



**DERİN ÖĞRENME KULLANARAK
TIBBİ GÖRÜNTÜ BÖLÜTLEME**

Rüya KAYMAK

**Yüksek Lisans Tezi
Mekatronik Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Ayşegül UÇAR
HAZİRAN-2019**

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME KULLANARAK TIBBİ GÖRÜNTÜ BÖLÜTLEME

YÜKSEK LİSANS TEZİ

Rüya KAYMAK

(132134102)

Tezin Enstitüye Verildiği Tarih : 14 Mayıs 2019

Tezin Savunulduğu Tarih : 14 Haziran 2019

Tez Danışmanı : Doç. Dr. Ayşegül UÇAR (F.Ü)
Diğer Jüri Üyeleri : Dr. Öğr. Üyesi Dursun ÖZTÜRK (B.Ü)
Dr. Öğr. Üyesi Hakan ÇELİK (F.Ü)



HAZİRAN-2019

ÖNSÖZ

Öncelikle lisans ve yüksek lisans eğitimimde danışmanım olan ve beni sürekli destekleyen değerli hocam Sayın Doç. Dr. Ayşegül UÇAR'a, gerçekleştirdiğim çalışmaların tamamlanmasında sağlamış olduğu katkıların yanı sıra göstermiş olduğu ilgi ve hoşgörüsünden dolayı çok teşekkür ederim.

Sürekli yanımda olan eşim Arş. Gör. Çağrı KAYMAK'a da çalışmalarım süresince göstermiş olduğu manevi desteğinden dolayı çok teşekkür ederim.

Son olarak da, hayatım boyunca her daim yanımda olan, sevgi ve desteklerini hiçbir zaman esirgemeyen aileme teşekkürü borç bilir, saygılarımı sunarım.

Rüya KAYMAK
ELAZIĞ – 2019

İÇİNDEKİLER

Sayfa No

ÖNSÖZ	II
İÇİNDEKİLER	III
ÖZET	V
SUMMARY	VI
ŞEKİLLER LİSTESİ	VII
TABLolar LİSTESİ	IX
KISALTMALAR	X
1. GİRİŞ	1
1.1. Tezin Organizasyonu	7
2. DERİN ÖĞRENME	8
2.1. Makine Öğrenmesi	9
2.1.1. Gözetimli Öğrenme	9
2.1.2. Gözetimsiz Öğrenme	10
2.2. Yapay Sinir Ağları	11
2.2.1. Nöron	11
2.2.1.1. Giriş	13
2.2.1.2. Ağırlık	13
2.2.1.3. Eşik	13
2.2.1.4. Birleştirme Fonksiyonu	14
2.2.1.5. Aktivasyon Fonksiyonu	14
2.2.1.5.1. Sigmoid Aktivasyon Fonksiyonu	14
2.2.1.5.2. tanh Aktivasyon Fonksiyonu	16
2.2.1.5.3. Düzeltilmiş Doğrusal Birim Aktivasyon Fonksiyonu	16
2.2.1.6. Çıkış	16
2.2.2. İleri Beslemeli Sinir Ağı	16
2.2.3. Geriye Yayılım Algoritması	19
2.3. Derin Öğrenme Yazılım Geliştirme Araçları ve Kütüphaneleri	20
2.3.1. Caffe	21
2.3.2. Torch	22
2.3.3. Theano	22
2.3.4. TensorFlow	22
2.3.5. Keras	23
2.3.6. DIGITS	23
3. KONVOLÜSYONEL SİNİR AĞLARI	26
3.1. Mimari	26
3.1.1. Konvolüsyon Katmanı	27
3.1.2. Öznitelik Seçici Katman	31
3.1.3. Düzeltilmiş Doğrusal Birim Katmanı	33
3.1.4. Düzenleştirme	34
3.1.5. Tam Bağlı Katman	35
3.1.6. Sınıflandırıcı	35
3.2. Eğitim	36
3.2.1. İleri Hesaplama	36
3.2.2. Kayıp Eniyileme	37

3.2.3.	Geriye Yayılım.....	37
3.3.	Hiperparametreler	38
3.3.1.	Öğrenme Oranı.....	39
3.3.2.	Küçük Yığın Boyutu	39
3.3.3.	Ağırlık Başlatma	39
3.3.4.	Düzenleştirme Değeri	40
3.3.5.	Momentum Katsayısı	40
3.3.6.	Eğitim Devir Sayısı	40
3.4.	Bilinen Bazı KSA Mimarileri	41
4.	ANLAMSAL GÖRÜNTÜ BÖLÜTLEME VE TAM KONVOLÜSYONEL AĞLAR	44
4.1.	Tıbbi Görüntü Bölütleme	46
4.2.	Konvolüsyonel Sinir Ağından Tam Konvolüsyonel Ağa Geçiş	49
5.	DENEYSEL ÇALIŞMALAR.....	57
5.1.	Veri Seti	57
5.2.	Performans Değerlendirme Metrikleri	58
5.3.	Modellerin Eğitimleri ve Geçerlilikleri.....	61
6.	SONUÇLAR VE ÖNERİLER	76
	KAYNAKLAR	78
	ÖZGEÇMİŞ.....	84

ÖZET

DERİN ÖĞRENME KULLANARAK TIBBİ GÖRÜNTÜ BÖLÜTLEME

Bu tezde, ilk olarak son yıllarda makine öğrenmesi alanında çok fazla ilgi duyulan derin öğrenme yaklaşımı hakkında detaylı bilgilere yer verilmiştir. Daha sonra, en tehlikeli cilt kanseri türü olan melanomun tespitine yardımcı olabilmek adına otomatik bir cilt lezyonu bölütleme uygulaması gerçekleştirilmiştir. Bu tıbbi görüntü bölütleme uygulaması, derin öğrenme tabanlı olan Konvolüsyonel Sinir Ağı (KSA) mimarilerinde değişiklik yapılarak elde edilen Tam Konvolüsyonel Ağ (TKA) mimarileriyle gerçekleştirilmiştir.

Uygulama için gerçekleştirilen deneysel çalışmalarda veri seti olarak ISIC 2017 kullanılmış olup FCN-AlexNet, FCN-8s, FCN-16s ve FCN-32s isimli dört farklı TKA mimarisinden yararlanılmıştır. Bu mimariler ve veri seti dikkate alındığında, bu çalışma bilimsel yazında ilk kez gerçekleştirilmiştir. Deneysel çalışmalar için öncelikle TKA'lar ayrı ayrı eğitilmiş ve eğitilen bu ağ modellerinin geçerlilik veri seti üzerindeki doğrulukları ve Dice katsayıları karşılaştırılmıştır. Ayrıca, lezyon bölütleme çıkarımları görselleştirilerek kullanılan TKA mimarilerinin lezyonları ne kadar keskinlikte bölütleyebildiği gözlemlenmiştir.

Elde edilen deneysel sonuçlar, cilt lezyonu bölütleme uygulamalarında TKA'ların elverişli olduğunu göstermiştir. Bununla beraber, deneysel sonuçların bilimsel yazına ve tıbbi görüntü bölütleme ile ilgili çalışmalar yapan araştırmacılara katkı sağlayacağı düşünülmektedir.

Anahtar Kelimeler: Derin öğrenme, Konvolüsyonel Sinir Ağı, Tam Konvolüsyonel Ağ, Tıbbi görüntü bölütleme.

SUMMARY

MEDICAL IMAGE SEGMENTATION USING DEEP LEARNING

In this thesis, first of all, detailed information about the deep learning approach which is attracted much attention in the field of machine learning in recent years is given. Then, an automatic application about skin lesion segmentation is performed to assist in the detection of melanoma, the most dangerous type of skin cancer. This medical image segmentation application is implemented by using Fully Convolutional Network (FCN) architectures obtained by modifying deep learning-based Convolutional Neural Network (CNN) architectures.

ISIC 2017 is used as the dataset and four different FCN architectures named FCN-AlexNet, FCN-8s, FCN-16s and FCN-32s are used in the experimental studies performed for the application. Considering these architectures and dataset, this study is carried out for the first time in the literature. For the experimental studies, FCNs are first trained separately and accuracies on the validation dataset and Dice coefficients of these trained network models are compared. Besides, lesion segmentation inferences are visualized to take account of how precisely FCN architectures can segment lesions.

The experimental results obtained show that FCNs are suitable for skin lesion segmentation. In addition, it is thought that the experimental results will contribute to the scientific literature and the researchers who are working on medical image segmentation.

Keywords: Deep learning, Convolutional Neural Network, Fully Convolutional Network, Medical image segmentation.

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 1.1.	“Deep Learning” kelimesinin Google arama motorunda son 5 yılda dünya genelindeki sorgu trendi	4
Şekil 2.1.	Temel derin öğrenme mimarileri	8
Şekil 2.2.	Gözetimli öğrenme modelinin iş akışı	10
Şekil 2.3.	Gözetimsiz öğrenme modelinin iş akışı.....	11
Şekil 2.4.	Biyolojik nöronun yapısı	12
Şekil 2.5.	Yapay nöronun yapısı	13
Şekil 2.6.	Sigmoid aktivasyon fonksiyonunun karakteristiği	15
Şekil 2.7.	Tanh aktivasyon fonksiyonunun karakteristiği.....	15
Şekil 2.8.	DDB aktivasyon fonksiyonunun karakteristiği	16
Şekil 2.9.	3 katmanlı ileri beslemeli bir YSA modeli	17
Şekil 2.10.	İleri beslemeli derin bir YSA modeli.....	18
Şekil 2.11.	DIGITS ana konsolu	24
Şekil 3.1.	Basit bir KSA mimarisi örneği	27
Şekil 3.2.	Tek bir konvolüsyon katmanının gösterimi	28
Şekil 3.3.	Görüntü matrisi ve filtre örneği	29
Şekil 3.4.	Görüntü matrisi ve öznitelik haritasındaki ilk öznitelik	30
Şekil 3.5.	Görüntü matrisi ve öznitelik haritasındaki ilk iki öznitelik	30
Şekil 3.6.	Görüntü matrisi ve öznitelik haritasının son durumu	30
Şekil 3.7.	Örnek bir alt örnekleme işlemi	32
Şekil 3.8.	2×2 boyutundaki filtre ve 2 kaydırma ile maksimum öznitelik seçme işlemi..	32
Şekil 3.9.	DDB işlem örneği.....	33
Şekil 3.10.	Bir ağdaki seyreltme işlem örneği: (a) Standart bir sinir ağı, (b) Seyreltme işleminden sonra sinir ağı	34
Şekil 4.1.	(a) Sınıflandırma, (b) Bulma, (c) Bölütleme	44
Şekil 4.2.	Örnek bir bölütlenecek görüntü-etiket görüntüsü çifti: (a) Bölütlenecek görüntü, (b) Etiket görüntüsü.....	45
Şekil 4.3.	TKA ile piksel piksel tahmin işlemi	46
Şekil 4.4.	Tıbbi görüntü örnekleri: (a) BT akciğer tarama, (b) MR beyin tarama, (c) X-ışını göğüs tarama, (d) Dermoskopi görüntüsü.....	47

Şekil 4.5.	Örnek bir anlamsal tıbbi görüntü bölütleme işlemi	48
Şekil 4.6.	KSA'nın tam bağlı katmanlarını konvolüsyon katmanlarına dönüştürme	49
Şekil 4.7.	AlexNet'in ilk tam bağlı katmanının girişi, ağırlıkları ve aktivasyonları.....	51
Şekil 4.8.	AlexNet'in ilk konvolüsyonelleştirilen katmanının girişi, ağırlıkları ve aktivasyonları.....	52
Şekil 4.9.	FCN-32s, FCN-16s ve FCN-8s mimarilerinin görselleştirilmesi	56
Şekil 5.1.	ISIC 2017 veri setinden örnekler: (a) Bölütlenecek görüntüler, (b) Anlamsal etiketli kesin referans görüntüler	58
Şekil 5.2.	Bölütleme üzerine istatistiksel alanların tanımı.....	59
Şekil 5.3.	Ağırlıkların rastgele başlatılmasıyla FCN-AlexNet'in eğitimi sırasında eğitim-geçerlilik kaybı, geçerlilik doğruluğu ve Dice katsayısı	62
Şekil 5.4.	Önceden eğitilmiş modelin kullanılmasıyla FCN-AlexNet'in eğitimi sırasında eğitim-geçerlilik kaybı, geçerlilik doğruluğu ve Dice katsayısı.....	64
Şekil 5.5.	Önceden eğitilmiş modeli kullanarak eğitilen FCN-AlexNet ile örnek bir test görüntüsü için lezyon bölütleme işlemi-1: (a) Test görüntüsü, (b) Kesin referans görüntüsü, (c) Bölütleme çıkarımı	65
Şekil 5.6.	Önceden eğitilmiş modeli kullanarak eğitilen FCN-AlexNet ile örnek bir test görüntüsü için lezyon bölütleme işlemi-2: (a) Test görüntüsü, (b) Kesin referans görüntüsü, (c) Bölütleme çıkarımı	66
Şekil 5.7.	Önceden eğitilmiş modeli kullanarak eğitilen FCN-AlexNet ile örnek bir test görüntüsü için lezyon bölütleme işlemi-3: (a) Test görüntüsü, (b) Kesin referans görüntüsü, (c) Bölütleme çıkarımı	67
Şekil 5.8.	Önceden eğitilmiş modelin kullanılmasıyla FCN-8s'nin eğitimi sırasında eğitim-geçerlilik kaybı, geçerlilik doğruluğu ve Dice katsayısı	68
Şekil 5.9.	Önceden eğitilmiş modeli kullanarak eğitilen FCN-8s ile örnek bir test görüntüsü için lezyon bölütleme işlemi-1: (a) Test görüntüsü, (b) Kesin referans görüntüsü, (c) Bölütleme çıkarımı	69
Şekil 5.10.	Önceden eğitilmiş modeli kullanarak eğitilen FCN-8s ile örnek bir test görüntüsü için lezyon bölütleme işlemi-2: (a) Test görüntüsü, (b) Kesin referans görüntüsü, (c) Bölütleme çıkarımı	70
Şekil 5.11.	Önceden eğitilmiş modeli kullanarak eğitilen FCN-8s ile örnek bir test görüntüsü için lezyon bölütleme işlemi-3: (a) Test görüntüsü, (b) Kesin referans görüntüsü, (c) Bölütleme çıkarımı	71
Şekil 5.12.	Önceden eğitilmiş modelin kullanılmasıyla FCN-16s'nin eğitimi sırasında eğitim-geçerlilik kaybı, geçerlilik doğruluğu ve Dice katsayısı	72
Şekil 5.13.	Önceden eğitilmiş modelin kullanılmasıyla FCN-32s'nin eğitimi sırasında eğitim-geçerlilik kaybı, geçerlilik doğruluğu ve Dice katsayısı.....	72
Şekil 5.14.	Bazı örnek test görüntüleri için lezyon bölütleme çıkarımlarının kullanılan TKA mimarilerine göre karşılaştırılması.....	74
Şekil 5.15.	Modellerin eğitim süreleri	75

TABLÖLAR LİSTESİ

Sayfa No

Tablo 2.1. Biyolojik sinir ağı yapısının yapay sinir ağı karşılıkları.....	12
Tablo 3.1. AlexNet'in mimarisi	41
Tablo 3.2. VGG-16'nın mimarisi.....	42
Tablo 4.1. FCN-AlexNet mimarisi	54
Tablo 5.1. Hata matrisi.....	60
Tablo 5.2. Ağ parametrelerinin değerleri.....	61

KISALTMALAR

ROI	: İlgilenilen bölge – Region of interest
YSA	: Yapay Sinir Ağı
ILSVRC	: ImageNet Büyük Ölçekli Görsel Tanıma Yarışması – ImageNet Large Scale Visual Recognition Challenge
CPU	: Merkezi İşlem Birimi – Central Processing Unit
GPU	: Grafik İşlem Birimi – Graphics Processing Unit
KSA	: Konvolüsyonel Sinir Ağı
TKA	: Tam Konvolüsyonel Ağ
MR	: Manyetik Rezonans
BT	: Bilgisayarlı Tomografi
DVM	: Destek Vektör Makinesi
PROMISE	: Prostat MR Görüntü Bölütleme – Prostate MR Image Segmentation
BRATS	: Beyin Tümörü Bölütleme – Brain Tumor Segmentation
3B	: 3 Boyutlu
CRF	: Koşulsal Rastgele Alan – Conditional Random Field
ISIC	: Uluslararası Cilt Görüntüleme İşbirliği – International Skin Imaging Collaboration
DDB	: Düzeltilmiş Doğrusal Birim
k-EK	: k-En yakın Komşuluk – k-Nearest Neighbor
Tanh	: Tanjant hiperbolik
CUDA	: Compute Unified Device Architecture – Birleşik Hesaplama Aygıt Mimarisi
BVLC	: Berkeley Görü ve Öğrenme Merkezi – Berkeley Vision and Learning Center
API	: Uygulama Programlama Arayüzü – Application Programming Interface
cuDNN	: CUDA Derin Sinir Ağı kütüphanesi – CUDA Deep Neural Network library
DIGITS	: NVIDIA Derin Öğrenme GPU Eğitim Sistemi – NVIDIA Deep Learning GPU Training System
RGB	: Red, Green, Blue – Kırmızı, Yeşil, Mavi
LRN	: Local Response Normalization – Yerel Tepki Normalizasyonu
PET	: Pozitron Emisyon Tomografi – Positron Emission Tomography
ISBI	: Biyomedikal Görüntüleme Uluslararası Sempozyum – International Symposium on Biomedical Imaging
YP	: Yanlış Pozitif
YN	: Yanlış Negatif
DP	: Doğru Pozitif
DN	: Doğru Negatif
IoU	: Kesişimin birleşime oranı – Intersection over Union

1. GİRİŞ

Gelişen teknolojiyle modern kamera sistemleri, cep telefonundan gözetim sistemlerine ve otonom araçlardan sağlık sektörüne kadar birçok yere yerleştirilerek düşük maliyette çok kaliteli görüntüler elde edebilir. Bu durum, bu görüntüleri yorumlayıp anlayabilen sistemlere olan talebi arttırmaktadır.

Görüntülerin yorumlanmasına yıllardır çeşitli şekillerde yaklaşılmıştır. Ancak nesneleri belirlemek ve önemini değerlendirmek amacıyla görüntüleri incelemeyi içeren süreç aynıdır. Görsel bilgidен öğrenme problemleri genellikle görüntü sınıflandırma, nesne yer tespiti ve bulma ve anlamsal bölütleme olarak sınıflandırılır [1].

Anlamsal görüntü bölütleme, tıbbi görüntü işleme, otonom sürüş, robotik navigasyon, örüntü tanıma, makine öğrenmesi ve sahne anlayışı başta olmak üzere birçok kritik uygulama için doğal dünyayı haritalama ve sınıflandırma işlemidir. Görüntü sınıflandırma için piksel seviyesinde bir etiketleme olan anlamsal bölütleme, sahne anlayışı için önemli bir tekniktir. Çünkü her bir piksel verilen bir anlamsal sınıfa ait olarak etiketlenmektedir. Temel olarak bölütleme sonuçları, nesnenin betimlenmesi ve sınıflandırılması gibi önemli görevleri etkilediğinden tüm görüntü analizi işlemlerini etkiler. Bu yüzden görüntü bölütleme, herhangi bir görüntüde ilgilenilen bölgelerin kısaca ROI'nin (Region of interest) sınıflandırılmasını, nitelendirilmesini ve görselleştirilmesini kolaylaştırmak için önemli bir süreçtir.

Tıbbi görüntülerde organların veya diğer yapıların bölütlendirilmesi, hacim ve şekil ile ilgili klinik parametrelerin nicel analizine izin verir. Tıbbi görüntü bölütleme, tıbbi görüntü analizi ve tanı gibi birçok uygulamaya sahip sayısal görüntü işlemede görüntü bölütlemenin bir alt alanıdır. Tıbbi görüntü bölütleme, radyologların insan vücudu yapılarının anatomisini görselleştirmelerine ve incelemelerine, biyolojik süreçleri benzetim yapmalarına, patolojilerin yerini saptamalarına, hastalıkların ilerlemelerini izlemelerine yardımcı olan bilgiler sağlamada büyük önem taşımaktadır. Tipik bir tıbbi görüntü; lezyon veya tümör bölgesi ile arka plandan oluşabilir.

İnsan vücudundaki anatomik yapıların değişkenliği ve karmaşıklığı, tıbbi görüntü bölütlemenin zor bir problem olarak tanımlanmasına neden olmuştur [2]. Tıbbi görüntünün radyolog tarafından el ile bölütlenmesi yalnızca yorucu ve zaman alıcı bir süreç değil, aynı zamanda çok fazla dikkat ve tecrübe gerektiren bir iştir. Ayrıca, görüntüleme tekniğinden kaynaklanan gürültü ve görüntüdeki kontrastın düşük olması radyoloğun görsel analizini

zorlaştırabilmektedir. Böylece, gecikmeli veya hatalı tanı hastaya zarar vereceğinden tıbbi görüntü bölütlemenin mümkün olduğunca az kullanıcı etkileşimi gerektiren otomatik algoritmalar kullanarak yapılmasını zorunlu hale getirmektedir.

Tıbbi görüntü bölütlemenin bilgisayarlı otomasyonu, tıbbi görüntüleme uygulamalarında önemli bir yere sahiptir. Tıbbi görüntüleri tarama ve bilgisayara yükleme işlemleri mümkün hale gelir gelmez, araştırmacılar otomatik analiz için sistemler geliştirmiştir. Sistemin işlevselliği büyük ölçüde bölütleme doğruluğuna bağlıdır.

Son yıllarda makine öğrenmesi algoritmalarının bilgisayarla görme görevlerindeki muazzam başarısı, tıbbi kayıtların ve tanısal görüntülemenin büyük bir şekilde arttırıldığı zamanla kesişmektedir. Makine öğrenmesi, piyasada bulunan birçok başarılı tıbbi görüntü analiz sisteminin temelini oluşturmaktadır. Makine öğrenmesi algoritmaları, ilacın keşfedilmesinden klinik karar vermeye kadar tüm tıp alanlarına derinlemesine yatırım yapılma potansiyeline sahiptir.

Makine öğrenmesi alanlarından olan Yapay Sinir Ağları (YSA'lar) yeni bir yaklaşım olmayıp, geçmişi yaklaşık 60 yıl öncesine dayanmaktadır. 1990'lı yıllara kadar çeşitli alanlarda kullanılan YSA'lar, doğrusal olmayan sistemler üzerinde tatmin edici başarılar sağlayamamıştır. Bu yüzden belli bir süre YSA'lar üzerine fazla çalışma yapılmamıştır. 2006 yılında Hinton ve arkadaşları [3], YSA'ları konuşma tanıma problemlerinde kullanmışlar ve başarılı sonuçlar elde etmişlerdir. Böylece YSA'lar, tekrar bilim dünyasında gündeme gelmiştir. Artık araştırmacılar YSA'ların çoğu alandaki problemlere çözüm olacağını düşünmüş, ancak kısa süre sonra çok katmanlı YSA'ların eğitimindeki başarısızlık gibi çeşitli nedenlerle bunun yanlış bir düşünce olduğunu anlamışlardır. Daha sonra ImageNet Büyük Ölçekli Görsel Tanıma Yarışması, kısaca ILSVRC (ImageNet Large Scale Visual Recognition Challenge) gibi görüntü tanıma yarışmalarında YSA tabanlı sistemler liderliği ele geçirmiş ve diğer sistemlere büyük farklar atarak her sene birinci olmaya başlamışlardır. Zaman ilerledikçe, özellikle internet sayesinde çok büyük miktarlardaki veri sayısal ortamda üretilmeye ve saklanır hale gelmeye başlamıştır. Büyük miktarlardaki bu verilerin işlenmesinde bilgisayarlardaki Merkezi İşlem Birimleri kısaca CPU'lar (Central Processing Units) yavaş kalmıştır. Grafik İşlem Birimi kısaca GPU (Graphics Processing Unit) teknolojisindeki gelişmelerle beraber grafik işlemcinin paralel hesaplama mimarisinden yararlanılarak hesapsal işlemler çok daha hızlı yapılabilmektedir. İşlem gücündeki bu artışla birlikte uygulamalarda daha “derin” sinir ağlarının kullanılması yaygınlaşmıştır. Bu sayede

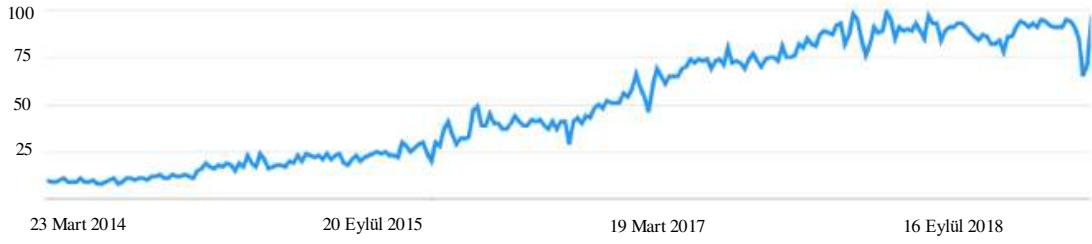
makine öğrenmesinde yeni bir yaklaşım olan “Derin Öğrenme (Deep Learning)” kavramı ortaya çıkmıştır.

