances", *IEEE Signal Processing Magazine*, 32 (3): 53–69 (2015).
 110. Zhang, J., Li, W., Ogunbona, P., and Xu, D., "Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective", *ACM Computational Survey*, 52 (1): 7:1–7:38 (2019).
 111. Pan, S. J. and Yang, Q., "A survey on transfer learning", *IEEE Transactions On Knowledge And Data Engineering*, 22 (10): 1345–1359 (2010).
 112. Ghifary, M., Balduzzi, D., Kleijn, W. B., and Zhang, M., "Scatter component analysis: A unified framework for domain adaptation and domain generalization", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 39 (7): 1414–1430 (2017).
 113. Weiss, K., Khoshgoftaar, T. M., and Wang, D., "A survey of transfer learning", *Journal Of Big Data*, 3 (1): 9 (2016).
 114. Mansour, Y., Mohri, M., and Rostamizadeh, A., "Domain adaptation with multiple sources", *Advances in Neural Information Processing Systems 21*, Granada, Spain, (2008).
 115. Yang, Q. and Pan, S. J., "Transfer learning with applications", *2013 PAKDD Tutorial*, Gold Coast, Australia, (2013).

116. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C., "A survey on deep transfer learning", *Artificial Neural Networks and Machine Learning – ICANN 2018*, Rhodes, Greece, (2018).
117. Wang, M. and Deng, W., "Deep visual domain adaptation: A survey", *Neurocomputing*, 312: 135–153 (2018).
118. Wilson, G. and Cook, D. J., "A survey of unsupervised deep domain adaptation", *ArXiv:1812.02849 [Cs, Stat]*, (2018).
119. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T., "Deep domain confusion: Maximizing for domain invariance", *ArXiv:1412.3474 [Cs]*, (2014).
120. Muandet, K., Balduzzi, D., and Schölkopf, B., "Domain generalization via invariant feature representation", *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, US, (2013).
121. Gholami, A., Subramanian, S., Shenoy, V., Himthani, N., Yue, X., Zhao, S., Jin, P., Biros, G., and Keutzer, K., "A novel domain adaptation framework for medical image segmentation", *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI): Brain Lesion (BrainLes) Workshop*, Shenzhen, China, (2019).
122. Kushibar, K., Valverde, S., González-Villà, S., Bernal, J., Cabezas, M., Oliver, A., and Lladó, X., "Supervised domain adaptation for automatic sub-cortical brain structure segmentation with minimal user interaction", *Scientific Reports*, 9 (1): 1–15 (2019).
123. Dou, Q., Ouyang, C., Chen, C., Chen, H., and Heng, P.-A., "Unsupervised cross-modality domain adaptation of Convnets for biomedical image segmentations with adversarial loss", *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, (2018).
124. Javanmardi, M. and Tasdizen, T., "Domain adaptation for biomedical image segmentation using adversarial training", *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, Washington, DC, (2018).
125. Venkataramani, R., Ravishankar, H., and Anamandra, S., "Towards continuous domain adaptation for medical imaging", *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, (2019).
126. Yao, L., Prosky, J., Covington, B., and Lyman, K., "A strong baseline for domain adaptation and generalization in medical imaging", *Medical Imaging with Deep Learning (MIDL 2019)*, London, UK, (2019).
127. Perone, C. S., Ballester, P., Barros, R. C., and Cohen-Adad, J., "Unsupervised domain adaptation for medical imaging segmentation with self-ensembling", *NeuroImage*, 194: 1–11 (2019).

128. Gholami, B., Sahu, P., Rudovic, O., Bousmalis, K., and Pavlovic, V., "Unsupervised multi-target domain adaptation: An information theoretic approach", *ArXiv:1810.11547 [Cs]*, (2018).
129. Alberta Laboratory Quality Enhancement Program (ALQEP), "Blood film staining effects", Educational Supplement, *The College of Physicians & Surgeons of Alberta (CPSA)*, AB, Canada, 5 (2004).
130. Kouw, W. M. and Loog, M., "An introduction to domain adaptation and transfer learning", *ArXiv:1812.11806 [Cs, Stat]*, (2018).
131. Labati, R. D., Piuri, V., and Scotti, F., "All-IDB: The acute lymphoblastic leukemia image database for image processing", *2011 18th IEEE International Conference on Image Processing*, Brussels, Belgium, (2011).
132. Elen, A. and Turan, M. K., "A new approach for fully automated segmentation of peripheral blood smears", *International Journal Of Advanced And Applied Sciences*, 5 (1): 81–93 (2018).
133. Sarrafzadeh, O., Dehnavi, A. M., Rabbani, H., and Talebi, A., "A simple and accurate method for white blood cells segmentation using K-means algorithm", *2015 IEEE Workshop on Signal Processing Systems (SiPS)*, Hangzhou, China, (2015).
134. Sarrafzadeh, O., Dehnavi, A. M., Rabbani, H., Ghane, N., and Talebi, A., "Circlet based framework for red blood cells segmentation and counting", *2015 IEEE Workshop on Signal Processing Systems (SiPS)*, Hangzhou, China, (2015).
135. Mohamed, M., Far, B., and Guaily, A., "An efficient technique for white blood cells nuclei automatic segmentation", *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Seoul, Korea, (2012).
136. Contis, L., Rollins-Raval, M., and Raval, J., "Experience with CellaVision DM96 for peripheral blood differentials in a large multi-center academic hospital system", *Journal Of Pathology Informatics*, 3 (1): 29 (2012).
137. Maaten, L. van der and Hinton, G., "Visualizing data using t-SNE", *Journal Of Machine Learning Research*, 9 (Nov): 2579–2605 (2008).
138. Wong, S. C., Gatt, A., Stamatescu, V., and McDonnell, M. D., "Understanding data augmentation for classification: When to warp?", *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Gold Coast, QLD, Australia, (2016).
139. Huang, S.-W., Lin, C.-T., Chen, S.-P., Wu, Y.-Y., Hsu, P.-H., and Lai, S.-H., "AugGAN: Cross domain adaptation with GAN-based data augmentation", *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, (2018).