Derin öğrenme, gizli katmanların sayısı arttırılmış derin mimariye sahip YSA’lardan oluşan yöntemler bütünüdür. Bu derin mimarinin her katmanında, probleme ait öznitelikler öğrenilir ve öğrenilen bu öznitelik bir üst katmana bir giriş oluşturur. Böylelikle en alt katmandan en üst katmana doğru, özniteliklerin en basitinden en karmaşık olanına kadar öğrenildiği bir yapı oluşturulur. Bu yapıyı anlamak için insan beynindeki görme sistemi incelenebilir. Sinirler aracılığıyla gözlere gelen sinyaller çok katmanlı hiyerarşik bir yapıda değerlendirilir. Gözlerden sonra sinyalin uğradığı ilk katmanda kenar ve köşe gibi görüntünün yerel ve temel öznitelikleri tespit edilir. Sonraki katmanda, bu öznitelikler bir araya getirilerek ağız, burun vb. ayrıntılar ve daha sonraki katmanlarda da sırasıyla yüz, kişi ve nesnelerin yerleşimi gibi görüntünün geneline ait olan öznitelikleri tespit edilebilir. Bilgisayarla görme uygulamalarında hem öznitelik çıkarımı hem de sınıflandırma yetisini bir arada bulunduran Konvolüsyonel Sinir Ağları (KSA’lar) yaklaşımı bu şekilde çalışmaktadır.

Genom ile ilgili verilerden sonra görüntüleme, sağlık sektöründe en hızlı büyüyen veri sektörüdür. Sağlık kuruluşları, son derece değerli sinyaller ve bilgiler içeren aşırı miktarda veri üretir. Bu veriler, geleneksel analiz yöntemlerinin işleyebileceği boyutu aşar. Sağlık hizmetlerinde derin öğrenmenin kullanımı henüz başlangıç seviyesinde olmasına rağmen akademik alanda birçok güçlü girişim mevcuttur ve çok sayıda büyük şirket makine öğrenmesine dayalı sağlık projeleri yürütmektedir.

Derin öğrenme, son yıllarda geliştirilen yapay zekâ uygulamalarının başarısını oldukça yüksek seviyelere çıkarmıştır. Derin öğrenme, bilgisayarla görme, ses tanıma, doğal dil işleme ve gömülü sistemler gibi birçok alanda kullanılmaktadır. Son yıllarda devasa boyutlarda veri setleri kullanılarak gerçekleştirilen ILSVRC’de yarışmacılar, KSA yaklaşımlarına yönelmişler ve çok büyük başarılar elde etmişlerdir [3]. Google [4], Facebook [5], Microsoft [6] ve Baidu [7] gibi şirketler derin öğrenmedeki ilerleyişi fark edip büyük yatırımlarla bu konu üzerinde çalışmalar yürütmektedir.

Derin öğrenme kelimesinin ingilizce olarak Google arama motorunda son 5 yılda dünya genelinde ne kadar sorgulandığını gösteren grafik Şekil 1.1’de verilmiştir.



Şekil 1.1. “Deep Learning” kelimesinin Google arama motorunda son 5 yılda dünya genelindeki sorgu trendi [8]

KSA’ların geliştirilmesi, yüksek miktardaki etiketli verilere dayanmaktadır. Genel bir şekilde ifade etmek gerekirse KSA’lar, milyonlarca parametrenin öğrenilmesini gerektiren uçtan uca eğitime (ek açıklamalı ham verilerden sınıf etiketlerine) dayanmakta olup görsel tanımlayıcılara dayanan yöntemlerden daha başarılıdır.

Anlamsal bölütlemeye yapılan çalışmaların bir kısmında görüntü sınıflandırma için tasarlanmış KSA mimarileri doğrudan benimsenmeye çalışılmıştır. Ancak, standart KSA mimarileri piksel piksel sınıflandırmanın yapıldığı anlamsal bölütlemeye uygun olmadığı için sonuçların çok da tatmin edici olmadığı görülmüştür. Bunun sebebi uzaysal konumun kaybedilmesidir. Bir taraftan, tekrarlanan konvolüsyon ve öznetelik seçici katmanlar öznetelik haritalarının uzaysal çözünürlüğünü düşürürken diğer taraftan da tam bağlı katmanlar çıkış üretmek için uzaysal bilgiyi tamamen atarak sınıf olasılık değerleri üretirler.

Yakın tarihte, anlamsal bilgiyi tam bir piksel konumuna getirmeyi mümkün kılan ağ mimarileri ortaya çıkmıştır [9, 10]. İlk olarak Long, Shelhamer ve Darrell [9], sınıflandırma için oluşturulan önceden eğitilmiş VGG-16 [11] KSA mimarisini Tam Konvolüsyonel Ağlar’a (TKA’lara) dönüştürmüşlerdir. Bunun için KSA mimarisinin tüm tam bağlı katmanları konvolüsyon katmanlar ile değiştirmiş ve sekme bağlantılarıyla birlikte orijinal uzaysal çözünürlüğü geri getiren ters konvolüsyon katmanları eklemişlerdir. Böylece, FCN-8s, FCN-16s ve FCN-32s olarak adlandırılan üç farklı TKA sürümü elde edilmiş olup çıkıştaki uzaysal çözünürlük giriş görüntüsünün uzaysal çözünürlüğüne getirilerek her bir piksel için sınıf olasılık değerlerinin üretilmesi sağlanmıştır.

KSA mimarilerinin tıbbi görüntü bölütlemeye en bilinenlerinden biri Ronneberger, Fischer ve Brox tarafından yayınlanan U-Net’tir [12]. U-Net’teki iki ana mimari yenilik, eşit miktarda üst örnekleme ve alt örnekleme katmanlarının birleşimidir.

Bilimsel yazında, derin öğrenme kullanarak dermoskopi, Manyetik Rezonans (MR) ve Bilgisayarlı Tomografi (BT) gibi görüntüleme sistemleriyle elde edilen tıbbi görüntüler üzerinde gerçekleştirilen birçok başarılı çalışma bulunmaktadır. Yapılan çalışmalardan bazılarına ilişkin ayrıntılara aşağıda yer verilmiştir:

Yu ve arkadaşları [13], cilt kanseri türü olan melanomu, melanom dışı lezyonlardan ayırmak için tam konvolüsyonel çok derin bir kalıntı ağını dermoskopik görüntüler üzerinde uygulamışlardır.

Codella ve arkadaşları [14], melanomu tanımak için KSA, seyrek kodlama ve Destek Vektör Makinesi'ni (DVM'yi) entegre eden bir karma yaklaşım önermişlerdir.

Codella ve arkadaşları bir diğer çalışmada [15], cilt lezyonu bölütlemesi ve sınıflandırması için derin öğrenme yaklaşımlarındaki son gelişmeleri birleştiren bir sistem önermişlerdir.

Kawahara, Bentaieb ve Hamarneh [16], melanom tanıma amacıyla çok ölçekli öznetelikler çıkarmak için TKA kullanmıştır.

Jin ve arkadaşları [17], karaciğer BT görüntüleri üzerinde karaciğer bölütlenmesi için özel bir TKA yapısı önermişlerdir.

Brosch ve arkadaşları [18], hem konvolüsyonel kodlayıcı bir ağ hem de U-Net mimarisine benzer bir ters konvolüsyonel bir ağ mimarisini beyin MR görüntülerindeki ak madde lezyonlarını bölütlemek için kullanmışlardır. Ayrıca, 3B konvolüsyonlar ile birlikte ilk konvolüsyon ve son ters konvolüsyon katmanları arasında tek bir sekme bağlantısı kullanmışlardır.

Beyin MR görüntüleme bölütlemesinin iyi bir derlemesi Akkuş ve arkadaşları [19] tarafından yapılmış ve bölütlemede kullanılan çeşitli KSA mimarileriyle değerlendirme metriklerini gözden geçirmişlerdir.

Milletari, Navab ve Ahmadi [20], 2012 yılındaki Prostat MR Görüntü Bölütleme kısaca PROMISE (Prostate MR Image Segmentation) yarışmasının veri setindeki manyetik rezonans görüntülerinden prostat görüntülerini bölütlemek için 3B KSA kullanmışlardır. Çalışma kapsamında önerdikleri V-Net, U-Net mimarisinden ilham almıştır ve 50 MR prostat taraması üzerinde eğitilerek 30 benzer taramada test edilmiştir. V-Net, yarışmadaki en üst sırada yer alan takımlarinkine benzeyen bir başarıma ulaşmıştır.

Pereira ve arkadaşları [21], 11 adet konvolüsyon katmanına sahip derin bir KSA mimarisi tasarımıyla 3×3 boyutunda küçük filtreler uygulamışlardır. KSA'ları, 274 MR beyin tümörü taraması üzerinde eğitilmiş olup 2013 yılındaki Beyin Tümörü Bölütleme kısaca BRATS

(Brain Tumor Segmentation) yarışmasında birinciliği ve 2015 yılındaki BRATS yarışmasında ise ikinciliği elde etmiştir.

Chen ve arkadaşları [22], yaptıkları çalışmada üst örneklenmiş filtreler ve tam bağlı Koşulsal Rastgele Alan kısaca CRF (Conditional Random Field) kullanılmasını önermişlerdir. Bunlar, her filtrenin görüş alanını birden fazla ölçekte büyötmeye yardımcı olur.

Kline ve arkadaşları [23], MR görüntülerinden polikistik böbreklerin tam otomatik bölütlenmesi için kodlayıcı ve kod çözücü ağlardan oluşan bir derin sinir ağı mimarisi önermişler ve başarılı sonuçlar elde etmişlerdir.

Mishra ve Daescu [24], 2017’de düzenlenen uluslararası biyomedikal sempozyumundaki yarışmada Uluslararası Cilt Görüntüleme İşbirliği kısaca ISIC (International Skin Imaging Collaboration) dermoskopik veri seti görüntülerindeki cilt lezyonlarını otomatik olarak bölütlemek için bir KSA mimarisi önermişlerdir. Önerilen mimari, bazı parametreleri U-Net mimarisiyle birleştirerek elde edilmiştir. 2017 yılındaki bu yarışmanın en başarılı ilk üç katılımcısı [25-27], hassas ve doğru lezyon bölütlemeleri elde etmek için derin öğrenme mimarileri kullanmıştır.

Vesal, Ravikumar ve Maier [28], cilt lezyonu bölütlemedeki zorluklardan bahsederek cilt lezyonu bölütlemesi için doğru ve otomatik bir yaklaşım gerektiğinin altını çizmişlerdir. Bu amaçla SkinNet adlı bir KSA önermişlerdir. Önerilen KSA, U-Net’in değıştirilmiş bir sürümüdür. ISIC 2017 veri setini kullanarak yaklaşımlarının performansını bilimsel yazındakilerle karşılaştırmışlar ve çok başarılı olduklarını kanıtlamışlardır.

Bu tez çalışmasında, ilk olarak son yıllarda makine öğrenmesi alanında çok fazla ilgi duyulan derin öğrenme yaklaşımı hakkında ayrıntılı bilgilere yer verilmiştir. Daha sonra, derin öğrenme tabanlı KSA mimarilerinde değışiklik yapılarak elde edilen FCN-AlexNet, FCN-8s, FCN-16s ve FCN-32s TKA mimarileriyle otomatik cilt lezyonu bölütleme uygulaması gerçekleştirilmiştir. Bu uygulama ile dermoskopik görüntülerde melanomun daha iyi tespit edilebilmesine yardımcı olunması amaçlanmıştır. Veri seti ve kullanılan ağ mimarilerine göre bu çalışma, bilimsel yazında ilk kez gerçekleştirilmiştir. Elde edilen deneysel sonuçlar, cilt lezyonu başta olmak üzere tıbbi görüntü bölütleme uygulamalarında TKA’ların elverişli olduğunu göstermiştir.

1.1. Tezin Organizasyonu

Tez, 6 bölümden oluşmaktadır. Birinci bölümde, derin öğrenme tabanlı tıbbi görüntü bölütleme ile ilgili olan tezin hedefi ve bu alandaki bilimsel yazın taramasına yer verilmiştir.

İkinci bölümde, öncelikle derin öğrenme kavramının daha iyi anlaşılabilmesi için gerekli kavramlara ayrıntılı olarak yer verilmiştir. Daha sonraki kısımda, tez çalışmasının temelini oluşturan derin öğrenme mimarilerinden KSA'ların, çok katmanlı ileri beslemeli YSA türü olmasından dolayı ileri beslemeli sinir ağı hakkında bilgiler verilmiştir.

Üçüncü bölümde, KSA'lar hakkında detaylı bilgilere yer verilmiştir. İlk olarak KSA mimarilerini oluşturan yapılar detaylı olarak açıklanmış, daha sonra KSA'nın eğitimi ve bu eğitimin performansını etkileyen parametreler hakkında bilgiler verilmiştir.

Dördüncü bölümde, öncelikle görüntü sınıflandırma ile anlamsal görüntü bölütleme arasındaki esas fark anlatılmış ve tıbbi görüntü bölütleme ile ilgili temel bilgilere yer verilmiştir. Sonrasında, tez kapsamında gerçekleştirilen tıbbi görüntü bölütleme uygulamasında kullanılan TKA'lara, KSA'lardan nasıl geçiş yapıldığı detaylı olarak açıklanmıştır.

Beşinci bölümde, ilk olarak gerçekleştirilen tıbbi görüntü bölütleme uygulaması için kullanılan veri seti ve performans değerlendirme metrikleri hakkında bilgiler verilmiştir. Daha sonra, dördüncü bölümde anlatılan TKA mimarileri kullanılarak elde edilen deneysel sonuçlara yer verilmiş ve TKA'ların bölütleme performansları karşılaştırılmıştır.

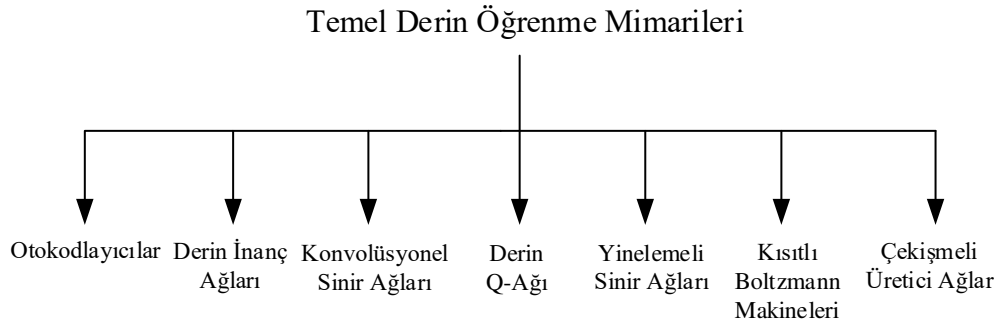
Altıncı bölümde ise elde edilen sonuçlar ve öneriler sunulmuştur.

2. DERİN ÖĞRENME

Derin öğrenme, makinelerin görüntü ve ses gibi büyük miktardaki verileri algılaması ve anlamasına yönelik bir model oluşturmak için kullanılan popüler bir makine öğrenmesi yaklaşımıdır. Temel olarak bu yaklaşım, YSA'ların yapısal olarak daha karmaşık hali olan derin mimarilere dayanmaktadır. Buradaki derin mimari, gizli katmanlarının sayısı arttırılmış YSA'ları ifade etmektedir.

Derin öğrenme algoritmalarının makine öğrenmesindeki var olan algoritmalarından ayrılan tarafı; çok yüksek miktarda veriye ve bu yüksek veriyi işleyebilecek çok yüksek hesaplama gücü olan donanımlara ihtiyaç duymasıdır. Son yıllarda, özellikle bilgisayarla görme alanında etiketli verilerin sayısı aşırı derecede artmıştır. GPU tabanlı paralel hesaplama gücü alanındaki büyük ilerlemelerin de etkisiyle derin öğrenme yaklaşımı daha çok ilgi görmüştür. Binlerce hesaplama çekirdeği sahip olan GPU'lar, bu verilerin işlenmesinde CPU'lar ile karşılaştırıldığında 10 ile 100 kat arasında uygulama performansı sunmaktadır. Günümüzde derin öğrenme, otomatik konuşma tanıma, görüntü tanıma ve doğal dil işleme başta olmak üzere birçok uygulama alanına sahiptir.

Birçok farklı derin öğrenme mimarisi bulunmaktadır. Temel olarak derin öğrenme mimarileri, kullanım amacı ve ağ yapılarına göre Şekil 2.1'deki gibi gruplandırılmıştır [29].



Şekil 2.1. Temel derin öğrenme mimarileri [29]

Derin öğrenme kavramının daha iyi anlaşılabilmesi için öncelikle makine öğrenmesinin ne olduğunun bilinmesi ve YSA yapılarının iyi bir şekilde benimsenmesi gerekmektedir. Bu sebeple öncelikle bu bilgilere yer verilecektir. Ayrıca, çalışmamızın temelini oluşturan derin öğrenme mimarilerinden KSA'ların, çok katmanlı ileri beslemeli YSA türü olmasından dolayı ileri beslemeli sinir ağı üzerinde durulacaktır.

2.1. Makine Öğrenmesi

Makine öğrenmesi, çeşitli matematiksel yöntemler kullanarak mevcut verilerden bilgi edinen ve edindiği bu bilgi ile tahminler yapabilen yöntemler bütünü olarak düşünülebilir. Yani, makineler öğrenme yeteneğinin kazandırılmasıdır. Nesne tanıma, yüz tanıma, ses tanıma, tıbbi teşhis tahmini, hava tahmini gibi birçok uygulama makine öğrenmesi ile gerçekleştirilmektedir [3].

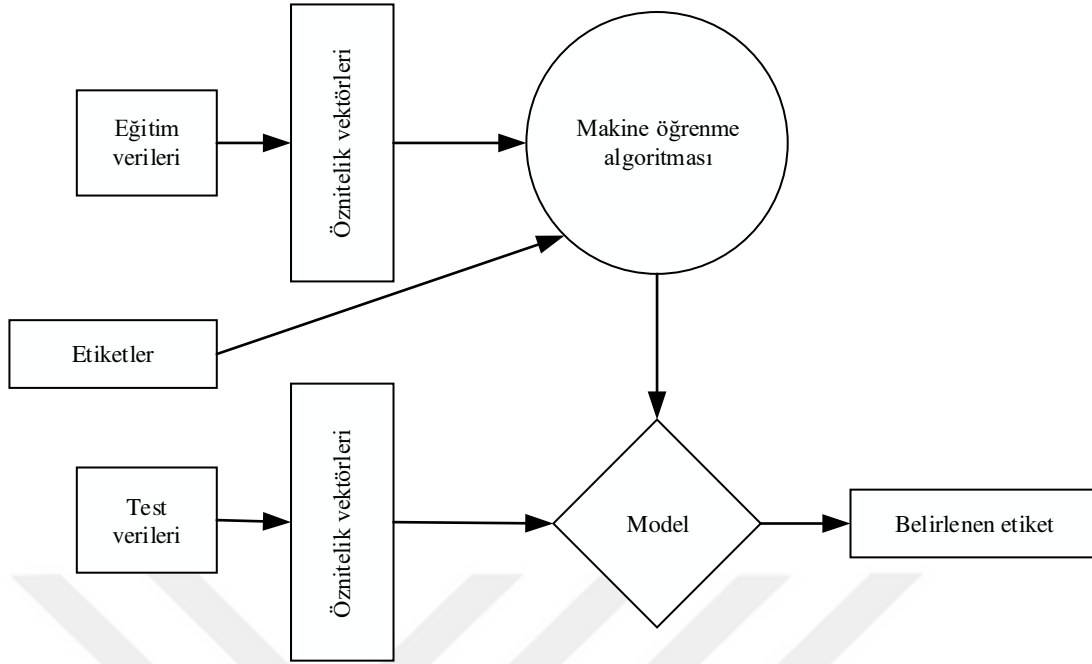
Makine öğrenmesi algoritmaları eldeki ham veriyi kullanmadan önce bu verileri temsil eden anlamlı verileri yani öznitelikleri çıkarır. Her bir ham veri belirli bir sınıfa ait olarak tanımlı olabilmektedir. Eğitim aşamasında, algoritmada kullanılan matematiksel yöntemler mevcut öznitelikleri işler ve ortaya bir model ortaya koyar. Artık öğrenen bu model aracılığıyla herhangi bir ham veri hakkında tahminde bulunulabilir [3].

Makine öğrenmesi kullanılarak oluşturulan sistemler genel olarak iki farklı öğrenme modeli kullanmaktadır: gözetimli ve gözetimsiz öğrenme [30].

2.1.1. Gözetimli Öğrenme

Gözetimli öğrenme, sistemin etiketli veriler kullanılarak eğitilmesiyle sağlanan bir öğrenme türüdür. Eğitim aşamasında etiketli veriler kullanıldığı için model, ham bir verinin sınıfını tahmin edebilmektedir.

Şekil 2.2, gözetimli öğrenme modelinin iş akışını göstermektedir.



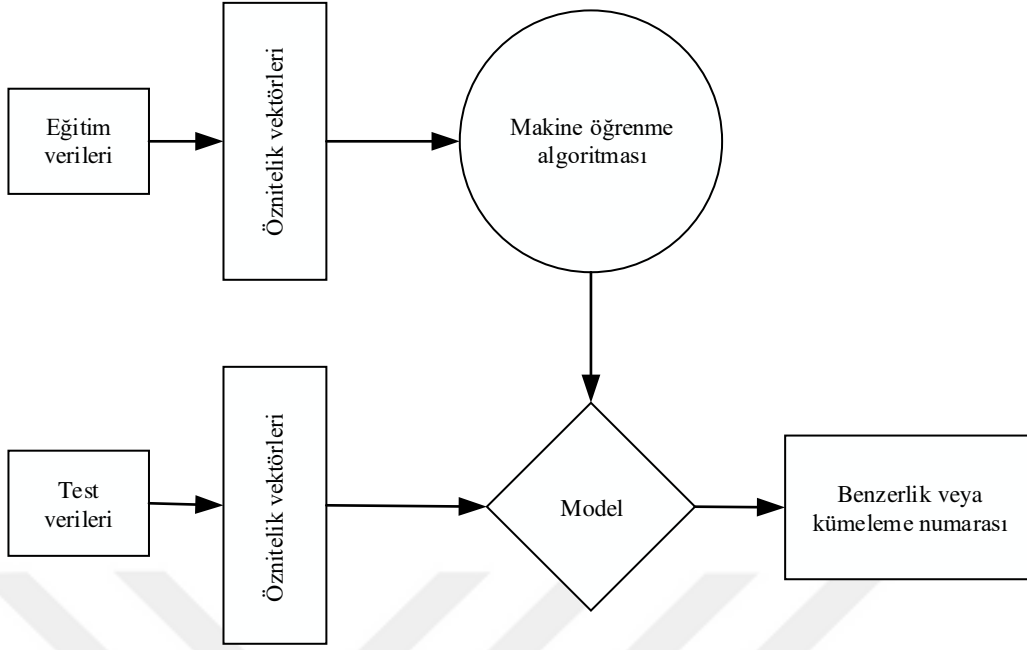
Şekil 2.2. Gözetimli öğrenme modelinin iş akışı

DVM, YSA, Basit Bayes, k-En yakın Komşu (k-EK), Lojistik Regresyon, Rastgele Orman ve Karar Ağaçları gibi algoritmalar, gözetimli öğrenme modeli oluşturulurken yaygın olarak kullanılmaktadır [31]. Sıklıkla, sınıflandırma ve regresyon analizi gibi amaçlarda kullanılmaktadırlar.

2.1.2. Gözetimsiz Öğrenme

Gözetimsiz öğrenmede, sistemin etiketsiz veriler kullanılarak öğrenmesi sağlanır. Eğitim aşamasında etiketli veriler kullanılmadığı için model, ham verileri birbirlerine benzer olmalarına göre gruplandırabilmektedir. Gözetimsiz öğrenmede amaç, veri setindeki örneklerin çıkışları bilinmediği için tanıma veya sınıflandırma değildir.

Şekil 2.3, gözetimsiz öğrenme modelinin iş akışını göstermektedir.



Şekil 2.3. Gözetimsiz öğrenme modelinin iş akışı

Parçalayıcı ve hiyerarşik kümeleme algoritmaları, gözetimsiz öğrenme modeli oluşturulurken yaygın olarak kullanılan algoritmalarlardır. Genellikle kümeleme, olasılık yoğunluk tahmini ve boyut indirgeme gibi amaçlar için kullanılmaktadırlar.

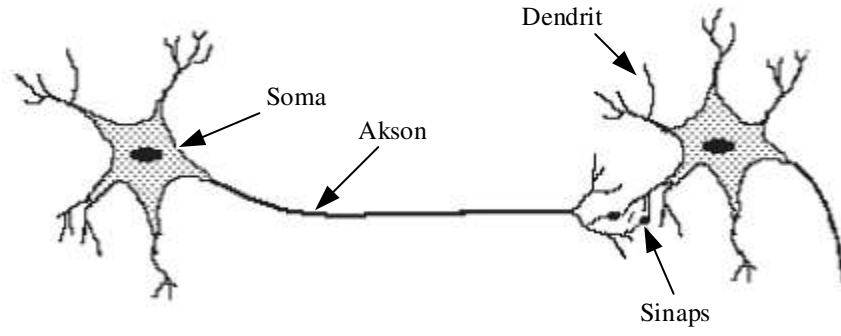
2.2. Yapay Sinir Ağları

YSA'lar, insandaki öğrenme işleyişinden harekete çıkarak geliştirilmiştir. Biyolojik sinir sisteminde bulunan nöronların (sinir hücresi) birbirleri ile ilişki kurması gibi, YSA sistemlerinde de yapay nöron olarak tanımlanan yapılar, birbirleri ile bağlantılı olacak şekilde modellenmişlerdir. Bu sayede YSA'ların, öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahip olacağı düşünülmüştür [3].

YSA'lar, kontrol ve sistem tanımlama, görüntü ve ses tanıma, tahmin ve kestirim, arıza analizi, tıp, haberleşme, trafik, üretim yönetimi ve daha birçok alanda kullanılabilmektedir.

2.2.1. Nöron

Biyolojik nöronlar, biyolojik sinir ağlarının en temel elemanlarıdır. Bir nöron; soma, dendrit ve akson olarak adlandırılan üç ana kısımdan oluşur. Şekil 2.4'te biyolojik bir nöron yapısı gösterilmiştir.



Şekil 2.4. Biyolojik nöronun yapısı

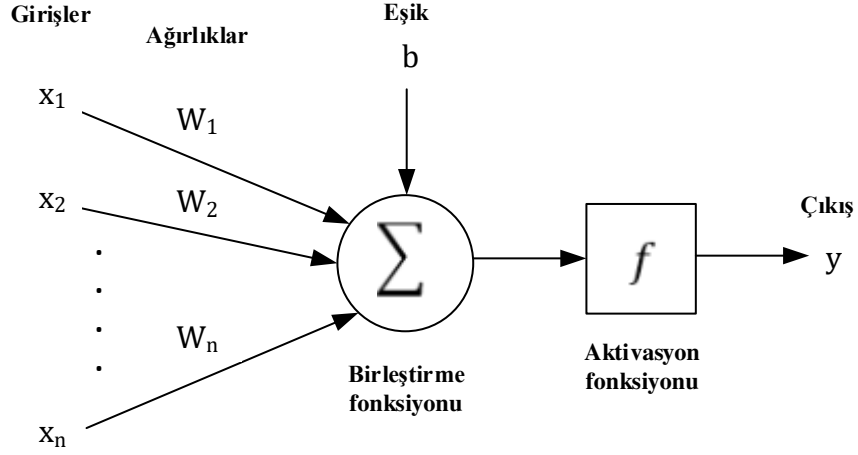
Bir nörona diğer nöronlardan gelen uyarımlar, dendritler aracılığıyla somaya taşınır ve nöron içi aktivasyonun bozulmasıyla oluşan kimyasal bir süreç içerisinde aksonlarla diğer nöronlara iletilir. Sinaptik boşluklar (sinaps) ise uyarımların diğer nöronlara taşınabilmesinde akson uçları ile dendritler arasında rol oynar. Sinaptik boşluklar içerisinde yer alan sinaptik kesecikler, gelen uyarımların diğer nöronlara dendritler aracılığıyla geçmesini sağlayan elemanlardır. Sinaptik boşluğa, sinaptik kesecikler tarafından sağlanan nöro-iletken maddenin dolması uyarımların diğer nöronlara geçişini sağlar. Nöronlara gelen uyarımlarla uyumlu olarak, nöronlar arasındaki mevcut sinaptik ilişkilerin değişimi veya nöronlar arasında yeni sinaptik ilişkilerin kurulması “öğrenme” sürecine karşılık gelmektedir.

Biyolojik sinir ağlarının nöronları olduğu gibi yapay sinir ağlarının da nöronlar vardır. Tablo 2.1’de, biyolojik sinir ağı yapısının YSA karşılıkları verilmiştir.

Tablo 2.1. Biyolojik sinir ağı yapısının yapay sinir ağı karşılıkları

Biyolojik Sinir Sistemi	Yapay Sinir Sistemi
Nöron	Hesaplama birimi
Dendrit	Birleştirme fonksiyonu
Soma	Aktivasyon fonksiyonu
Akson	Nöron çıkışı
Sinaps	Ağırlık

Nöron, YSA'daki temel hesaplama birimi olarak adlandırılabilir. Nöronlara düğüm veya birim de denebilmektedir. Yapay bir nöronun yapısı Şekil 2.5'te gösterilmiştir.



Şekil 2.5. Yapay nöronun yapısı

2.2.1.1. Giriş

Girişler, bir nörona dış dünyadan gelen bilgilerdir. Bunlar, ağız öğrenmesi istenen örnekler tarafından belirlenir.

2.2.1.2. Ağırlık

Ağırlıklar, nörona gelen bilginin önemini ve nöron üzerindeki etkisini göstermektedir. Her bir girişe ait ayrı ağırlıklar bulunmaktadır. Örnek olarak Şekil 2.5'teki W_1 ağırlığı, x_1 girişinin nöron üzerindeki etkisini göstermektedir. Ağırlıkların büyük ya da küçük olması önemli veya önemsiz olduğu anlamına gelmemektedir.

2.2.1.3. Eşik

Bias olarak da adlandırılan eşik, her bir nörona eğitilebilir sabit bir değer sağlamak amacıyla kullanılmaktadır.

2.2.1.4. Birleştirme Fonksiyonu

Birleştirme fonksiyonu, bir nörona gelen net girişi hesaplar. Bunun için çok sayıda birleştirme fonksiyonu bulunmasına rağmen yaygın olarak kullanılanı, ağırlıklı toplamıdır. Her gelen bilgi kendi ağırlığıyla çarpılarak toplanır. Böylece ağa gelen net giriş bulunmuş olur.

2.2.1.5. Aktivasyon Fonksiyonu

Aktivasyon fonksiyonu, nörona gelen net girişi işleyerek nöronun bu girişe karşılık üreteceği çıkışı belirler. Aktivasyon fonksiyonu genelde doğrusal olmayan bir fonksiyon seçilir. Aktivasyon fonksiyonunun amacı, nonlinearliği nöronun çıkışına (2.1)'deki gibi aktarmaktır. YSA'ların bir özelliği olan “nonlinearlik”, aktivasyon fonksiyonlarının doğrusal olmayışından kaynaklanmaktadır.

$$y = f\left(\sum_{i=1}^n W_i x_i + b\right) \quad (2.1)$$

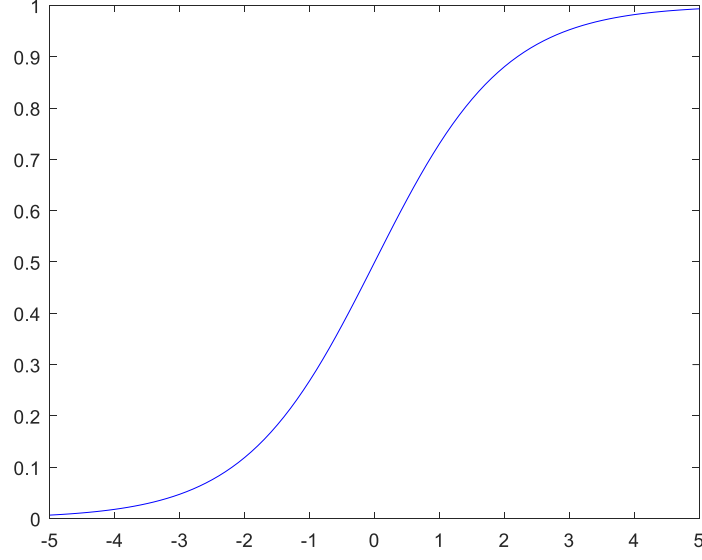
Aktivasyon fonksiyonu seçerken dikkat edilmesi gereken önemli husus, fonksiyonun türevinin kolay hesaplanabilir olmasıdır. Böylelikle hesaplamaların hızlı bir şekilde gerçekleşmesi sağlanmış olur.

Bilimsel yazında; doğrusal, adım, sigmoid, tanjant hiperbolik (tanh), Düzeltilmiş Doğrusal Birim (DDB), eşik değer ve sinüs fonksiyonları gibi birçok aktivasyon fonksiyonu bulunmaktadır. Ancak, yapay sinir ağları uygulamalarında genellikle sigmoid, tanh ve DDB aktivasyon fonksiyonları kullanılmaktadır.

2.2.1.5.1. Sigmoid Aktivasyon Fonksiyonu

(2.2) ile ifade edilen sigmoid aktivasyon fonksiyonu, sürekli ve türevi alınabilir bir fonksiyondur. Şekil 2.6'da verilen karakteristiğe sahiptir. YSA'larda en çok kullanılan aktivasyon fonksiyonlarından biridir. Genellikle son katmanda ikili sınıflandırma problemleri için kullanılır. Bu fonksiyon, giriş değerlerinin her biri için 0 ile 1 arasında bir değer üretir.

$$\sigma(x) = \frac{e^x}{1 + e^x} \quad (2.2)$$

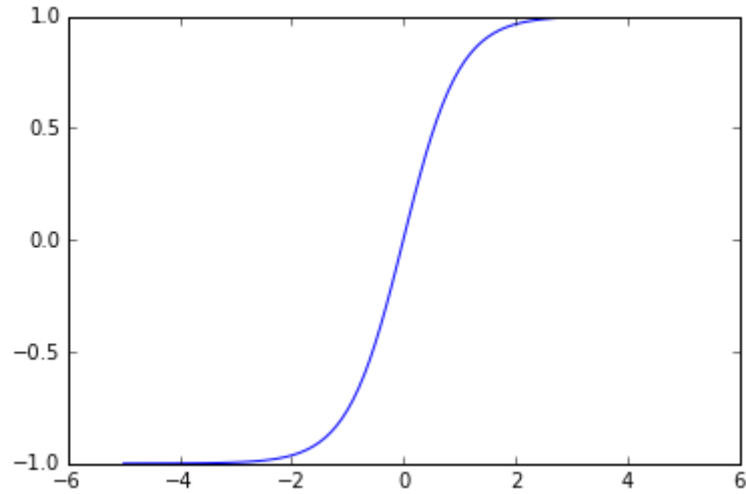


Şekil 2.6. Sigmoid aktivasyon fonksiyonunun karakteristiği

2.2.1.5.2. tanh Aktivasyon Fonksiyonu

(2.3) ile ifade edilen tanh aktivasyon fonksiyonu, Şekil 2.7’de verilen karakteristiğe sahiptir. Bu fonksiyon, sigmoid aktivasyon fonksiyonuna benzerdir. Ancak, çıkış değerleri -1 ile 1 arasında değişmektedir.

$$\tanh(x) = 2\sigma(2x) - 1 \quad (2.3)$$

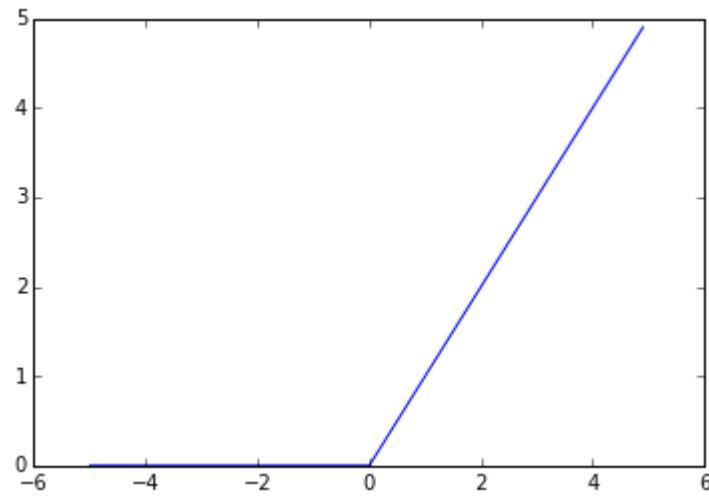


Şekil 2.7. Tanh aktivasyon fonksiyonunun karakteristiği

2.2.1.5.3. Düzeltilmiş Doğrusal Birim Aktivasyon Fonksiyonu

(2.4) ile ifade edilen Düzeltilmiş Doğrusal Birim kısaca DDB aktivasyon fonksiyonu, giriş değerlerinin her biri için 0'ı eşik değer olarak bir çıkış üretir. Şekil 2.8'de verilen karakteristiğe sahiptir. DDB'de Sigmoid'e göre parametreler daha hızlı bir şekilde öğrenilmektedir. Son zamanlarda, YSA'larda kullanımı oldukça popüler hale gelmiştir.

$$f(x) = \max(0, x) \quad (2.4)$$



Şekil 2.8. DDB aktivasyon fonksiyonunun karakteristiği

2.2.1.6. Çıkış

Aktivasyon fonksiyonu tarafından belirlenen çıkış değeridir. Üretilen çıkış, dış dünyaya veya başka bir nörona gönderilir. Ayrıca istenmesi durumunda, nöron kendi çıkışını kendisine giriş olarak da gönderebilir.

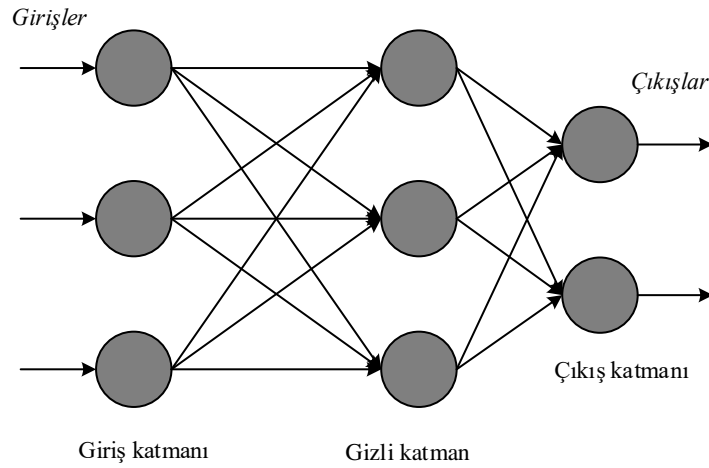
2.2.2. İleri Beslemeli Sinir Ağı

İleri beslemeli yapay sinir ağında, bilgi akışı sadece ileri yönde gerçekleşmektedir. Ağdaki nöronlar, katmanlar şeklinde düzenlenir ve bir katmandaki nöronların çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilir. İleri beslemeli sinir ağları, temel olarak 3 çeşit katmandan oluşur. Bu katmanlar; giriş, gizli (ara) ve çıkış katmanıdır. Giriş katmanı, dış dünyadan aldığı bilgileri hiçbir değişikliğe uğratmadan gizli katmandaki

nöronlara iletir. Daha sonra bu bilgi, sırasıyla dış dünya ile bağlantısı olmayan gizli katman/katmanlarda ve bilgiyi ağdan dış dünyaya aktaran çıkış katmanında işlenerek istenen girişe karşılık ağ çıkışı belirlenir.

İleri beslemeli bir sinir ağı, bir veya birden fazla gizli katmana sahip olabileceği gibi hiç bir gizli katmana sahip olamayabilir. Eğer ağ, hiç bir gizli katman içermiyorsa tek katmanlı algılayıcı, bir veya birden fazla gizli katman içeriyorsa çok katmanlı algılayıcı olarak isimlendirilir. Çok katmanlı algılayıcılar, tek katmanlı algılayıcılar ile doğrusal olmayan ilişkiler öğrenilemediği için geliştirilmiş olup, sıkça tercih edilmektedir.

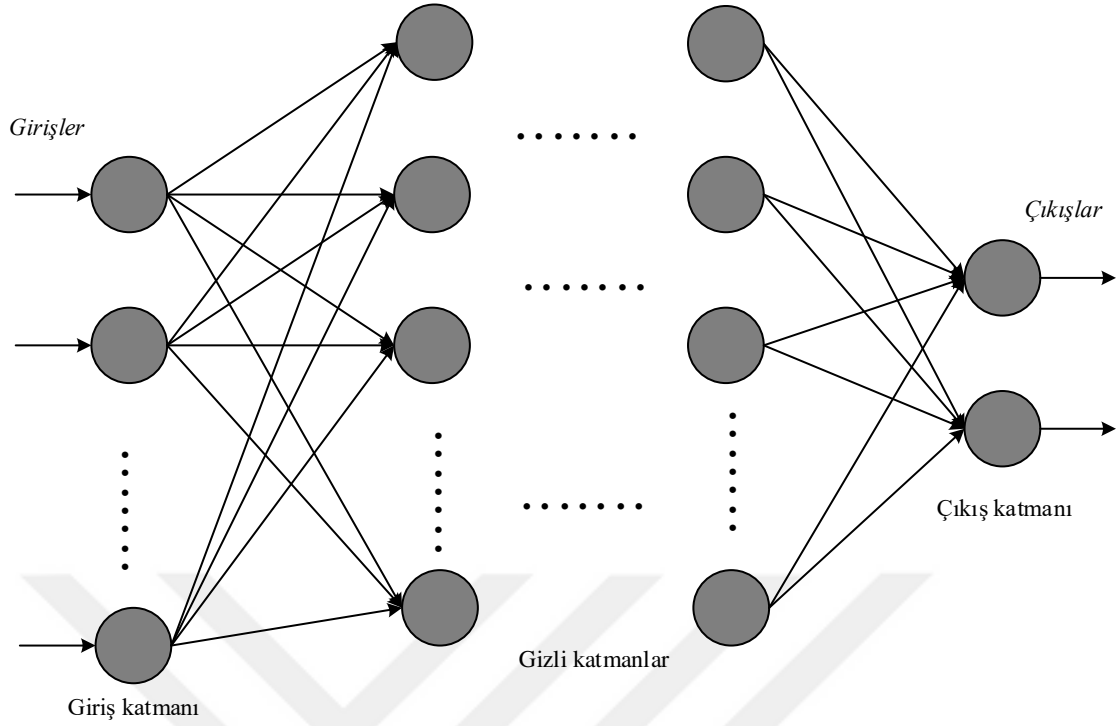
Şekil 2.9’da, 3 katmanlı ileri beslemeli bir YSA modeli örnek olarak gösterilmiştir.



Şekil 2.9. 3 katmanlı ileri beslemeli bir YSA modeli

Çok katmanlı ileri beslemeli bir sinir ağında, her nöron sadece bir sonraki nöronlarla bağlantılıdır, yani aynı katmandaki nöronlar arasında bir bağlantı yoktur.

Çok katmanlı ileri beslemeli bir sinir ağındaki derinlik kavramı, sahip olduğu gizli katmanların sayısı ile ilişkilidir. Ağın gizli katmanlarının sayısı arttıkça derinliği artmaktadır. Kısaca, birden fazla gizli katmana sahip olan bir ağ, derin sinir ağı olarak ifade edilebilir. Şekil 2.10’da, örnek olarak derin bir YSA modeli gösterilmiştir.



Şekil 2.10. İleri beslemeli derin bir YSA modeli

Bir bilgi kaynağından öğrenebilme yeteneği, YSA'ların en önemli özelliklerinden biridir. Çok katmanlı sinir ağlarında öğrenme işlemi, her adımda ağırlıkların değiştirilmesiyle gerçekleşir. Bu yüzden, ağırlıkların nasıl belirleneceği önemlidir. Bilgi tüm ağda saklandığı için bir nöronun sahip olduğu ağırlık değeri tek başına bir anlam ifade etmez. Tüm ağdaki ağırlıklar en uygun değerleri almalıdır. Bu ağırlıklara ulaşabilmek için yapılan işlem, ağı eğitilmesi işlemidir. Kısacası ağı öğrenmesi, ağırlıkların en uygun değerlerinin bulunmasıyla gerçekleşir. Ayrıca, çok katmanlı sinir ağlarının tasarımı gerçekleştirilirken ağdaki gizli katman sayısı, her bir gizli katmanda bulunacak nöronların sayısı, en makul sürede en uygun çözümün bulunması ve ağı doğruluğunun testi gibi birçok dikkat edilmesi gereken husus vardır [32].

En çok bilinen öğrenme algoritmalarından olan geriye yayılım algoritması, çok katmanlı sinir ağlarının eğitiminde sıkça kullanılmaktadır. Bu algoritma sayesinde ağ, büyük veri setlerindeki karmaşık yapıyı keşfetmektedir. Ağ, hem girişler hem de istenen çıkışlar verilmektedir. Ağ, kendisine gösterilen girişlerden genellemeler yaparak problem uzayını temsil eden bir çözüm uzayı üretmektedir. Daha sonra ağa verilen benzer girişler için bu çözüm uzayı, sonuçlar üretebilmektedir [33].

2.2.3. Geriye Yayılım Algoritması

Geriye yayılım algoritması, basitliği ve başarısından dolayı çok katmanlı sinir ağlarının eğitiminde kullanılan en popüler algoritmalarından biridir [32, 33]. Bu algoritma, hataları geriye doğru çıkıştan girişe azaltmaya çalıştığından geriye yayılım olarak adlandırılmıştır. Hata, ağın ürettiği çıkış ile istenen çıkış arasındaki farkı ifade etmekte olup, her bir katmandaki ağırlıkları yeniden hesaplamak için kullanılmaktadır.

Geriye yayılım algoritması, genellikle gradyan iniş (en dik iniş) gibi eniyileme yöntemleriyle birlikte kullanılmaktadır. Gradyan iniş yöntem, hatayı minimuma yaklaştırmak için kullanılır. Hatanın minimize edilmeye çalışıldığı süreç boyunca tüm ağırlıklar güncelleştirilerek ağ çıkışı optimize edilmeye çalışılır.