140. Maya, S. and Ueno, K., "DADIL: Data augmentation for domain-invariant learning", *1st International Workshop on Utility-Driven Mining (UDM 2018)*, London, UK, (2018).
141. Volpi, R., Namkoong, H., Sener, O., Duchi, J. C., Murino, V., and Savarese, S., "Generalizing to unseen domains via adversarial data augmentation", *Advances in Neural Information Processing Systems 31*, Montreal, QC, Canada, (2018).
142. Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V., "AutoAugment: Learning augmentation policies from data", *ArXiv:1805.09501 [Cs, Stat]*, (2018).
143. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial networks", *ArXiv:1406.2661 [Cs, Stat]*, (2014).
144. Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H., "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification", *Neurocomputing*, 321: 321–331 (2018).
145. Nie, D., Trullo, R., Lian, J., Petitjean, C., Ruan, S., Wang, Q., and Shen, D., "Medical image synthesis with context-aware generative adversarial networks", *Medical Image Computing and Computer Assisted Intervention – MICCAI 2017*, Quebec City, QC, Canada, (2017).
146. Volpi, R., Morerio, P., Savarese, S., and Murino, V., "Adversarial feature augmentation for unsupervised domain adaptation", *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, US, (2018).
147. Huang, H., Yu, P. S., and Wang, C., "An introduction to image synthesis with generative adversarial nets", *ArXiv:1803.04469 [Cs]*, (2018).
148. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C., "Improved training of Wasserstein GANs", *Advances in Neural Information Processing Systems 30*, Long Beach, CA, US, (2017).
149. Arjovsky, M., Chintala, S., and Bottou, L., "Wasserstein GAN", *ArXiv:1701.07875 [Cs, Stat]*, (2017).
150. Radford, A., Metz, L., and Chintala, S., "Unsupervised representation learning with deep convolutional generative adversarial networks", *6th International Conference on Learning Representations*, Vancouver, BC, Canada, (2018).
151. Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S., "CNN features off-the-shelf: An astounding baseline for recognition", *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus, OH, USA, (2014).

152. Bengio, Y., "Practical recommendations for gradient-based training of deep architectures", *Neural Networks: Tricks of the Trade*, Springer, Berlin, Heidelberg, 437–478 (2012).
153. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., "Dropout: A simple way to prevent neural networks from overfitting", *Journal Of Machine Learning Research*, 15 (Jun): 1929–1958 (2014).
154. Shao, L., Zhu, F., and Li, X., "Transfer learning for visual categorization: A survey", *IEEE Transactions On Neural Networks And Learning Systems*, 26 (5): 1019–1034 (2015).
155. Internet: Özmen, B., "DeepAugment", <https://github.com/barisozmen/deepaugment> (2019).
156. Sun, S., Shi, H., and Wu, Y., "A survey of multi-source domain adaptation", *Information Fusion*, 24: 84–92 (2015).
157. Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Wortman, J., "Learning bounds for domain adaptation", *Advances in Neural Information Processing Systems 20*, Vancouver, BC, Canada, (2007).
158. Zhang, L., "Transfer adaptation learning: A decade survey", *ArXiv:1903.04687 [Cs]*, (2019).
159. Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y., "Random erasing data augmentation", *ArXiv:1708.04896 [Cs]*, (2017).
160. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A., "Adversarial examples are not bugs, they are features", *ArXiv:1905.02175 [Cs, Stat]*, (2019).
161. Hendrycks, D., Lee, K., and Mazeika, M., "Using pre-training can improve model robustness and uncertainty", *ArXiv:1901.09960 [Cs, Stat]*, (2019).
162. Yu, H., Hu, M., and Chen, S., "Multi-target unsupervised domain adaptation without exactly shared categories", *ArXiv:1809.00852 [Cs, Stat]*, (2018).
163. Chen, Z., Zhuang, J., Liang, X., and Lin, L., "Blending-target domain adaptation by adversarial meta-adaptation networks", *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, Long Beach, CA, US, (2019).
164. Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., and Liang, J., "Convolutional neural networks for medical image analysis: Full training or fine tuning?", *IEEE Transactions On Medical Imaging*, 35 (5): 1299–1312 (2016).

ÖZGEÇMİŞ

Yusuf Yargı BAYDİLLİ 1986 yılında İstanbul'da doğdu; ilk ve orta öğrenimini aynı şehirde tamamladı. 2009 yılında Abant İzzet Baysal Üniversitesi Fizik Bölümünden mezun oldu ve 2010 yılında Hakkâri Üniversitesi Bilgisayar Mühendisliği Anabilim Dalına Araştırma Görevlisi olarak atandı. 2011 yılında ise Yüksek Lisans ve Doktora eğitimini gerçekleştirmek üzere Karabük Üniversitesi Bilgisayar Mühendisliği Anabilim Dalına 35. Madde kapsamında görevlendirildi. Halen aynı yerde eğitim ve meslek hayatına devam etmektedir.

ADRES BİLGİLERİ

Adres : Karabük Üniversitesi
Bilgisayar Mühendisliği Bölümü
Balıklarkayası Mevkii / KARABÜK
E-posta : yusufbaydilli@karabuk.edu.tr