Çok katmanlı ileri beslemeli bir sinir ağ için geriye yayılım algoritması temel olarak 6 adımdan oluşmaktadır [32]:

$q = 1, 2, 3, \dots, Q$: katmanın numarası,

H_i^p : q 'inci katmandaki i nöronunun girişi,

y_i^q : q 'inci katmandaki i nöronunun çıkışı,

W_{ij}^q : $(q - 1)$ 'inci katmandaki i nöronunu, q 'inci katmandaki j nöronuna bağlayan ağırlık.

Adım 1: Gerçek değerli küçük sayılarla W ağırlık vektörü başlatılır.

Adım 2: Rastgele bir çalışma modeli seçilerek q 'inci katmandaki her bir j nöronu için ileri yönde çıkış değerleri hesaplanır. Çıkış, (2.5) ile hesaplanır.

$$y_i^q = f\left(\sum_i y_i^{q-1} W_{ij}^q\right) \quad (2.5)$$

Adım 3: Çıkış nöronlarındaki hatalar (2.6) ile hesaplanır.

$$\delta_i^Q = (y_i^Q - y_i^p) f'(H_i^Q) \quad (2.6)$$

Adım 4: Giriş katmanı hariç diğer katmanlardaki tüm i nöronları için geriye yayılım ile hata değerleri (2.7) ile hesaplanır.

$$\delta_i^{q-1} = f'(H_i^{q-1}) \sum_i \delta_i^q W_{ij}^q \quad (2.7)$$

Adım 5: Bütün ağırlıklar, W_{ij} 'leri kullanarak (2.8) ile güncellenir.

$$W_{ij}^{yeni} = W_{ij}^{eski} + \Delta W_{ij}^q \quad (2.8)$$

ΔW_{ij}^q , (2.9) ile hesaplanır.

$$\Delta W_{ij}^q = \eta \delta_i^q y_i^{q-1} \quad (2.9)$$

Adım 6: Toplam hata kabul edilebilir bir değere gelene kadar Adım 2'ye dönüp her bir p modeli için işlemler tekrarlanır.

2.3. Derin Öğrenme Yazılım Geliştirme Araçları ve Kütüphaneleri

Derin öğrenmenin iş akışı çok aşamalı, yinelemeli bir süreçtir. Bu süreçte, ilk olarak veriler toplanmalı ve gerekirse ön işleme tabi tutulmalıdır. Başarılı bir derin öğrenmenin gerçekleşebilmesi için büyük boyuttaki veri setlerine ihtiyaç duyulabilmektedir. Günümüzde, internet ve büyük veri kaynakları sayesinde veri setleri hızla büyümektedir. Bu veri setlerini kullanarak ağ eğitilir. Ağı eğitmek için mevcut ağlar kullanılabilir veya yeni ağlar geliştirilebilir. Eğitim aşamasından sonra oluşturulan ağ modellerinin beklendiği gibi çalıştığını doğrulamak için test edilmesi gerekir. Genellikle bu noktada, sonuçların iyileştirilmesi için bazı işlemlerin tekrarlanması gereklidir. Bu işlemler, verileri yeniden işlemek, ağı düzenlemek, ağların veya çözücülerin parametrelerini değiştirmek ve istenen sonuç elde edilinceye kadar yeniden test etmeyi içerir. Derin öğrenme sürecinde yapılacak olan bu yoğun hesapsal işlemler için bilgisayarların CPU'ları yetersiz kalmaktadır. Çünkü belli işlem kapasitesi ve mimariye sahip CPU'lar çok sayıda işlemi aynı anda yerine getiremediği için modelin eğitim ve test aşamaları çok fazla zaman almaktadır. Bu yüzden, CPU'lar yerini verilerin paralel işlenmesine olanak veren GPU'lara bırakmıştır. Bu sayede, derin öğrenme hızlı bir şekilde gerçek hayattaki uygulamalarda kullanılmaya başlamıştır.

Derin öğrenme uygulamalarında GPU'ların paralel hesaplamalar yapabilmesi, NVIDIA'nın sunduğu CUDA eklentisi ile sağlanmaktadır [34]. CUDA, NVIDIA'nın GPU gücünü kullanarak hesaplama performansında yüksek derecede artışa olanak sağlayan paralel hesaplama mimarisidir. CUDA, genel amaçlı GPU hesaplamaları için grafik işlemci çekirdeklerinin kullanılmasını sağlar. İlk sürümü 15 Şubat 2007'de yayınlanan CUDA eklentisi GeForce, Quadro ve Tesla serisine sahip NVIDIA GPU'larda çalışmaktadır. Yazılım geliştiriciler ve araştırmacılar, CUDA destekli GPU'lar aracılığıyla bilgisayarla görme, sismik analiz ve hesaplama dayalı Fen Bilimleri gibi birçok alandaki uygulamalarını hızlı bir şekilde gerçekleştirebilmektedir.

Derin öğrenme algoritmalarının uygulanabilmesi için Caffe, Torch, Theano, TensorFlow, Keras ve DIGITS başta olmak üzere birçok popüler yazılım geliştirme aracı ve kütüphanesi mevcuttur. Bunların büyük çoğunluğu, GPU üzerinde de çalışabilmektedirler.

2.3.1. Caffe

Yangqing Jia tarafından hazırlanan Caffe [35] derin öğrenme yapısı, Berkeley Görü ve Öğrenme Merkezi kısaca BVLC (Berkeley Vision and Learning Center) ve kullanıcı topluluğu tarafından geliştirilmiştir. Caffe, tıpkı insan beyni gibi hızlı ve modüler olacak şekilde tasarlanmıştır.

Model ve eniyilemeler kodlama yapılmaksızın ayar dosyası üzerinden yapılabilmektedir. GPU üzerinde eğitim işlemini yapmak için CPU ve GPU değişimi bir etiket ayarı ile gerçekleştirilebilmektedir. Komut satırı, C++, Python ve MATLAB arayüzlerine sahiptir.

Caffe'nin en önemli özelliklerinden biri "Model Zoo" olarak adlandırılan bir internet arşivine sahip olmasıdır. Burada önceden eğitilmiş ağ modelleri bulunmaktadır. Kullanıcılar, bu modellere ücretsiz bir şekilde erişip kendi veri setlerinde kullanabilme olanağına sahiptirler. Ayrıca, açık kaynak kodlu olan Caffe'nin geliştirilmesine katkı sağlayan büyük bir topluluk mevcuttur.

Caffe, endüstriyel ve akademik araştırma uygulamalarında sıkça tercih edilmektedir. Bunun en önemli sebebi verileri hızlı işleyebilmesidir. Caffe bir günde 60 milyon üzerinde görüntüyü tek bir NVIDIA K40 GPU ile işleyebilmektedir. Caffe'nin erişilebilir en hızlı KSA uygulaması olduğuna inanılmaktadır [35].

2.3.2. Torch

LuaJIT diliyle yazılan Torch [36], makine öğrenmesi algoritmaları için geniş destek sağlayan bilimsel bir hesaplama yapısıdır. LuaJIT diliyle yazılması ve C/CUDA uygulama temelini kullandığı için kolay ve verimli bir kütüphanedir. Sayısal eniyileme yöntemlerini kullanabilen bu kütüphane, bünyesinde çeşitli sinir ağı ve enerji tabanlı modelleri bulundurmaktadır. Ayrıca, açık kaynak kodlu olup hızlı ve etkili bir GPU desteği sağlar.

Torch, hesapsal işlemleri oldukça basitleştirirken bilimsel algoritmalar oluşturmada azami esneklik ve hız sağlamaktadır. Torch, Lua topluluğunun kullanıma sunduğu yazılım paketlerinin yanısıra büyük kitledeki kullanıcı topluluğu da makine öğrenmesi, bilgisayarla görme, sinyal işleme, görüntü, video, ses ve ağ iletişimi gibi birçok alanda yazılım paketleri sunmaktadır.

Torch sürekli geliştirilmekte olup şu anda Facebook, Google, Twitter ile çeşitli firmalar tarafından kullanılmaktadır.

2.3.3. Theano

Theano [37], çok boyutlu dizileri (tensör) içeren matematik ifadelerini etkili bir şekilde tanımlamayı, değerlendirmeyi ve eniyilemeyi hızlı bir şekilde sağlayan bir Python kütüphanesidir. Bu kütüphane, NumPy kütüphanesiyle birlikte tümleşik olarak çalıştığı için yoğun matematik işlemlerini kolayca gerçekleştirebilmektedir. Ayrıca, dinamik C kodu oluşturabilme seçeneği de sunarak ifadeleri daha hızlı değerlendirebilme olanağı tanımaktadır.

Theano, “float32” tipindeki verileri işlemek için GPU üzerinde çalıştırıldığında CPU’ya kıyasla 140 kata kadar çıkan hızlara ulaşmıştır [38].

2.3.4. TensorFlow

Tensorflow [39], veri akış grafiklerini kullanarak sayısal hesaplamalar yapan açık kaynak kodlu bir derin öğrenme kütüphanesidir. Bu kütüphane, öncelikli olarak makine öğrenmesi ve derin sinir ağları hakkındaki araştırmaları yürütmek amacıyla Google tarafından geliştirilmiştir. Veri akış grafiklerinde bulunan düğümler, matematiksel işlemleri temsil ederken grafik kenarları ise aralarında iletilen tensörleri temsil eder.

TensorFlow, esnek mimarisi sayesinde hesaplamayı tek bir Uygulama Programlama Arayüzü (API) ile sunucu, mobil veya masaüstü cihazdaki bir veya birden çok CPU veya GPU'ya dağıtmaya olanak verir. Komut satırı, C++ ve Python arayüzlerine sahiptir.

Günümüzde popüler bir yere sahip olan Snapchat, Twitter, Google, eBay gibi uygulamalarda TensorFlow'dan da yararlanılmaktadır.

2.3.5. Keras

TensorFlow ve Theano derin öğrenme kütüphanelerinin üzerine kurulmuş modüler bir Python kütüphanesi olan Keras [40], GPU ya da CPU üzerinde çalışmasını bu iki temel kütüphane üzerinden sağlar. Keras'ın yapılandırma dosyasında ufak değişiklikler yapılarak arka planda TensorFlow ya da Theano kullanılmasına olanak sağlanabilmektedir.

Keras, TensorFlow ve Theano kütüphanelerinin arayüzünü oldukça sadeleştirdiği için çok kullanışlı olup bu iki kütüphaneye göre daha kolay uygulama geliştirilebilmektedir. Keras'ın görüntü işleme uygulamalarında oldukça yaygın bir kullanımı vardır.

2.3.6. DIGITS

Derin öğrenmenin başarısı, büyük ve karmaşık derin sinir ağlarını eğitmek için eğitim süresini aylardan sadece bir kaç güne indiren GPU'ları kullanarak önemli ölçüde arttırılmıştır. Caffè, Torch, TensorFlow ve Theano'yu içeren başlıca derin öğrenme yazılım araçları, GPU hızlandırmasını bünyesine dahil etmiştir. Derin sinir ağlarının hem endüstriyel hem de akademik alandaki artan önemi ve GPU'ların büyük rolü nedeniyle 2015 yılında NVIDIA, CUDA Derin Sinir Ağı kütüphanesini (cuDNN) [41] tanıtmıştır. 2016 yılında ise NVIDIA CEO'su ve kurucularından olan Jen-Hsun Huang GPU Teknolojisi konferansında Derin Öğrenme GPU Eğitim Sistemi'ni (DIGITS) [42] kullanıma sunmuştur. DIGITS, kullanıcıların KSA'larını geliştirip test etmelerine yardımcı olan derin öğrenme GPU eğitim sistemidir. Bu sistem, popüler derin öğrenme yazılım araçlarından Caffè, Torch ve TensorFlow'u web arayüzü desteği sağlayarak görselleştirirken eğitim süresini büyük ölçüde azaltmak için de cuDNN kullanarak GPU hızlandırmasını destekler.

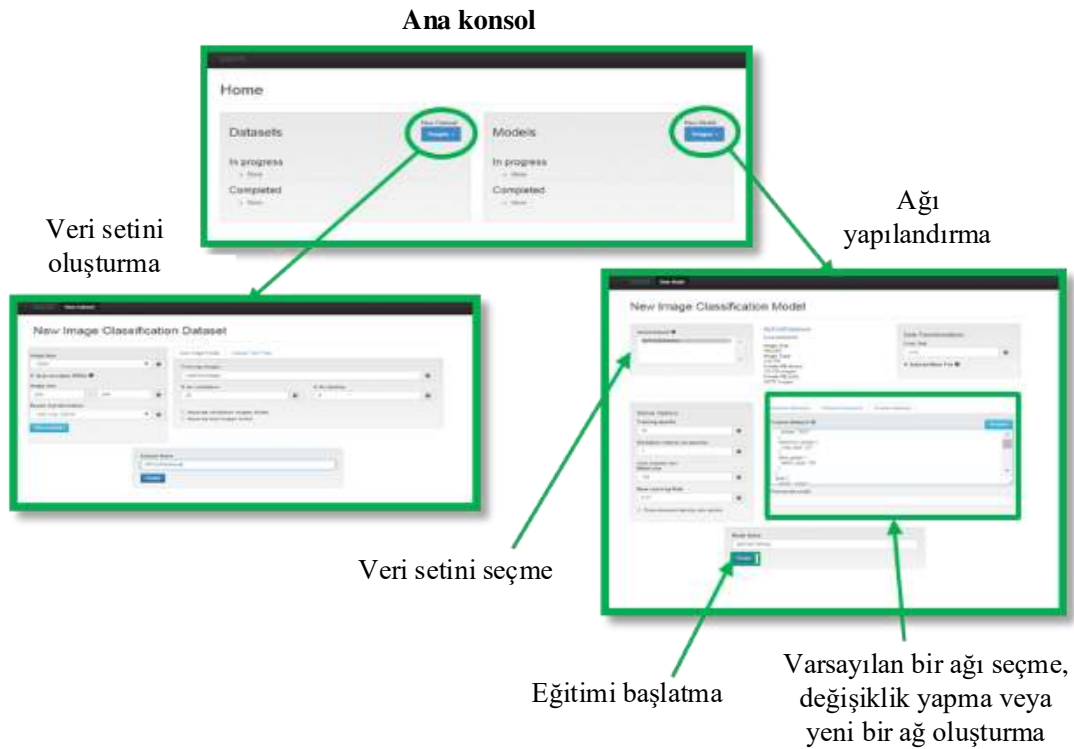
DIGITS, derin sinir ağlarını geliştirmek, eğitmek ve görselleştirmek için yeni bir sistemdir. Derin öğrenmenin gücünü, sezgisel bir tarayıcı tabanlı arayüze yerleştirir. Bu şekilde kolayca web tarayıcısı üzerinden erişilebilen bir web uygulaması olarak çalışır.

DIGITS, GitHub’da mevcut olan açık kaynak bir yazılım olduğu için geliştiriciler projeyi genişletebilir veya özelleştirebilirler ve de projeye katkıda bulunabilirler.

DIGITS, ağı görselleştirmeyi ve doğruluklarını hızlı bir şekilde karşılaştırmayı kolaylaştırır. Bir model seçtiğinizde DIGITS, eğitim aşamasının durumunu ve doğruluğunu gösterir. Ayrıca, ağ eğitilirken veya eğitim tamamlandıktan sonra görüntü yükleme ve test seçeneği sunar.

DIGITS, görüntü sınıflandırma, anlamsal bölütleme, nesne bulma da dahil olmak üzere birçok eğitim hedefini destekler. Buna ek olarak DIGITS, bir iş yükü yöneticisi de içerir. İş yükü yöneticisi, kullanıcının farklı türde birden fazla iş başlatmasına izin verir ve yerel kaynaklara erişimi koordine eder. Birden fazla GPU varsa, işler aynı anda yürütülebilir. Kullanılabilir kaynaklardan daha fazla iş oluşturulursa, kaynaklar serbest kalana kadar işler kuyruğa alınır. DIGITS kontrol paneli, kullanıcının tüm aktif işleri ve mevcut durumlarını izlemesini sağlar.

Şekil 2.11, görüntülerden veri seti oluşturabileceğiniz ve onları eğitim için hazırlayabileceğiniz ana konsol penceresini göstermektedir. Bir veri setine sahip olur olmaz, ağ modelinizi yapılandırabilir ve eğitime başlayabilirsiniz.



Şekil 2.11. DIGITS ana konsolu [42]

DIGITS, ađın eniyilenmesi iin gerekli araları da sađlamaktadır. Ađ yapılandırması iin yapılan ayarlamalar takip edilebilir ve bias, aktivasyon fonksiyonları ve katmanlar gibi parametreler deđiřtirilerek dođruluk maksimize edilebilmektedir. Ana konsol, srmekte olan eđitim iřlemlerinin yanı sıra makine zerinde var olan veri setlerini ve nceden eđitilmiř ađ modellerini de listeler. Bu sayede, ađın eđitim sresi dřrlp modelin dođruluđu arttırılabilmektedir.



3. KONVOLÜSYONEL SİNİR AĞLARI

1989 yılında LeCun tarafından bilgisayarla görme uygulamaları için tanıtılan KSA'lar, çok katmanlı ileri beslemeli YSA'ların bir türüdür [43]. Günümüzde KSA'lar, nesne bulma, nesne tanıma ve anlamsal görüntü bölütleme gibi birçok bilgisayarla görme uygulamaları için modelleri başarılı bir şekilde öğrenebildikleri için derin öğrenme yöntemleri arasında gittikçe popüler hale gelmektedir. KSA'lar görüntü tanımanın yanında konuşma tanıma, görüntü alt yazılama, doğal dil işleme gibi birçok alanda da tercih edilmektedir.

KSA'lar, ham verilerden hiyerarşik öznitelikler çıkaran sınıflandırıcılar olarak düşünülebilir. KSA'larda görüntüler ağı giriş olarak verilir ve herhangi bir öznitelik çıkarıcı yöntem kullanmadan oluşturulan öznitelik hiyerarşisiyle otomatik bir şekilde öğrenme gerçekleşir.

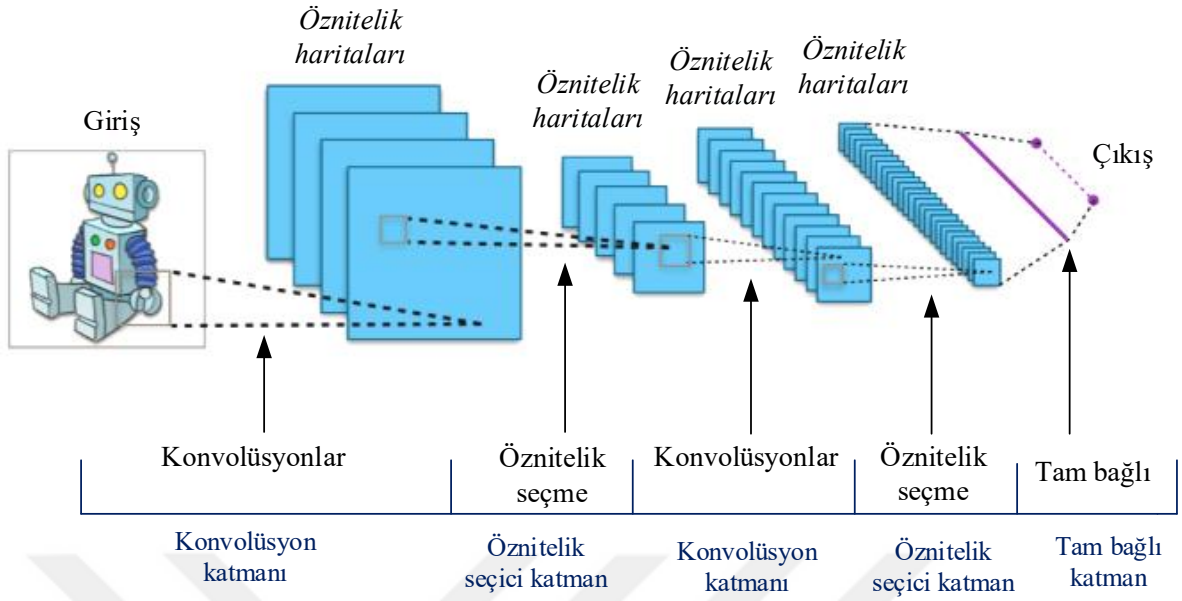
3.1. Mimari

Geleneksel ileri beslemeli YSA'larda bir katmandaki bütün nöronlar bir sonraki katmanın bütün nöronlarına bağlıdır. Bu tür bağlı katmanlara tam bağlı katman denmekte olup KSA'larda tam bağlı katmanlara ek olarak, çıkışı üretmesi için giriş görüntüsüne konvolüsyon uygulanır. Bu, giriş katmanındaki bütün bölgelerin bir sonraki katmanda bulunan nöronlara bağlı olduğu, yerel bağlantılara neden olur. Böylece, bu katmanda kullanılan her bir öğrenilebilir filtre ile giriş görüntüsü konvüle edilerek farklı öznitelik haritaları oluşturulur. Öznitelik haritaları, yüksek katmanlara doğru gittikçe daha karmaşık genellemeler elde etmeyi sağlayarak dönme ve bozulmaya karşı daha duyarsız hale gelir.

Ayrıca, çoğunlukla konvolüsyon katmanında elde edilen öznitelik haritaları öznitelik seçici katmana tabi tutularak uzaysal boyut indirgemesi ve önemli özniteliklerin korunması işlemleri gerçekleştirilir.

Bir sınıflandırıcı, sınıf olasılıklarının bir çıkış olarak üretilmesi amacıyla daima son katmandır. Konvolüsyon ve öznitelik seçici katmanlardan elde edilen son çıkış, bir veya birden fazla tam bağlı katmana aktarılır. Daha sonra da Softmax gibi aktivasyon fonksiyonlarının kullanıldığı sınıflandırıcı katmana aktarılarak çıkış tahmini elde edilir.

Basit bir KSA mimarisi Şekil 3.1'deki gibi konvolüsyon, öznitelik seçme ve tam bağlı katmanların kombinasyonlarından meydana gelmektedir.



Şekil 3.1. Basit bir KSA mimarisi örneği

3.1.1. Konvolüsyon Katmanı

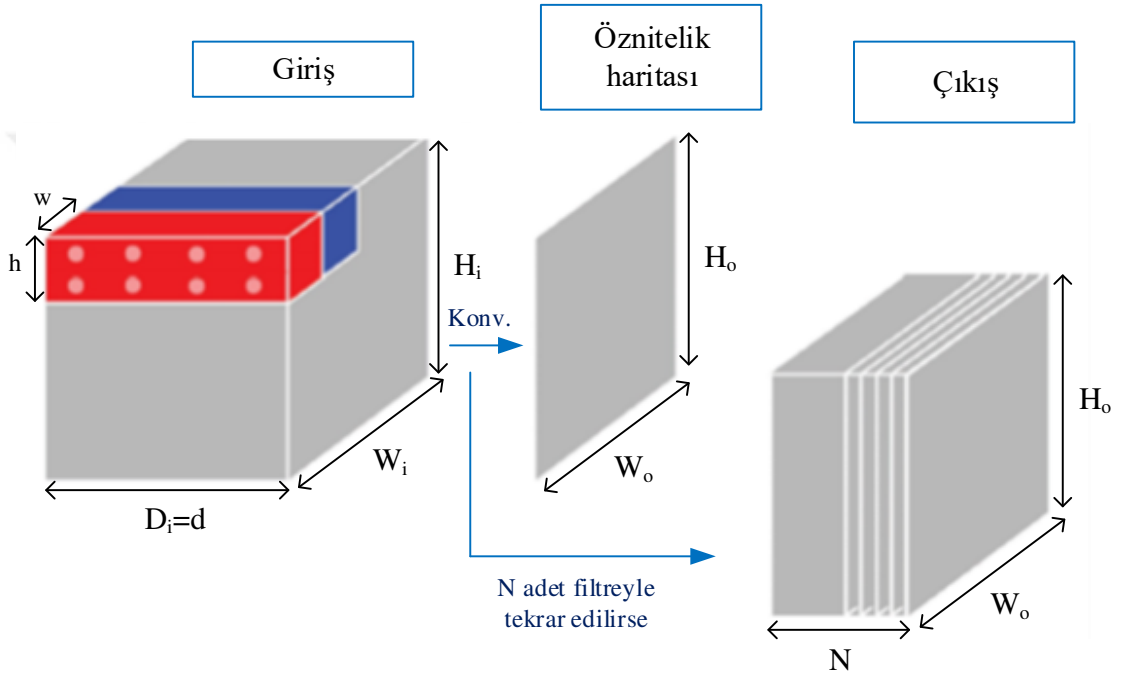
KSA'ların en temel katmanı olan konvolüsyon katmanındaki amaç, giriş görüntüsünün öğrenilebilir filtreler ile konvüle edilerek özniteliklerini çıkarmaktır. Her bir filtre ile bir öznitelik haritası üretilir.

KSA'lar, RGB (Red, Green, Blue–Kırmızı, Yeşil, Mavi) görüntülere uygulandığında, görüntünün 3 boyutlu bir matris olduğu gerçeğine dikkat çeker ve katmanların her biri benzer şekilde düzenlenir. Bu durum, Şekil 3.2'de gösterilmiştir. KSA'ların her bir katmanı, h ve w 'nin uzaysal boyutlar ve d 'nin çekirdeğin (veya filtre) öznitelik kanallarının sayısı olmak üzere $d \times h \times w$ boyutlu belirli bir hacimdeki filtrelerden oluşur. Bu filtrelerin her biri, giriş görüntüsünün eş bir hacmiyle konvolüsyona tabi tutulur. H_i ve W_i 'nin uzaysal boyutlar ve D_i 'nin kanal sayısı olduğu $D_i \times H_i \times W_i$ boyutlu görüntü, uzaysal boyutlar boyunca kaydırılır.

Konvolüsyon, her bir filtredeki nöronların girişte karşılık gelen değerlerle eleman eleman çarpımının toplamını ifade eder. Bu durumda, KSA'lardaki ilk katmanın giriş görüntüsü olduğu düşünülebilir. Her bir katmanda tek bir filtreyle konvolüsyon, kaydırma ve doldurma gibi parametrelerle 2 boyutta bir çıkış elde edilmesini sağlar. Bu işlem sonucunda, girişteki bir filtre için öznitelik haritası veya aktivasyon haritası elde edilir. KSA'ların her konvolüsyon katmanında, her biri öznitelik haritası ile sonuçlanan N adet filtre kullanılır.

Bu öznitelik haritaları, bir konvolüsyon katmanının çıkışını elde etmek için belli bir hacimde bir araya yığılırlar.

Belli bir katmanın bir filtresindeki tek bir nöron, bu tür konvolüsyonları izleyerek önceki tüm katmanlardaki bağlı nöronlar ile haritalanabilir. Buna nöronun etkili alıcı alanı denir. Alt katmanlar, girişin küçük alanlarını temsil etmeyi öğrenirken üst katmanlar ise giriş görüntüsünün daha büyük bir alt bölümüne cevap verdikleri için daha özel anlamlar öğrenirler. Böylece, yerelden genele doğru bir öznitelik hiyerarşisi oluşturulur.



Şekil 3.2. Tek bir konvolüsyon katmanının gösterimi [44]

Şekil 3.2'deki kırmızı ve mavi alanlar, giriş hacmi boyunca kaydırılarak konvolüsyona tabi tutulan $d \times h \times w$ boyutundaki aynı filtrelerin iki konumunu ifade eder. Filtre boyutunun 2×2 olduğunu göz önünde bulundurduğumuzda kaydırma parametresi s 'nin 2 olduğunu görebiliriz. RGB giriş görüntüsü için $D_i=d=3$ olur.

Bir filtre için s parametresi, her uzaysal boyutta filtrenin hareket ettiği aralık olarak tanımlanır. Doldurma parametresi p ise girişin dış kenarlarına eklenen piksellerin sayısına karşılık gelir. Bundan dolayı kaydırma, alt örneklemenin bir giriş aracı olarak düşünülebilir [45].

Genellikle, $h=w=f$ şeklindeki kare filtreler kullanılır. Böyle bir katmanın çıkış hacmi (3.1), (3.2) ve (3.3) eşitlikleri kullanılarak hesaplanır.

$$D_o = N \quad (3.1)$$

$$H_o = \frac{H_i - f + 2p}{s} + 1 \quad (3.2)$$

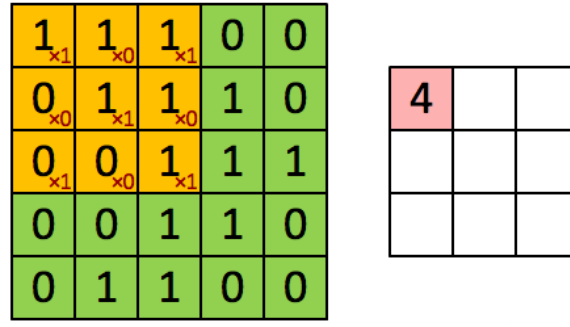
$$W_o = \frac{W_i - f + 2p}{s} + 1 \quad (3.3)$$

Konvolüsyon işleminin daha iyi anlaşılabilmesi için Şekil 3.3'te örnek bir sistem verilmiştir. Burada ikili görüntüyü temsil eden 5×5 boyutunda bir görüntü matrisi ve bu matris üzerinde kaydırılacak 3×3 boyutunda bir filtre görülmektedir.

1 x1	1 x0	1 x1	0	0
0 x0	1 x1	1 x0	1	0
0 x1	0 x0	1 x1	1	1
0	0	1	1	0
0	1	1	0	0

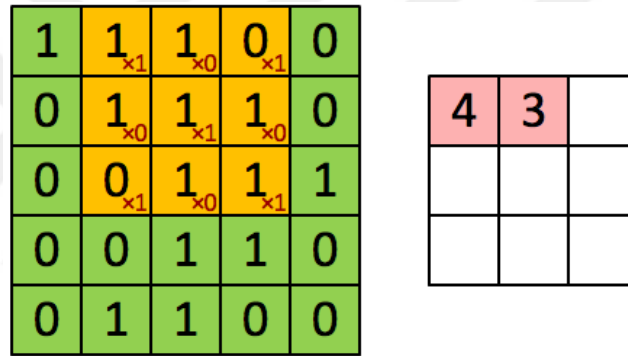
Şekil 3.3. Görüntü matrisi ve filtre örneği [46]

Şekil 3.3'teki sarı alanda görüldüğü gibi filtre değerleri, çarpım durumunda gösterilmiştir. Daha önce de belirtildiği gibi filtre değerleri, üzerinde bulunduğu matris ile eleman eleman çarpılır ve toplamaları alınır. Şekil 3.4'te, ilk konvolüsyon işlemi sonucunda çıkarılan öznitelik gösterilmiştir.



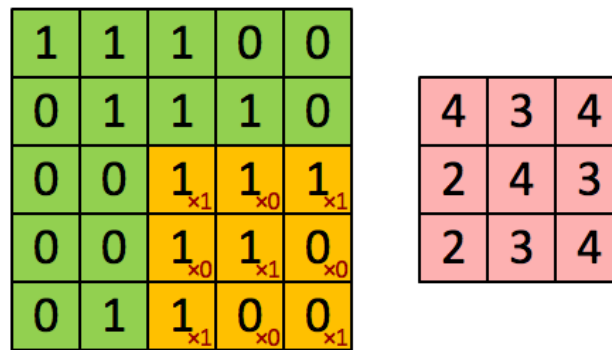
Şekil 3.4. Görüntü matrisi ve öznitelik haritasındaki ilk öznitelik [46]

Filtrenin kaydırılma işlemi soldan sağa doğru olup matrisin sonuna gelene kadar devam eder. Ele aldığımız örnekte kaydırma değeri 1 olarak alınmıştır. Filtrenin bir sonraki kaydırılma işlemi sonucunda çıkarılan diğer öznitelik Şekil 3.5’te gösterilmiştir.



Şekil 3.5. Görüntü matrisi ve öznitelik haritasındaki ilk iki öznitelik [46]

Sıralı şekilde filtrelerin kaydırılmasıyla işlemler tamamlanarak Şekil 3.6’da gösterildiği gibi öznitelik haritasının son durumu elde edilmiş olur.



Şekil 3.6. Görüntü matrisi ve öznitelik haritasının son durumu [46]

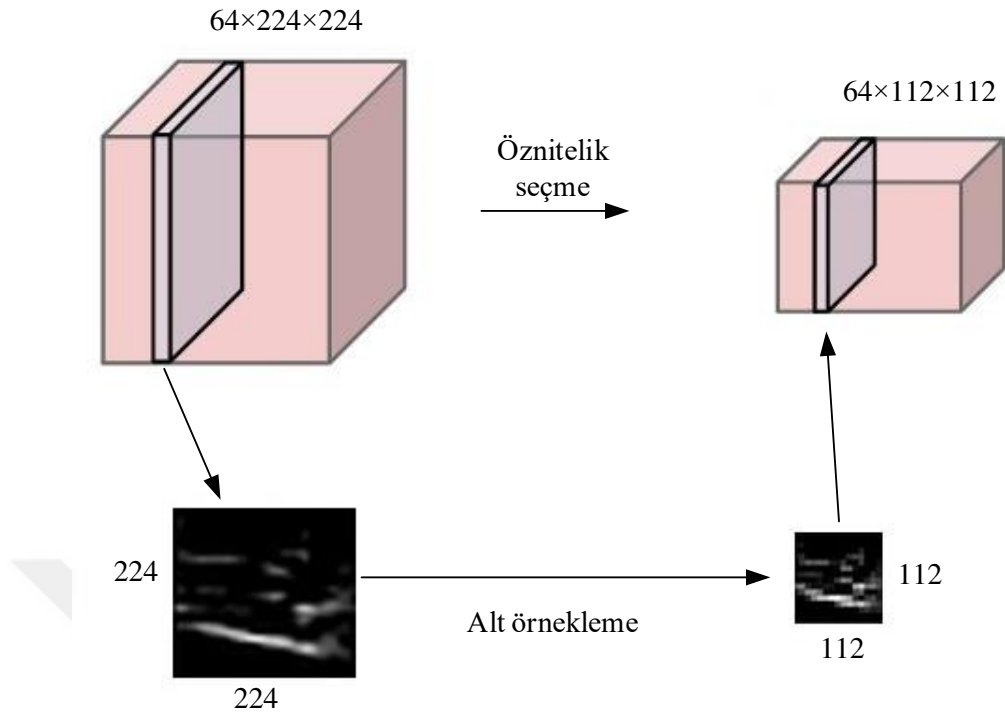
3.1.2. Öznitelik Seçici Katman

Konvolüsyon katmanlarının arasına sıklıkla, giriş özniteliklerini uzaysal olarak alt örnekleme yardımcı olan öznitelik seçici katmanlar serpiştirilir. Öznitelik seçici katmanlar, giriş olarak verilen görüntünün alt örnekleme işlemini yapar. Bu işlem, giriş görüntüsü üzerinde bir filtrenin kaydırılmasıyla gerçekleştirilir.

Giriş görüntüsü (genellikle örtüşmeyen) alt alanlara bölünür ve her bir alt alan doğrusal olmayan öznitelik seçici fonksiyonlarla örneklenir. Bu fonksiyonların en iyi bilinenleri; maksimum ve ortalama öznitelik seçici fonksiyonlardır. Bu çalışmada kullanılan maksimum öznitelik seçici fonksiyonla, her bir alt alandan maksimum değer döndürülür. Ortalama öznitelik seçici fonksiyon ise alt alanın ortalama değerini döndürür.

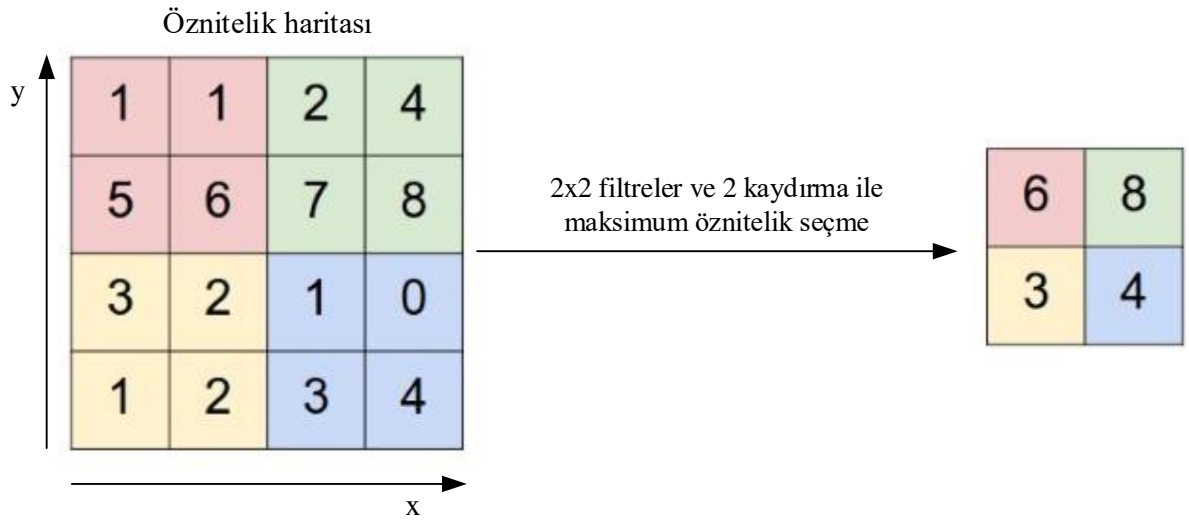
Öznitelik seçme işlemi, görüntüdeki dönüşümsel varyansın miktarını azaltarak ağa gürbüzlük sağlar. Ek olarak, gereksiz öznitelikleri de atarak ağın hesapsal maliyetini de azaltır, dolayısıyla da daha verimli hale getirir.

Öznitelik seçici katmanlar, çıkış boyutları üzerinde kontrol olanağı sağlayan bir kaydırma parametresine de sahiptir. Konvolüsyon katmanlarının çıkış boyutu için kullanılan eşitliklerin aynısı bu katman için de kullanılabilir. Şekil 3.7’de $64 \times 224 \times 224$ olan giriş hacminin 2×2 filtre ve 2 kaydırma ile $64 \times 112 \times 112$ çıkış hacmine alt örneklendiği görülebilir. Öznitelik seçme işlemi her bir öznitelik haritasına ayrı ayrı uygulanır ve şekilde görüldüğü gibi öznitelik haritasının boyutu azaltılır.



Şekil 3.7. Örnek bir alt örnekleme işlemi [1]

Şekil 3.8, 2×2 boyutlu filtre 2 kaydırma ile kaydırılarak maksimum öznitelik seçici fonksiyon ile gerçekleştirilen öznitelik seçme işlemini göstermektedir.



Şekil 3.8. 2×2 boyutundaki filtre ve 2 kaydırma ile maksimum öznitelik seçme işlemi [1]

3.1.3. Düzeltilmiş Doğrusal Birim Katmanı

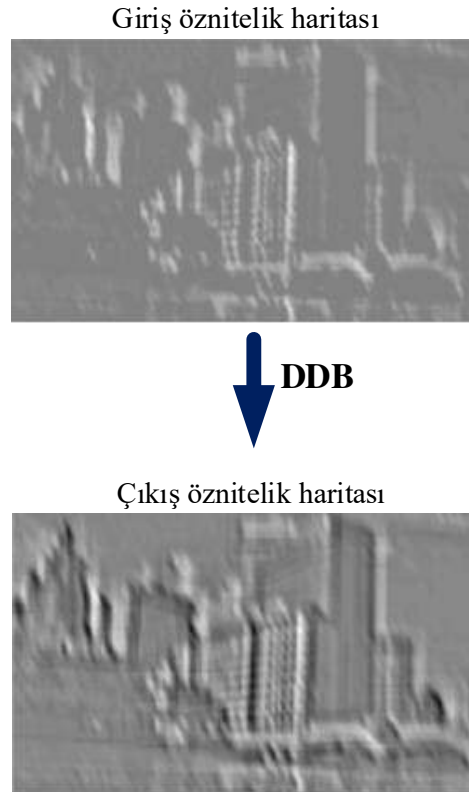
Bölüm 2’de de anlatıldığı gibi aktivasyon fonksiyonları, nörona gelen net girişi işleyerek nonlinearliği nöronun çıkışına aktarmaktadır. Genellikle, konvolüsyon katmanının çıkışları aktivasyon fonksiyonlarına uygulanır. Bu amaçla önerilen nonlinearlik katmanları sigmoid, tanh ve DDB gibi fonksiyonlarından oluşabilir. Daha önceki çalışmalarda DDB’nin KSA’larda daha etkili olduğu görüldüğünden sıklıkla tercih edilmektedir [47].

Bir DDB katmanı, negatif girişleri 0’a eşikler ve pozitif girişleri değişmeden geçirirerek (3.4)’te ifade edildiği gibi aktifleştirir.

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & \text{diğer} \end{cases} \quad (3.4)$$

Burada; x DDB’nin girişini, $f(x)$ ise doğrultulmuş çıkışı ifade etmektedir.

DDB katmanında, her bir piksel değeri için ayrı ayrı işlem uygulanır. Örneğin; giriş öznitelik haritasında siyah alanları negatif değerli piksellerin, beyaz alanları ise pozitif değerli piksellerin temsil ettiği düşünülürse DDB’nin çıkışı Şekil 3.9’daki gibi olur.

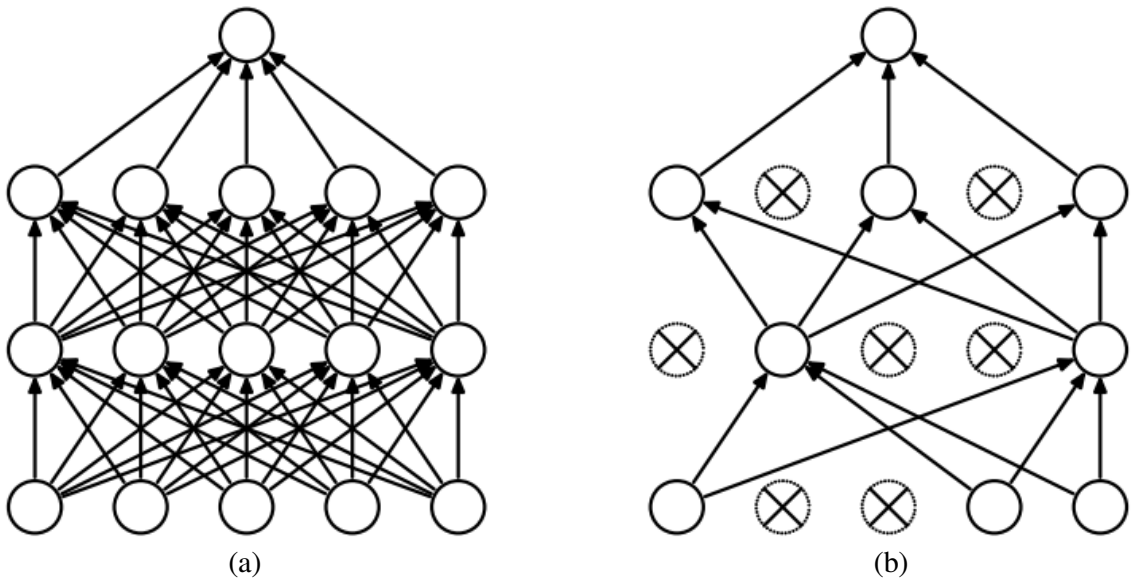


Şekil 3.9. DDB işlem örneği [48]

3.1.4. Düzenleştirme

Eğitim verileri üzerindeki aşırı uygunluk büyük bir sorundur. Özellikle de ağı tek başına eğitim setine çok iyi uyabilecek kadar güçlü olduğu derin sinir ağlarıyla uğraşırken bu durum büyük bir problemdir. Bu nedenle aşırı uygunluktan kaçınılmalıdır. Bunun için geliştirilen yöntemler, düzenleştirme yöntemleri olarak adlandırılır. Düzenleştirme işlemiyle modeldeki bazı nöronların etkisi azaltılarak ağı ezber moduna geçmesi önlenmiş olur.

Dropout (Seyreltme), ağı eğitim aşamasına dahil edilen basit ve etkili bir düzenleştirme stratejisidir. İlk olarak Srivastava ve arkadaşları tarafından tanıtılan Dropout, bir olasılık değeriyle nitelendirilen Dropout katmanları olarak uygulanır [49]. Eğitim aşamasındaki her bir nöron, belirlenmiş bir p olasılığıyla aktif tutulur. Böylece sinir ağı, belirli nöronları bırakarak veya bırakmayarak tekrarlı bir şekilde örneklenir. Bırakılan nörona bağlı tüm kenarlar, her eğitim iterasyonunda çıkarılır ve bir sonraki iterasyondan önce eski haline getirilir. Bu işlem, 3 katmanlı bir sinir ağı için Şekil 3.10'da gösterilmiştir.



Şekil 3.10. Bir ağıdaki seyreltme işlem örneği: (a) Standart bir sinir ağı, (b) Seyreltme işleminden sonra sinir ağı [49]

Tüm nöronlar, tahmin aşamasında aktif kalır. Eğitim sırasında alt örneklemeli Dropout ağlarını hesaba katmak için tüm ağı üzerindeki her nöronun aktivasyonu p ile ölçeklenir.

3.1.5. Tam Bağlı Katman

Konvolüsyon, öznitelik seçici ve DDB katmanlarıyla yüksek seviyeli öznitelikler çıkarıldıktan sonra genellikle ağın sonuna tam bağlı katman yerleştirilir. Bu katmandaki nöronlar, önceki katmandaki tüm aktivasyonlara tamamen bağlıdır. Tam bağlı katmanın en önemli özelliği, alınan tüm aktivasyonları hesaba katarak bu katmandaki nöronların, hangi özniteliklerin en çok hangi sınıfa karşılık geldiğini belirleyebilmesidir. Kısacası tam bağlı katman, sınıflandırıcıyı besleyen katman olarak düşünülebilir.

Tam bağlı katmandaki bir nöron, tüm giriş nöronlarından aktivasyonları aldığı için uzaysal bilgi kaybolur. Uzaysal bilginin çok önem arz ettiği anlamsal tıbbi görüntü bölütleme uygulamamızda bu durum istenmemektedir. Bu durumun üstesinden gelmenin bir yolu, tam bağlı katmanı eşdeğeri olan bir konvolüsyon katmanı olarak görmektir. İleride anlatılacak olan TKA'ların temeli de buna dayanmaktadır.

3.1.6. Sınıflandırıcı

Bir sınıflandırıcı, eldeki problemi ve kullanılan veriyi dikkate alarak seçilir. Bu tez çalışmasında, karşılıklı olarak özel sınıfların dışında bir sınıf için tahmin yapmaya izin veren Softmax fonksiyonu kullanılmıştır. İkili sınıf problemi için Softmax fonksiyonu lojistik bir regresyona indirgenir. Softmax fonksiyonu ile belli bir c sınıfına ait olarak belli bir giriş için sınıf olasılık değeri (3.5)'teki gibi hesaplanır.

$$p_c = \frac{e^{(s_c)}}{\sum_{i=1}^c e^{(s_i)}} \quad (3.5)$$

Burada; s_i , KSA'ların önceki katmanlarından belirli bir sınıf için elde edilen ağ çıkışlarıdır. Tek bir giriş için sınıflar arasındaki tüm olasılık değerlerinin toplamı daima 1'e eşittir. Burada, ağdaki çıkışlar normalize edilmemiş logaritma olasılıkları [1] olarak yorumlanır ve kayıp metriği Softmax fonksiyonunun negatif logaritma olasılığı olarak tanımlanır. Bu bir çapraz entropi kaybıdır.

Softmax fonksiyonunun haricinde, kaybın Hinge kaybı olarak tanımlandığı DVM sınıflandırıcıları kullanmak da yaygındır. DVM sınıflandırıcı, olasılıklara benzer şekilde

yorumlanabilir sonuçlar döndüren Softmax'in aksine her sınıf için ayarsız ağ çıkışları hesaplar.

3.2. Eğitim

KSA'larda öğrenme süreci 4 temel adıma ayrılabilir:

- İleri hesaplama,
- Hata/kayıp eniyilemesi,
- Geriye yayılım,
- Parametre güncellemeleri.

3.2.1. İleri Hesaplama

Giriş görüntüsü önce, genel olarak (3.6)'daki ortalama çıkarma ve (3.7)'deki normalizasyon adımını içeren bir ön işlem hattından beslenir.

$$x_i = x_i - x_{ort} \quad (3.6)$$

$$x_i = x_i / \sigma \quad (3.7)$$

Burada; $i = 1, 2, \dots, N$ olmak üzere x_i giriş, x_{ort} ortalama değer ve σ giriş vektörünün standart sapmasıdır. Genellikle ileri hesaplama, konvolüsyon veya öznitelik seçici katmanlarından oluşan alt katmanların, daha yüksekteki tam bağlı katmanlar tarafından takip edildiği durumdur. Ağ, giriş için belirli bir sınıfa ait olma olasılığını kodlayan sınıf çıkışını döndürür. Ağdan gelen çıkışlar, DVM sınıflandırıcısında olduğu gibi ölçeklenmemiş olabilir veya Softmax sınıflandırıcısındaki gibi negatif logaritma olasılığı elde edilebilir. Anlamsal bölütleme için görüntüdeki her bir piksele bir sınıf çıkışı sağlanır.

3.2.2. Kayıp Eniyileme

Ağ tarafından sağlanan sınıf çıkışları seti, ağırlık filtreleri ve biaslar gibi ağda öğrenilen parametrelerin değerlerini ayarlayarak eniyileme işlemine tabi tutulmalıdır. Hangi parametre setinin ideal olduğunu belirlemede oluşan belirsizlik, bir eniyileme problemi olarak formüle edilebilen kayıp fonksiyonuyla nicelleştirilir. Softmax sınıflandırıcısı için sınıf çıkışları s 'nin her bir vektörü için çapraz entropi kaybı (3.8)'de verildiği gibi hesaplanır.

$$L_i = -\log\left(\frac{e^{(s_c)}}{\sum_{i=1}^c e^{(s_i)}}\right) \quad (3.8)$$

Çapraz entropi değerlendirme fonksiyonu, (3.9) ile hesaplanır.

$$H(p, q) = -\sum_x p(x) \log q(x) \quad (3.9)$$

Burada; q , (3.5)'te tanımlanan Softmax fonksiyonunu, p ise gerçek olasılık dağılımı ifade eder.

Toplam kayıp (3.10) ile hesaplanır.

$$L = \sum_{i=1}^N L_i + \lambda R(W) \quad (3.10)$$

Burada; N eğitim örneklerinin sayısı, λ düzenleme katsayısı, L toplam kayıp ve $R(W)$ düzenleme terimidir. L 'yi minimize etmek için, problem bir eniyileme adımı olarak formüle edilip kayıp fonksiyonu minize edilir. Böylece olasılık maksimize edilmiş olur.

3.2.3. Geriye Yayılım

Bölüm 2'de ayrıntılı olarak açıklanan geriye yayılım, sinir ağlarıyla öğrenmede temel bir kavramdır. Geriye yayılımdaki amaç, başlangıçtaki ağırlık parametrelerini periyodik olarak güncellemektir. Problemin geriye yayılımı, maliyet fonksiyonunun eniyilenmesine yardımcı olur.

Makine öğrenmesinde eniyilemeye yönelik olan geriye yayılım yaklaşımı, fonksiyonların birbirine bağlı hesaplama birimlerinin bir grafiği olarak tasarlandığından farklıdır [50]. Bir fonksiyon, ayrı elemanlardan oluşan bir grafikte değerlendirilen dinamik bir nesne olarak düşünülür. Bu yorum, geriye yayılım algoritmasının türevlemede zincir kuralına benzer olduğunu görmemizi sağlar.

Eniyileme algoritması genellikle, gradyan inişi ve onun değişik türlerini anlamak amacıyla kullanılır. Gradyan inişinin basit bir uygulaması, yerel en uygunluk çevresinde gidip gelmede sorunlar ile karşılaştığı için derin bir ağda iyi bir şekilde çalışmayabilir. Bu durum, en uygun noktaya ulaşmak için gerekli olan gradyan inişin güncellenmesi işlemine yardımcı olan momentum parametresiyle düzeltilir.

3.3. Hiperparametreler

Sinir ağı mimarisini geliştirmenin önemli bir parçası en uygun hiperparametrelerin seçilmesidir. Hiperparametreler, eğitim sürecinden önce belirli değerlere ayarlanan, probleme ve veri setine göre değişiklik gösteren parametrelerdir. Bu değerleri seçmede farklı yöntemler mevcuttur:

- **El ile:** Genellikle problem hakkındaki var olan bilgileri kullanarak ve parametre değerlerini tahmin ederek belirlenir.

- **Arama algoritmaları ile:** Hiperparametreler için uygun aralıkları belirlemek için bir ızgara arama veya rastgele arama algoritması kullanılabilir. Daha sonra ağ, bu aralıklarda mevcut olan tüm parametrelerin kombinasyonlarını kullanarak birden çok model üzerinde eğitilir.

- **“Hiper” eniyileme ile:** Buradaki amaç, göreve göre modelin performansını eniyileyebilen otomatik bir yaklaşım oluşturmaktır. Eğitim aşamasında, veriyle sinir ağını beslemek için genellikle aşağıda verilen üç yöntemden birine uyulur:

1. *Yığın Gradyan İnişi:* Maliyet fonksiyonunun gradyanı tüm veri seti üzerinden hesaplanır.

2. *Küçük Yığın Gradyan İnişi:* Eğitim veri setinin bir alt kümesi (küçük yığın) ağ içine yerleştirilir ve bu gibi her küçük yığın için güncellemeler yapılır.

3. *Olasılıksal Gradyan İnişi:* Parametre güncellemeleri her eğitim örneği için gerçekleştirilir. Derin öğrenme modellerinde genellikle varsayılan olarak kullanılan eniyileme algoritmasıdır.

Bengio yaptığı çalışmada, model performansı için birçok araştırmacı tarafından benimsenen en etkili hiperparametrelerin bir listesini önermiştir [51]. Bütünlük oluşturmaları açısından bunlardan bazılarını aşağıda yer verilmiştir.

3.3.1. Öğrenme Oranı

Öğrenme oranı, bir ağı parametrelerini ne kadar hızlı güncellediğini ifade etmektedir. Öğrenme oranı η , çok küçük olduğunda model yakınsaması çok uzun zaman alır. Öte yandan η , çok büyük olur ise model ıraksayarak sapar ve kayıp süresiz olarak azalır çoğalır. En uygun öğrenmeyi sağlamak için başlangıç öğrenme oranı η_0 (genel olarak 0.01 standart kabul edilir) tanımlanır. Ardından bu oran, küçük yığın boyutu ve iterasyon sayısına bağlı olarak periyodik bir şekilde bozunum katsayısıyla ölçeklenerek güncellenir.

3.3.2. Küçük Yığın Boyutu

Derin öğrenme uygulamalarında, veri setinde bulunan tüm verilerin aynı anda işlenerek öğrenilmesi, bellek ve zaman açısından maliyetli bir işittir. Çünkü öğrenmenin her iterasyonunda geriye yayılım işlemi ile ağı üzerinde gradyan inişi hesaplanarak ağırlıklar güncellenmektedir. Bu hesaplama işleminde veri sayısı ne kadar fazla ise hesaplama da o oranda fazla sürmektedir. Bunun için veri seti küçük yığınlara ayrılmakta ve öğrenme işlemi bu küçük yığınlar üzerinde gerçekleştirilmektedir.

Küçük yığın boyutu, parametre güncellemelerinin gerçekleştiği ağı verilen alt örneklerin sayısıdır. Bu sayı, mevcut hesaplama gücüne dayalı olarak seçilir.

3.3.3. Ağırlık Başlatma

Eğitim algoritmasıyla ulaşılan yerel minimum, ağırlık matrisleri için kullanılan başlatma düzenine oldukça bağımlıdır [52]. Çoğu kez, biaslar 0'a ayarlanabilirken, ağırlıklar rastgele bir sıfır ortalamalı dağılımda değişmek üzere başlatılır.

3.3.4. Düzenleştirme Değeri

Bir geçerlilik seti, Dropout olasılığı için düzenleştirme katsayısı λ ve p 'nin ayarlanmasında yararlı olabilir. λ , eğitim sırasında geçerlilik seti üzerindeki model değerlendirilerek ayarlanır. Genellikle λ , kayıp fonksiyonuna bağımlıdır ve 10^{-3} ile 10^{-4} arasında değişir [51]. Dropout, yeterince etkili olduğu ispatlanmış makul bir varsayılan değer olan 0.5 kabul edilebilir, ancak farklı kullanıldığı durumlar da yaygındır [49].

3.3.5. Momentum Katsayısı

Momentum, ağıın eğitimi sürecinde önceki iterasyonların bilgisi ile bir sonraki iterasyonun yönünü bilmeye yardımcı olur. Momentum katsayısı ile ağıın öğrenmesi esnasında yerel bir en iyi noktaya takılıp kalmaması için ağırlık değişim değerinin belirli bir oranda bir sonraki değişime eklenmesi sağlanır. Ağıın yerel çözümler arasında dolaşmasıyla oluşan salınımların önlenmesine yardımcı olur. Momentum katsayısının çok küçük seçilmesi yerel çözümlerden kurtulmayı zorlaştırırken çok büyük seçilmesi ise tek bir çözüme ulaşmada sorunlar yaratabilmektedir. Genel olarak momentum katsayısı, 0.5 ile 0.9 arasında seçilir.

3.3.6. Eğitim Devir Sayısı

Tüm eğitim verisiyle bir kez ağıın eğitimi yapıldıktan sonra bir devir (epoch) tamamlanır. Eğitim devir sayısı, eğitim sırasında tüm eğitim verilerinin ağına gösterilme sayısıdır. Derin öğrenme problemlerinde en uygun ağırlık değerleri adım adım hesaplandığı için ilk devirle başarımlar düşük olacak, devir sayısı arttıkça başarımlar da artacaktır. Ancak, belli bir adımdan sonra modelin öğrenme durumu oldukça azalacaktır. Eğitim doğruluğu artsa bile geçerlilik doğruluğu azalmaya başlayana kadar devir sayısı arttırılmalıdır. Geçerlilik doğruluğunun en yüksek olduğu yerdeki devir sayısı, seçilmesi gereken ideal değeri gösterir.

3.4. Bilinen Bazı KSA Mimarileri

En çok bilinen KSA mimarileri olarak LeNet [43], AlexNet [47], VGGNet [11], GoogleNet [53] ve ResNet [54]'tir. Bir sonraki bölümde değinilecek olan TKA mimarileri için temel teşkil eden AlexNet ve VGGNet'in 16 katmanlı (konvolüsyon+tam bağlı katman sayısı) sürümü olan VGG-16 mimarileri ile ilgili detaylı bilgiler aşağıda verilmiştir:

Alex Krizhevsky, Ilya Sutskever ve Geoff Hinton tarafından geliştirilen AlexNet, bilgisayarla görmede KSA'yı popüler yapan ilk çalışmadır [47]. AlexNet, 2012'deki ILSVRC'de sunulmuş olup yarışmada ikinci ve üçüncü olan mimariden önemli ölçüde daha iyi performans göstererek birinci olmuştur. AlexNet, LeNet'e benzer bir mimariye sahip olmasına rağmen daha derin ve özellikli konvolüsyon katmanları birbirinin üzerine yığılmıştır. AlexNet, 1000 farklı sınıf içeren yüksek çözünürlüklü 1 milyondan fazla görüntü üzerinde eğitilmiştir.

AlexNet'in mimarisi Tablo 3.1'de verilmiştir.

Tablo 3.1. AlexNet'in mimarisi

Katman	Filtre Boyutu	Filtre (Öznitelik Haritası) Sayısı	Kaydırma, Doldurma	Görüntü Boyutu
Giriş	-	-	-	3×227×227
Konvolüsyon	11×11	96	4, 0	96×55×55
DDB	-	-	-	96×55×55
Normalizasyon (LRN)	-	-	-	96×55×55
Öznitelik Seçici (Maksimum)	3×3	-	2, 0	96×27×27
Konvolüsyon	5×5	256	1, 2	256×27×27
DDB	-	-	-	256×27×27
Normalizasyon (LRN)	-	-	-	256×27×27
Öznitelik Seçici (Maksimum)	3×3	-	2, 0	256×13×13
Konvolüsyon	3×3	384	1, 1	384×13×13
DDB	-	-	-	384×13×13
Konvolüsyon	3×3	384	1, 1	384×13×13
DDB	-	-	-	384×13×13
Konvolüsyon	3×3	256	1, 1	256×13×13
DDB	-	-	-	256×13×13
Öznitelik Seçici (Maksimum)	3×3	-	2, 0	256×6×6

Tam Bağlı	-	4096	-	4096×1×1
DDB	-	-	-	4096×1×1
Düzenleştirme (Dropout)	-	-	-	4096×1×1
Tam Bağlı	-	4096	-	4096×1×1
DDB	-	-	-	4096×1×1
Düzenleştirme (Dropout)	-	-	-	4096×1×1
Tam Bağlı	-	1000	-	1000×1×1
Softmax	-	-	-	1000×1×1

Simonyan ve Zisserman tarafından geliştirilen VGGNet'in, VGG-16 ve VGG-19 olarak isimlendirilen iki sürümü mevcuttur. VGG-16 mimarisi, GoogleNet'in birinci olduğu 2014'teki ILSVRC'de ikinci olmuştur. Şu sıralarda ResNet çok gelişmiş bir KSA modeli olarak görülse de VGG-16 basit mimarisinden dolayı daha fazla tercih edilmektedir. AlexNet gibi VGG-16 da, 1000 farklı sınıf içeren yüksek çözünürlüklü 1 milyondan fazla görüntü üzerinde eğitilmiştir.

VGG-16'nın mimarisi Tablo 3.2'de verilmiştir.

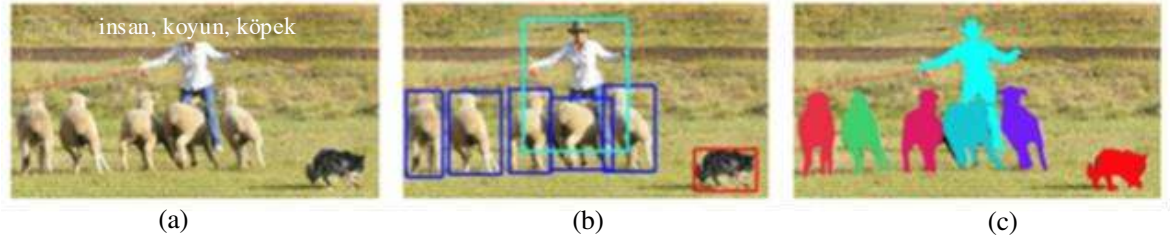
Tablo 3.2. VGG-16'nın mimarisi

Katman	Filtre Boyutu	Filtre (Öznitelik Haritası) Sayısı	Kaydırma, Doldurma	Görüntü Boyutu
Giriş	-	-	-	3×224×224
Konvolüsyon	3×3	64	1, 1	64×224×224
DDB	-	-	-	64×224×224
Konvolüsyon	3×3	64	1, 1	64×224×224
DDB	-	-	-	64×224×224
Öznitelik Seçici (Maksimum)	2×2	-	2, 0	64×112×112
Konvolüsyon	3×3	128	1, 1	128×112×112
DDB	-	-	-	128×112×112
Konvolüsyon	3×3	128	1, 1	128×112×112
DDB	-	-	-	128×112×112
Öznitelik Seçici (Maksimum)	2×2	-	2, 0	128×56×56
Konvolüsyon	3×3	256	1, 1	256×56×56
DDB	-	-	-	256×56×56
Konvolüsyon	3×3	256	1, 1	256×56×56
DDB	-	-	-	256×56×56

Konvolüsyon	3×3	256	1, 1	256×56×56
DDB				256×56×56
Öznitelik Seçici (Maksimum)	2×2	-	2, 0	256×28×28
Konvolüsyon	3×3	512	1, 1	512×28×28
DDB	-	-	-	512×28×28
Konvolüsyon	3×3	512	1, 1	512×28×28
DDB	-	-	-	512×28×28
Konvolüsyon	3×3	512	1, 1	512×28×28
DDB	-	-	-	512×28×28
Öznitelik Seçici (Maksimum)	2×2	-	2, 0	512×14×14
Konvolüsyon	3×3	512	1, 1	512×14×14
DDB	-	-	-	512×14×14
Konvolüsyon	3×3	512	1, 1	512×14×14
DDB	-	-	-	512×14×14
Konvolüsyon	3×3	512	1, 1	512×14×14
DDB	-	-	-	512×14×14
Öznitelik Seçici (Maksimum)	2×2	-	2, 0	512×7×7
Tam Bağlı	-	4096	-	4096×1×1
DDB	-	-	-	4096×1×1
Düzenleştirme (Dropout)	-	-	-	4096×1×1
Tam Bağlı	-	4096	-	4096×1×1
DDB	-	-	-	4096×1×1
Düzenleştirme (Dropout)	-	-	-	4096×1×1
Tam Bağlı	-	1000	-	1000×1×1
Softmax	-	-	-	1000×1×1

4. ANLAMSAL GÖRÜNTÜ BÖLÜTLEME VE TAM KONVOLÜSYONEL AĞLAR

Son yıllarda KSA’lardaki ilerleme ve çok güçlü hesaplama mimarilerinin ortaya çıkmasıyla birlikte bilgisayarla görme sistemlerinin doğrulukları ve yetenekleri hızla artmaktadır. Görüntü sınıflandırmada “görüntüdeki ne?”, nesne bulmada “nesneler görüntünün neresinde?” sorularını cevaplama çok iyi gelişmelere şahitlik edilmiştir. Ancak bunların yanında, görüntüdeki nesneleri bölütleyerek tanımlayan yeni uygulamalara olanak sağlayacak teknikler tasarlanmıştır. Şekil 4.1’de sınıflandırma, bulma ve bölütleme işlemleri örnek bir görüntü üzerinde gösterilmiştir.



Şekil 4.1. (a) Sınıflandırma, (b) Bulma, (c) Bölütleme [55]

Görüntüde nelerin olduğu dışında, konumlarını da öğrenmemizi gerektiren pek çok durum vardır. Görüntüdeki nesnelerin konumlarının belirlenmesi işlemi, bulma (kutu içerisine alma) veya bölütlemeyle gerçekleştirilmektedir.

Bölütleme, birçok görsel çözümün temelinde bulunması gereken bir aşamadır. Görüntü bölütleme, görüntüyü aralarında benzerlik gösteren belirli gruplara ayırma işlemidir. Görüntü bölütleme işleminin başarılı bir şekilde gerçekleşmiş kabul edilebilmesi için nesnelerin görüntüdeki diğer bileşenlerden bağımsız olması, ayrıca aynı doku ve renk özelliklerine sahip bölgelerin oluşturulması beklenir. Anlamsal görüntü bölütleme ise, bir görüntüdeki her bir pikseli verilen bir anlamsal sınıfa ait olarak etiketleme işlemidir.

Anlamsal görüntü bölütlemenin uygulama alanı çok geniştir. Tıbbi görüntü işlemeden otonom araçlara, sanayide kalite kontrolden e-ticaret ve uzaktan algılamaya kadar pek çok alanda kullanılabilir [56].

Daha önce de bahsedildiği gibi sınıflandırma ağları olan AlexNet ve benzeri standart KSA’lar uzaysal olmayan tahmini gerçekleştirmektedir. Görüntü sınıflandırma işleminde ağın çıktısı sınıf olasılıklarının tek bir dağılımıdır. Yani, 0 ile 1 arasındaki bir sayı her sınıf

için bir olasılığı verir. Sınıflandırma ağlarının bir diğer kısıtı da görüntülerdeki nesnenin konumunu söyleyememesidir. Bunu yapmak için eğitilmemiş oldukları düşünüldüğünde bu durum anlaşılabilir. Ancak bu durum görüntü bölütleme konusunda büyük bir engel teşkil eder. Görüntü bölütlemeye bütün bir görüntü için tek bir olasılık dağılımı tahmin etmek yerine görüntü bir kaç bloğa bölünür ve her bloğa kendi olasılık dağılımı atanır. Çok yaygın olarak görüntüler piksel seviyesine bölünür ve her piksel sınıflandırılır. Yani görüntüdeki her bir piksel için ağ, o pikselin hangi sınıfa ait olduğunu tahmin etmek üzere eğitilir. Bu, ağın görüntüdeki bir kaç nesne sınıfını tanımlamakla kalmaz aynı zamanda nesnelerin konumunu belirlemeye de olanak tanır.

Anlamsal görüntü bölütleme için kullanılan veri setleri bölütlenecek görüntüler ve genellikle bu görüntülerle aynı boyuttaki etiket görüntülerinden oluşur. Etiket görüntüsü, görüntünün kesin referans sınırlarını gösterir. Etiket görüntülerindeki şekiller, her nesnenin sınıfını temsil etmek üzere renklerle kodlanmıştır.

Şekil 4.2’de iki uçak ve iki insanın bulunduğu örnek bir bölütlenecek görüntü ve bu görüntünün etiket görüntüsü verilmiştir.



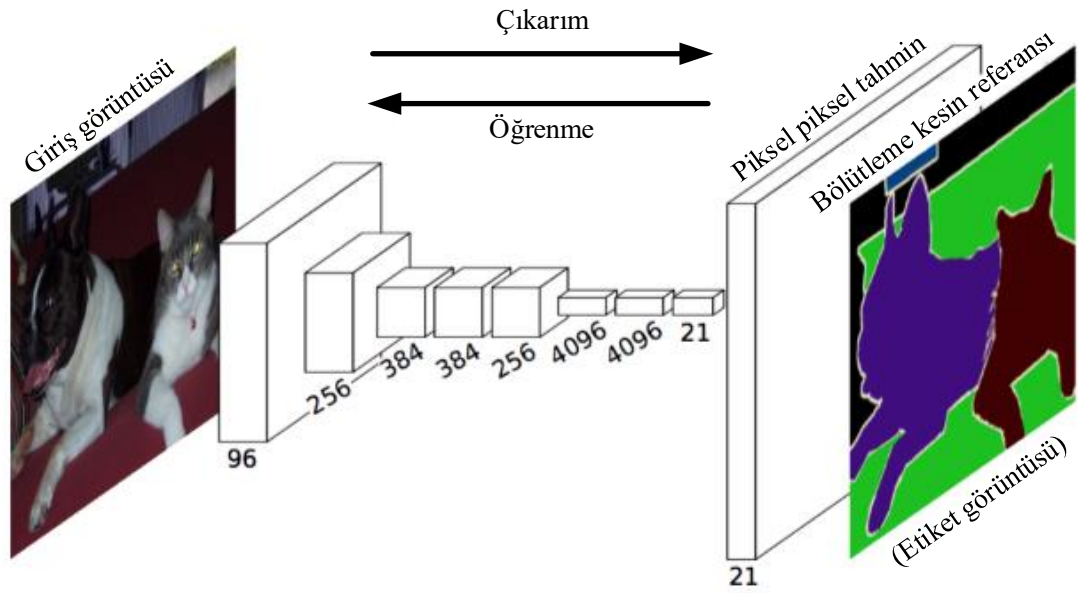
Şekil 4.2. Örnek bir bölütlenecek görüntü-etiket görüntüsü çifti: (a) Bölütlenecek görüntü, (b) Etiket görüntüsü [55]

Şekil 4.2’deki örnek görüntüler PASCAL VOC [57] veri setinden olup, bu veri setindeki etiket görüntüleri 8 bitlik tek kanallı, paletlenmiş bir görüntüdür. 1’den 20’ye kadar olan indeksler uçak, kuş ve insan gibi nesne sınıflarını temsil ederken indeks 0 (siyah piksel) arka planı, indeks 255 (beyaz piksel) ise eğitim sırasında nesne sınırları gibi yok sayılacak alanları temsil eder.

Görüntü bölütleme gibi yoğun bir tahminin gerçekleştirilmesi için KSA, TKA’ya dönüştürülmelidir. Çünkü KSA’ların tam bağlı katmanları konuma ait bilgileri atmaktadır.

Bu yüzden bir anlamsal görüntü bölütme uygulamasının gerçekleştirilebilmesi için KSA'ların tam bağı katmanlarının konvolüsyonelleştirilmesi gerekmektedir.

TKA'ya geçişte KSA'nın konvolüsyon kısmı tamamen yeniden kullanılabilir. Mevcut KSA'ların tam konvolüsyonel sürümleri, keyfi boyutlu girişlerden verimli bir çıkarım ve öğrenmeyle yoğun çıkışlar tahmin etmektedir. Hem öğrenme hem de çıkarım, Şekil 4.3'te gösterildiği üzere yoğun ileri besleme hesaplama ve geriye yayılım vasıtasıyla tek seferde tüm görseli sunmaktadır. Ağ içi üst örnekleme katmanları, piksel odaklı tahmine ve alt örnekleme aracılığıyla ağılardaki öğrenmeye olanak verir [9].

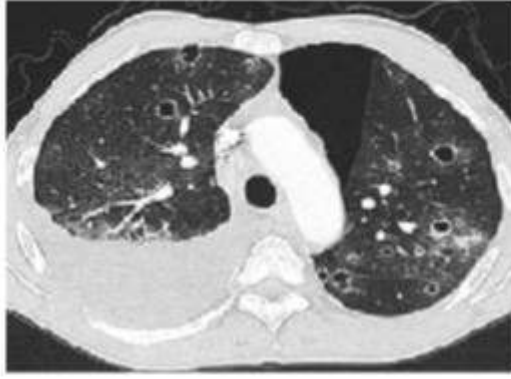


Şekil 4.3. TKA ile piksel piksel tahmin işlemi [9]

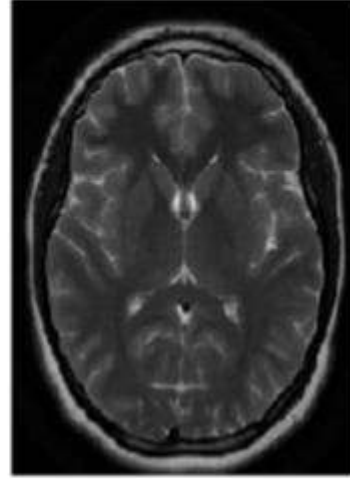
4.1. Tıbbi Görüntü Bölütme

Tıbbi görüntüler, hastanın elektronik sağlık kayıtlarının ayrılmaz bir parçası olup sağlık kuruluşlarının hastalığın tanısına ve hastaların tedavilerine yardımcı olmada önemli bir rol oynamaktadır.

Tıbbi görüntüler, ultrason, X-ışını (X-ray), BT, MR, Pozitron Emisyon Tomografi kısaca PET (Positron Emission Tomography), retinal fotoğrafçılık ve dermoskopi görüntüleme yöntemleriyle elde edilen görüntülerinden oluşmaktadır. BT ve MR gibi görüntüleme yöntemleri çoklu organları incelerken retinal fotoğrafçılık ve dermoskopi yöntemleri organa özgüdür. Şekil 4.4'te bazı örnek tıbbi görüntüler gösterilmektedir.



(a)



(b)



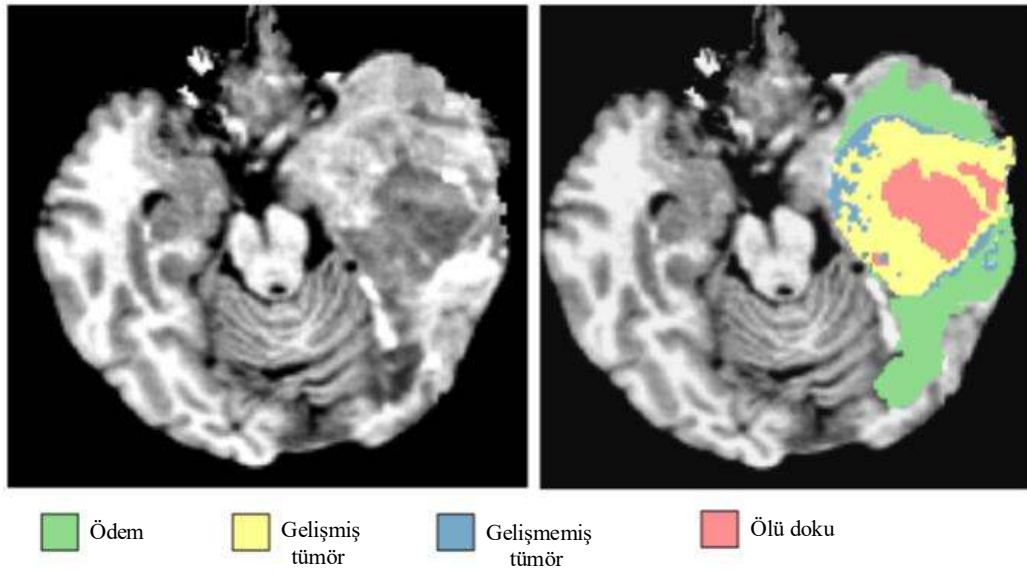
(c)



(d)

Şekil 4.4. Tıbbi görüntü örnekleri: (a) BT akciğer tarama, (b) MR beyin tarama, (c) X-ışını göğüs tarama, (d) Dermoskopi görüntüsü

Şekil 4.5'teki tıbbi görüntü üzerinde gösterildiği gibi bir gruptaki her piksel, bir bütün olarak nesne sınıfına karşılık gelir. Hastalık tanısı için önem teşkil eden tıbbi bir görüntüde bu sınıflar; lezyon veya tümör ile arka plan olabilir. Tıbbi görüntülerin etiket görüntüleri, derin sinir ağının öğrenmesi istenen anatomik öz niteliğin piksel konumunu belirtmektedir.



Şekil 4.5. Örnek bir anlamsal tıbbi görüntü bölütleme işlemi

Daha önce de belirtildiği gibi tez çalışması kapsamında cilt lezyonu bölütleme üzerine bir uygulama gerçekleştirilmiştir. Dermoskopik görüntü analizinde en önemli adımlardan biri, lezyon alanlarının otomatik tespittir. Cilt lezyonlarının düzgün bir şekilde bölütlenmesi, melanomun tespitini kolaylaştırmaktadır.

Derinin yüzeyinde meydana gelen pigmentli lezyonlar olan melanom, en ölümcül cilt kanseri biçimidir ve cilt kanseri ile ilişkili ölümlerin yaklaşık % 75'ini oluşturmaktadır [58]. Melanomun erken bir şekilde saptanması, hastanın hayatta kalma oranını önemli derecede artırır. Çünkü erken tanı melanomların beşte birinden azı metastaz hale gelir [59] ve son derece iyileştirilebilir yapıdadırlar.

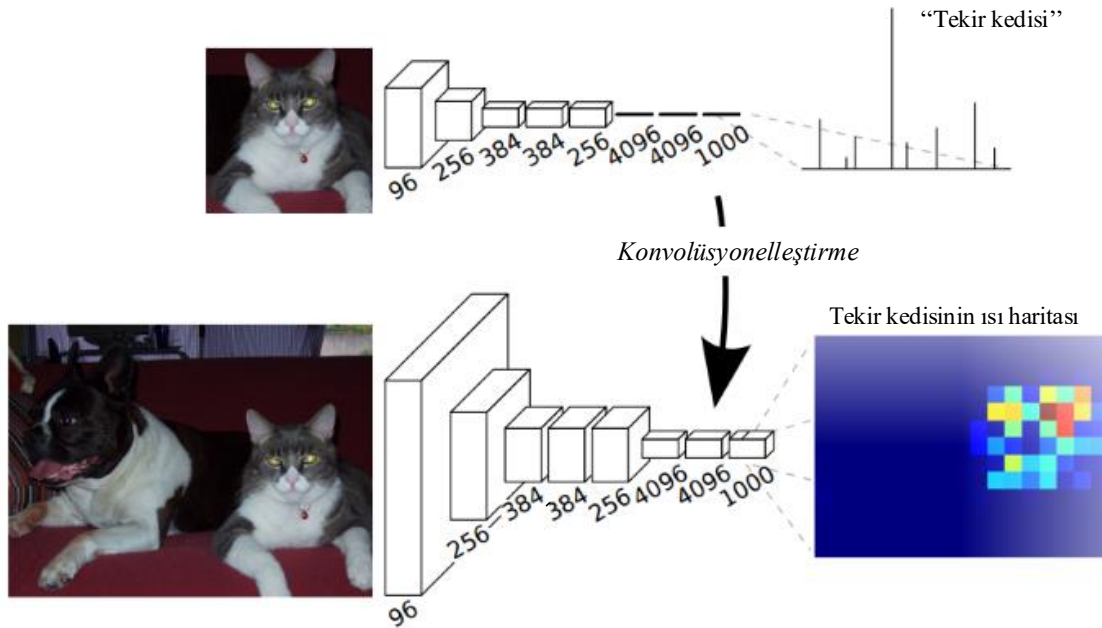
Tanı, ilk görsel tarama, dermoskopik analiz ve ardından biyopsi ve histopatolojik inceleme ile başlar. Melanomun tanısal performansını iyileştirmek için dermoskopi tekniği geliştirilmiştir. Dermoskopi, lezyonların netliğini artırmak için cilt bölgesinin büyütülmüş ve aydınlatılmış bir görüntüsünün elde edilmesini sağlayan bir cilt görüntüleme tekniğidir, bu da yüzey lezyonunu görselleştirerek cilt lezyonunun görsel etkisini artırır. Dermoskopi görüntülerinin görsel olarak incelenmesi, melanomun erken tanısında önemli bir rol oynamaktadır. Ancak, insanın görsel analizi öznel olabilmekte olup deneyimli dermatologlar ve radyologlar tarafından bile bazen yanlış kararlar verilebilmektedir [60]. Tanı ile ilgili karar vermeyi zorlaştıran başlıca üç etmen vardır. Birincisi, cilt lezyonları ve normal cilt bölgesi arasındaki düşük kontrast, lezyon alanlarının doğru şekilde bölütlendirilmesini zorlaştırır. İkincisi, melanom ve melanom dışı lezyonlar, yüksek derecede görsel benzerliğe

sahip olabilir, bu da melanom lezyonunu melanom dışı lezyondan ayırma zorluğuna neden olur. Üçüncüsü, hastalar arasındaki ten rengi, kıl veya damar gibi cilt koşullarının değişmesi renk ve doku bakımından farklı melanom görünimleri oluşturur. Bu nedenlerden dolayı melanomun güvenilir bir şekilde bilgisayar destekli otomatik tespiti, radyologların doğruluğunu ve verimliliğini artırmak için çok yararlıdır.

4.2. Konvolüsyonel Sinir Ağından Tam Konvolüsyonel Ağa Geçiş

Temelde, piksel başına bir olasılık dağılım tahmini yapan anlamsal görüntü bölütleme ve görüntü başına bir olasılık dağılım tahmini yapan görüntü sınıflandırma oldukça benzer durur ve her iki probleme de aynı tekniklerin uygulanması beklenebilir. Görüntü bölütleme, görüntü sınıflandırma problemine yalnızca uzaysal bir boyut katar. Bu yüzden, bir kaç küçük ayarlama, bir sınıflandırma sinir ağını anlamsal bölütleme sinir ağına dönüştürmeye yeterlidir. Bu bölümde, anlamsal görüntü bölütleme problemi için gereken TKA mimarisine geçiş, [9]’daki yöntemler kullanılarak AlexNet ve VGG-16 mimarileri üzerinden anlatılmıştır.

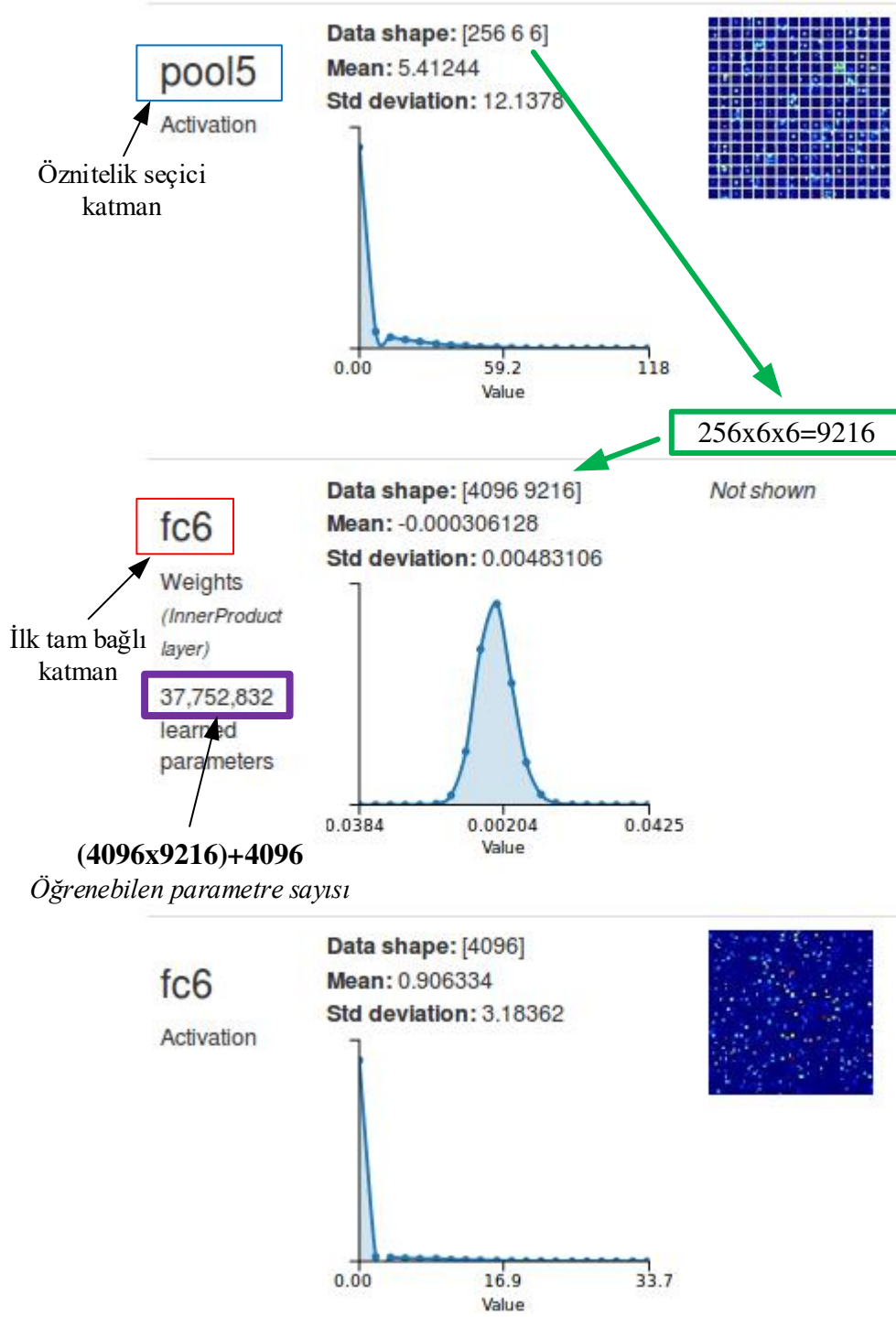
TKA, KSA’nın tam bağlı katmanlarını konvolüsyon katmanlarına dönüştürerek elde edilmektedir. Bu işlem, Şekil 4.6’da gösterilmiştir.



Şekil 4.6. KSA’nın tam bağlı katmanlarını konvolüsyon katmanlarına dönüştürme [9]

Konvolüsyonelleştirilmiş modellerin uzaysal çıkış haritaları, onları anlamsal görüntü bölütleme gibi yoğun problemler için vazgeçilmez bir seçenek haline getirir. Her çıkış nöronunda mevcut olan kesin referansla hem ileri hem de geri geçişler basittir ve her ikisi de konvolüsyonun özünde var olan hesapsal verimlilikten faydalanır.

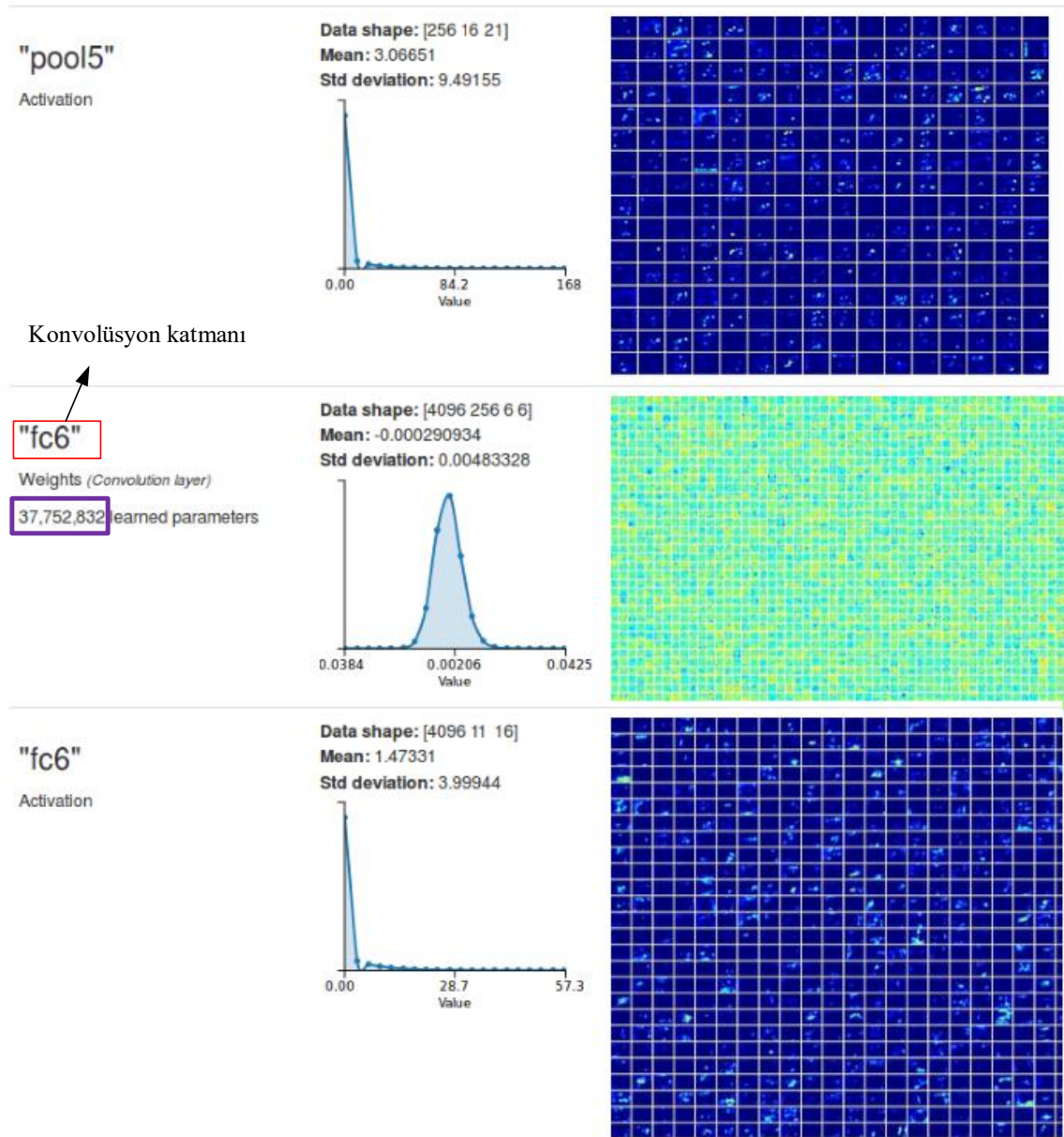
KSA'nın tam bağlı katmanlarını ortadan kaldırarak görünürde daha güçlü bir modelin nasıl oluşturulabileceğini anlamak için tam bağlı bir katman ile bir konvolüsyon katmanı arasındaki farkın bilinmesi gerekmektedir. Tam bağlı bir katmanda, her çıkış nöronu girişteki değerlerin ağırlıklı toplamını hesaplar. Bir konvolüsyon katmanında, her filtre alıcı alandaki değerlerin ağırlıklı toplamını hesaplar. Bu işlemler tam olarak aynı şeylermiş gibi görünse de sadece katman girişinin alıcı alanla aynı boyuta sahip olduğu durumlarda aynıdır. Eğer giriş alıcı alandan büyükse, o zaman konvolüsyon katmanı giriş penceresini kaydırır ve başka bir ağırlıklı toplam hesaplar. Bu işlem, giriş görüntüsü soldan sağa, yukarıdan aşağı doğru taranana kadar tekrar eder. Eninde sonunda, her filtre bir aktivasyon matrisi yani öznitelik haritası üretir. Tam bağlı bir katman, eşdeğer bir konvolüsyon katman ile değiştirilip filtrelerin boyutu katmanın giriş boyutuna ayarlanmalı ve tam bağlı katmandaki nöronlar kadar çok filtre kullanılmalıdır. Bu işlemlerin AlexNet'in ilk tam bağlı katmanında nasıl gerçekleştirilmesi gerektiği aşağıda detaylı bir şekilde açıklanmıştır. İlgili katmanların DIGITS'te görselleştirilmesi Şekil 4.7'de gösterildiği gibidir.



Şekil 4.7. AlexNet'in ilk tam bağlı katmanının girişi, ağırlıkları ve aktivasyonları

Şekil 4.7'de, fc6'nın girişini pool5'ten aldığı ve giriş biçiminin 256 kanallı 6×6 boyutunda bir görüntü olduğu görülmektedir. Ayrıca, fc6'daki aktivasyonların 4096 elemanlı bir vektör olduğu böylece fc6'nın 4096 adet çıkış nöronuna sahip olduğu da

görülmektedir. fc6, eşdeğer bir konvolüsyon katmanı ile değiştirilmek istenirse filtre boyutu 6×6 'ya, çıkış öznitelik haritalarının sayısı da 4096'ya ayarlanmak zorundadır. Her filtre için alıcı alanda sayı başına bir bias terimi ve bir ağırlık vardır. Alıcı alanın derinliği 256, boyutu 6×6 olduğundan filtre başına $(256 \times 6 \times 6) + 1 = 9217$ parametre vardır. 4096 adet filtre olduğundan oluşturulan bu konvolüsyon katmanı için toplam parametre sayısı 37752832'dir. Dikkat edilirse bu değer, Şekil 4.7'de belirtildiği gibi tam bağlı katmandaki toplam öğrenilebilir parametre sayısı ile eşittir. Konvolüsyonelleştirilen fc6 katmanının DIGITS'te görselleştirilmesi Şekil 4.8'de verilmiştir.



Şekil 4.8. AlexNet'in ilk konvolüsyonelleştirilen katmanının girişi, ağırlıkları ve aktivasyonları

Bu bilgilerle AlexNet mimarisindeki tüm bağı katmanlar, eşdeğer konvolüsyon katmanlarına dönüştürülerek TKA mimarisi elde edilebilir. Bu TKA, temel KSA gibi aynı sayıda öğrenebilen parametreye ve aynı hesapsal karmaşıklığa sahiptir. Temel KSA'nın konvolüsyonelleştirilmesi TKA'ya geçişte bir hayli esneklik sağlamaktadır. Artık TKA modeli, AlexNet'te olduğu gibi sabit giriş boyutu (224×224) ile çalışacak şekilde sınırlandırılmamıştır. TKA, sürgülü bir pencereymiş gibi girişi başından sonuna kadar tarayarak büyük görüntüleri işleyebilir ve tüm giriş için tek bir olasılık dağılımı üretmek yerine model, 224×224 'lük pencere başına bir tane üretir. Böylece, ağı çıkışı $N \times H \times W$ biçiminde bir tensör haline gelmiştir.

Burada; N sınıf sayısı, H dikey eksen boyunca kayan pencerelerin (filtre) sayısı ve W ise yatay eksen boyunca kayan pencerelerin sayısıdır. Özet olarak, sınıflandırma ağına çıkışına iki uzaysal boyut ekleyerek TKA'nın tasarımındaki ilk önemli aşama tamamlanmış olur.

Pencere başına bir sınıf olasılık dağılımı üreten bir TKA'nın tasarımı sırasında anlaşılabileceği üzere, pencere sayısı giriş görüntüsünün boyutuna, pencerenin boyutuna ve giriş görüntüsü taranırken pencereler arasında kullanılan kaydırma parametresine bağlıdır. İdeal olarak bir anlamsal görüntü bölütleme modeli, görüntüdeki piksel başına bir olasılık dağılımı üretmelidir. Giriş görüntüsü, konvolüsyonelleştirilmiş AlexNet'in sıralı katmanlarından geçtiği zaman girişteki piksel verisi etkili bir şekilde daha kaba ve daha yüksek seviyede öznitelik temsillerinin içine sıkıştırılır. Anlamsal görüntü bölütlemedeki amaç, girişteki her piksel için ince ayarlı bir sınıflandırmayı yeniden oluşturmak için bu kaba öznitelikleri ara değerlemektir. Bu işlem, “ters konvolüsyon” katmanları ile kolaylıkla yapılabilir. Ters konvolüsyon katmanları, konvolüsyonel karşılıklarının tersi işlemini yerine getirirler. Konvolüsyon katmanının çıkışı göz önünde bulundurulursa ters konvolüsyon katman, çıkışı üreten girişi bulur. Hatırlanacağı üzere, konvolüsyon veya öznitelik seçici katmandaki kaydırma parametresi, giriş işlenirken pencerenin ne kadar kaydırılacağını tanımlar ve buna bağlı olarak çıkışın nasıl alt örnekleneyeceğinin bir ölçüsüdür. Buna karşılık, ters konvolüsyon katmanındaki kaydırma parametresi, çıkışın nasıl üst örnekleneyeceğinin bir ölçüsüdür. Ters konvolüsyon katmanının çıkış hacmi olan $D_o \times H_o \times W_o$, (4.1), (4.2) ve (4.3) eşitlikleri kullanılarak hesaplanır.

$$D_o = N \quad (4.1)$$

$$H_o = s(H_i - 1) + f - 2p \quad (4.2)$$

$$W_o = s(W_i - 1) + f - 2p \quad (4.3)$$

Burada; s kaydırmayı, p doldurmayı, f filtre boyutunu, H_i ve W_i giriş boyutlarını ve N ise kanal sayısını ifade etmektedir.

TKA mimarisindeki son konvolüsyon katmanının aktivasyonlarını, giriş görüntüsüyle aynı boyutta bir çıkış elde etmek için ne kadar üst örnekleme gerektığının bilinmesi gerekir. Her katmanın incelenmesi ve ölçekleme katsayısının dikkatlice not edilmesi şarttır. Ölçekleme katsayısı, $1/s$ ile ifade edilir. Tablo 3.1’de, katman katman verilen AlexNet mimarisi göz önüne alındığında ölçekleme katsayılarının tamamı çarpıldığında kümülatif ölçekleme katsayısının $1/32$ olduğu sonucuna ulaşılır. Bu yüzden, ters konvolüsyon katmanı için gereken kaydırma parametresi 32’dir.

s kaydırmalı bir konvolüsyon katmanının tüm uzaysal boyutlarda giriş boyutunun $1/s$ katında bir çıkış verdiğini söylemek her zaman doğru olmaz. Çünkü, girişe $p > 0$ olan doldurma eklemek, aktivasyonların sayısını arttıracaktır. Aksine, $f > 1$ boyutlu filtreler eklemek de girişteki aktivasyonları azaltacaktır. Gerçekte, her konvolüsyon veya öznitelik seçici katmanın çıkışı $(p - (f - 1)/2)/s$ ile kaydırılır. Bunun sonucunda ilave piksellerden dolayı çıkış boyutu, hiç doldurma olmaması durumundan daha büyük olur. Ters konvolüsyon katmanının çıkışı ise $(f - 1)/2 - p$ ile kaydırılır. İlk katmandan son katman olan ters konvolüsyon katmanı boyunca bu işlemler kümülatif olarak gerçekleştirilir ve toplamda oluşan ilave pikseller Softmax katmanından önce kırılarak giriş boyutuna eşitlenir.

Tüm bu işlemler sonucunda elde edilen TKA mimarisi FCN-AlexNet olarak isimlendirilmiştir. 335×500 çözünürlüklü örnek RGB görüntüler için FCN-AlexNet mimarisi Tablo 4.1’de verilmiştir.

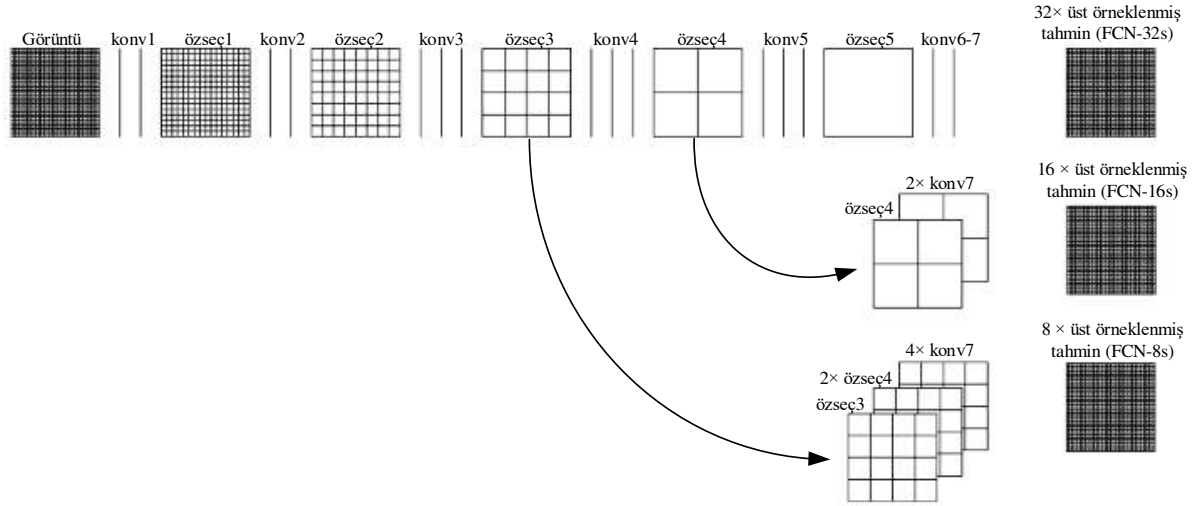
Tablo 4.1. FCN-AlexNet mimarisi

Katman	Filtre Boyutu	Filtre (Öznitelik Haritası) Sayısı	Kaydırma, Doldurma	Görüntü Boyutu
Giriş	-	-	-	$3 \times 335 \times 500$
Konvolüsyon	11×11	96	4, 100	$96 \times 132 \times 173$
DDB	-	-	-	$96 \times 132 \times 173$
Öznitelik Seçici (Maksimum)	3×3	-	2, 0	$96 \times 66 \times 86$
Normalizasyon (LRN)	-	-	-	$96 \times 66 \times 86$

Konvolüsyon	5×5	256	1, 2	256×66×86
DDB	-	-	-	256×66×86
Öznitelik Seçici (Maksimum)	3×3	-	2, 0	256×33×43
Normalizasyon (LRN)	-	-	-	256×33×43
Konvolüsyon	3×3	384	1, 1	384×33×43
DDB	-	-	-	384×33×43
Konvolüsyon	3×3	384	1, 1	384×33×43
DDB	-	-	-	384×33×43
Konvolüsyon	3×3	256	1, 1	256×33×43
DDB	-	-	-	256×33×43
Öznitelik Seçici (Maksimum)	3×3	-	2, 0	256×16×21
Konvolüsyon	6×6	4096	1, 0	4096×11×16
DDB	-	-	-	4096×11×16
Düzenleştirme (Dropout)	-	-	-	4096×11×16
Konvolüsyon	1×1	4096	1, 0	4096×11×16
DDB	-	-	-	4096×11×16
Düzenleştirme (Dropout)	-	-	-	4096×11×16
Konvolüsyon	1×1	21	1, 0	21×11×16
Ters Konvolüsyon	63×63	21	32, 0	21×383×543
Kırpma	-	-	-	21×335×500
Softmax	-	-	-	21×335×500

Kaydırmayla meydana gelen ilave 31 piksel dikkate alınarak Tablo 4.1 incelendiğinde, FCN-AlexNet'i oluşturmak için eklenen üst örnekleme (ters konvolüsyon) katmanının bir önceki konvolüsyon katmanının çıkışını 32 kat arttırdığı görülmektedir. Bu da uygulamada, ağın 32×32 piksel bloğu başına tek bir tahmin yaptığı anlamına gelir. Bu durum, görüntüdeki nesnelerin dış hatlarının kaba olacak şekilde bölütlenmesine neden olur. [9]'daki makale, bu kısıtlama için sekme mimarisi fikrini sunmaktadır. Bu mimarideki sekme bağlantıları, Şekil 4.9'da gösterildiği gibi VGG-16'dan türetilen TKA mimarisinin özseç3 ve özseç4 öznitelik seçici katmanlarının çıkışlarını doğrudan ağın çıkışına yeniden yönlendirebilmek için eklenmiştir. Bu öznitelik seçici katmanlar, ağda daha geride olduğu için alt seviye öznitelikler üzerinde çalışırlar ve daha ince ayrıntıları yakalayabilir.

[9]'da önerilen TKA mimarileri, VGG-16'daki tam bağlı katmanların eş değer konvolüsyon katmanlarına dönüştürülüp sekme bağlantılarının uygulanışına göre FCN-8s, FCN-16s ve FCN-32s olarak isimlendirilmiştir. Bu mimarilerin görselleştirilmesine Şekil 4.9'da yer verilmiştir.



Şekil 4.9. FCN-32s, FCN-16s ve FCN-8s mimarilerinin görselleştirilmesi [9]

Şekil 4.9'da, öznetelik seçici katmanlar uzaysal yoğunluğu ifade eden ızgaralar şeklinde gösterilirken konvolüsyon katmanları ise dikey çizgilerle gösterilmiştir. Şekil 4.9'da görüldüğü gibi FCN-32s'nin son çıkışı, sekme bağlantısı olmaksızın tek bir seferde olacak şekilde tahminler piksellere 32 kaydırma ile üst örnekleme yapılır. FCN-16s'nin son çıkışı, hem konv7 hem de özseç4 katmanından 16 kaydırma ile birleştirilir ve yüksek seviyeli anlamsal bilgi tespit ederken daha ince detayları tahmin etmesine izin verir. FCN-8s'nin son çıkışı ise, tahminlere özseç3 katmanını da ilave ederek 8 kaydırma ile daha keskin tahminler sağlar. Bu sayede FCN-8s mimarisi, 8x8 piksel bloklarına kadar daha ince bir ayar da tahminler yapabilmeye olanak sağlamaktadır.

5. DENEYSEL ÇALIŞMALAR

Bu çalışmada, TKA'ların tıbbi görüntü bölütlemedeki performansını gözlemlemek amacıyla melanomun tespitini kolaylaştıracak otomatik bir cilt lezyonu bölütleme uygulaması gerçekleştirilmiştir. Uygulama için Bölüm 4'te anlatılan 4 farklı TKA mimarisinden yararlanılmıştır: FCN-AlexNet, FCN-8s, FCN-16s ve FCN-32s.

Uygulama, DIGITS ortamında Caffe kütüphanesi vasıtasıyla gerçekleştirilmiş olup deneysel çalışmalar için kullanılan TKA mimarilerinin bölütleme performansları karşılaştırılmıştır. Eğitim ve geçerlilik görüntüleri için yapılan ön işlemler MATLAB paket programı üzerinde gerçekleştirilmiştir. Uygulamada veri seti olarak ISIC 2017 veri seti kullanılmıştır. Çalışmalar, 4. nesil Intel i5 3.4 GHz işlemci, 8 GB RAM ve NVIDIA GTX Titan X Pascal ekran kartına sahip bir masaüstü bilgisayar üzerinde gerçekleştirilmiştir. Kullanılan ekran kartının CUDA desteği sayesinde uygulama için gerekli hesaplamalarda GPU tabanlı paralel hesaplama gücünden yararlanılmıştır.

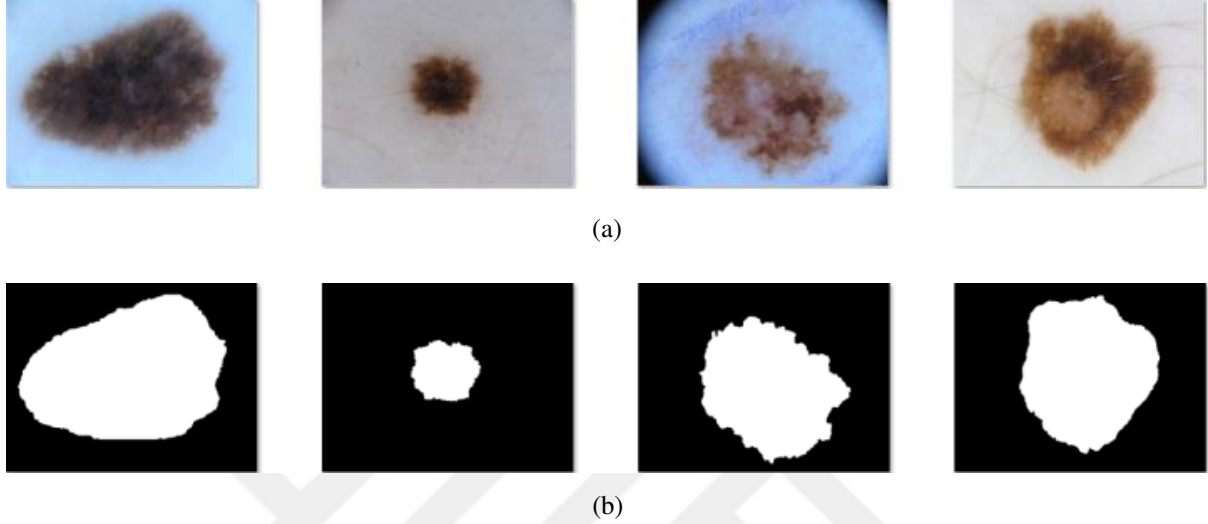
5.1. Veri Seti

Kısaca ISIC olarak isimlendirilen Uluslararası Cilt Görüntüleme İşbirliği, cilt lezyonunun otomatik analizine odaklanan bir işbirliğidir. 2016'da cilt lezyonun analizi için Biyomedikal Görüntülemede Uluslararası Sempozyum kısaca ISBI (International Symposium on Biomedical Imaging)'de başlattıkları yarışmadan bu yana veri setlerini sürekli genişletmiştir. Yarışmanın amacı, otomatik melanom tanı algoritmalarının gelişimini desteklemek için sabit veri seti görüntüleri sağlamaktır.

2017'de ISBI, bu yarışmanın ikincisine ev sahipliği yapmış olup veri seti şu anda uluslararası klinik merkezlerinden çeşitli cihazlardan elde edilen 20 binden fazla görüntü içermektedir [61]. Açıklamalı veri setleri, lezyon bölütleme, dermoskopik öznitelik çıkarımı ve lezyon sınıflandırma da dahil olmak üzere cilt lezyon görüntüleriyle ilgili üç görev için açık erişime sunulmuştur.

ISIC 2017 veri seti [62], bölütleme için kesin referansları, dermoskopik öznitelik çıkarımı için süper piksel maskeleri ve sınıflandırma için ek açıklamalar içeren 1504×1129, 1022×767, 944×1127, 767×576 gibi farklı çözünürlüklerde cilt lezyon görüntüleri içermektedir. Eğitim veri seti 2000, geçerlilik veri seti 150 ve test veri seti 600 RGB

görüntüden oluşur. Kesin referansları, lezyon sınırlarını uzmanların elle oluşturduğu lezyon ve arka plan sınıflarından oluşan ikili maskelerdir. Her bir piksel, piksel lezyon ise 1, piksel normal cilt bölgesine aitse 0 olarak etiketlenmiştir. ISIC 2017 veri setinden örnek görüntüler Şekil 5.1’de verilmiştir.



Şekil 5.1. ISIC 2017 veri setinden örnekler: (a) Bölütlenecek görüntüler, (b) Kesin referans görüntüleri

Uygulamada, eğitim veri setinin tamamı ağ modelinin eğitimi için kullanılmış olup test veri setindeki 250 görüntü geçerlilik setine dahil edilerek toplam 400 görüntü ise modelin geçerliliğini doğrulamak için kullanılmıştır. Veri setindeki görüntüler, çok yüksek çözünürlükte olduğu için yüksek hesaplama maliyeti gerektirmektedir. Hesaplama maliyetini azaltmak ve TKA’lar için lezyon görüntüleri ön işlemlere tabi tutulmuştur. Eğitim ve geçerlilik görüntülerinin her birinin çözünürlüğü 250×250 olacak şekilde yeniden boyutlandırılmıştır. Böylece tüm eğitim ve geçerlilik görüntüleri aynı boyuta getirilmiştir. Ayrıca gri seviye olan kesin referans görüntüleri, indeksli görüntülere dönüştürülmüştür.

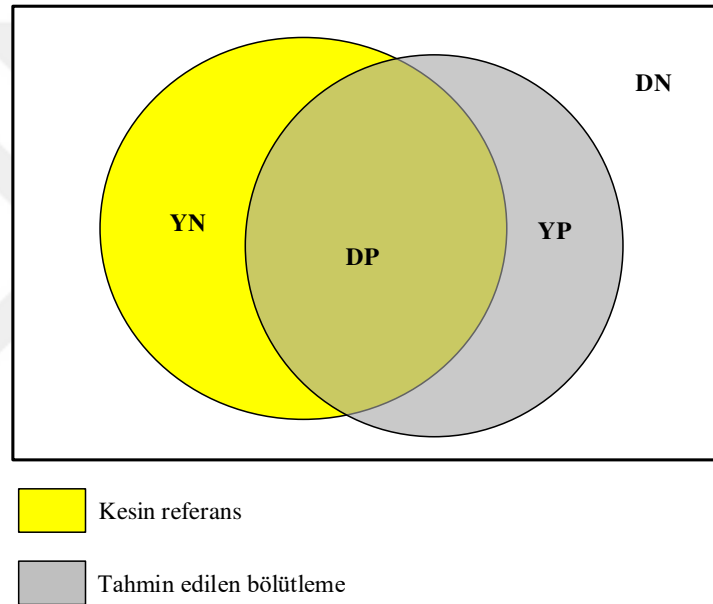
5.2. Performans Değerlendirme Metrikleri

Az sayıda sınıfa sahip bölütleme uygulamalarında derin sinir ağları, “sınıf dengesizliği” olarak bilinen duruma maruz kalabilir. Tez çalışmasında gerçekleştirilen uygulama göz önüne alınırsa cilt lezyonlarını temsil eden piksellerin dermoskopi görüntülerindeki tüm piksellerin yaklaşık %20’sini oluşturduğu varsayıldığında geçerlilik doğruluğunun %80 olduğunu iddia eden bir TKA, her pikseli arka plan olarak sınıflandırıyor olabilir. Buradan anlaşılabilecek üzere doğruluk, otomatik bölütleme performansını ölçmek için kullanılabilecek

en iyi metrik değildir. Bu nedenle lezyon etiketini temsil eden pikseller ile model tarafından öğrenilen pikseller arasında bir benzerlik belirlemek için ISBI 2017 lezyon bölütleme yarışması tarafından önerilen Dice katsayısı [63] metriği de kullanılmıştır.

Dice katsayısı, görüntü bölütleme algoritmasının sonuçlarının buna karşılık gelen kesin referans bölütlemeye ne kadar benzer olduğunu ölçmek için biraz farklı yaklaşımlar kullanır.

Otomatik bölütleme sonucunda, YP (Yanlış Pozitif), YN (Yanlış Negatif), DP (Doğru Pozitif) ve DN (Doğru Negatif) değerleri bölütleme işleminin başarı durumunu göstermektedir. Pozitif ve negatif alanlar, görüntü bölütleme çalışmalarında Şekil 5.2’de ifade edildiği gibidir.



Şekil 5.2. Bölütleme üzerine istatistiksel alanların tanımı

Burada; DP doğru sınıflandırılmış nesnedeki piksel sayısını, DN doğru sınıflandırılmış arka plandaki piksel sayısını, YP nesne olarak sınıflandırılmış arka plana ait piksel sayısını, YN arka plan olarak sınıflandırılmış nesneye ait piksel sayısını ifade etmektedir.

“Kesin referans” etiketli alanın insanlar tarafından açıklanan bölütleme bölgesi olduğu varsayıldığında, “Tahmin edilen bölütleme” etiketli alan derin öğrenme algoritması ile üretilen bölütleme sonucudur. İnsan ve yapay zeka sonuçları arasındaki örtüşme alanı, bir görüntü bölütleme algoritmasının açıklamalı kesin referans bölütlemesiyle tam olarak eşleşen pikselleri tanımladığı bölgedir.

DP, DN, YP ve YN değerleri hata matrisi (confusion matrix) olarak tanımlanan bir tabloya Tablo 5.1’de görüldüğü gibi aktarılır. Bu tablo, iki sınıflı bir veri setinde sınıflandırıcının veri seti üzerinde verdiği tahmin sonuçlarının doğruluğunu barındırır.

Tablo 5.1. Hata matrisi

		Asıl sınıf	
		Pozitif	Negatif
Tahmin edilen sınıf	Pozitif	Doğru Pozitif (DP)	Yanlış Pozitif (YP)
	Negatif	Yanlış Negatif (YN)	Doğru Negatif (DN)

Bölütleme çalışmalarında daha çok pozitif tabanlı metrikler kullanılmaktadır. Çünkü bölütlemelerde sıklıkla örtüşme alanlarının (DP, YP, YN) nitelendiği metrikler önemlidir. Dice katsayısı, etkin örtüşme alanının bütün bir birleşimde ne kadar etkin olduğunu ifade eder ve (5.1) ile hesaplanır. Dice katsayısı, kesin referans ile tahmin arasındaki örtüşme tabanlı bir metriktir [64]. Örtüşme alanı, birleşme alanına eşit ise mükemmel bir bölütlemenin gerçekleştiği söylenebilir. Bu durumda, Dice katsayısı 1’e eşittir.

$$\text{Dice} = \frac{2 \times \text{DP}}{(\text{DP} + \text{YP}) + (\text{DP} + \text{YN})} \quad (5.1)$$

Doğruluk ise elde edilen DP ve DN sonuçların, bütün bir sonuç kümesi üzerine başarısını temsil etmektedir [65]. Doğruluk, (5.2)’deki eşitlik ile hesaplanır.

$$\text{Doğruluk} = \frac{\text{DP} + \text{DN}}{(\text{DP} + \text{DN} + \text{YP} + \text{YN})} \quad (5.2)$$

Caffe kütüphanesi Dice metriğini desteklememektedir. Ancak bu metrik, özel bir Python katmanı vasıtasıyla modele eklenebilmektedir. Deneysel çalışmalar için bu işlem gerçekleştirilmiştir.

5.3. Modellerin Eğitimleri ve Geçerlilikleri

Uygulama için belirlenen ağ parametrelerinin değerleri Tablo 5.2’de verilmiştir. Bu parametreler, uygulama için kullanılan tüm TKA modellerinde kullanılmak üzere genel olarak ayarlanmıştır.

Tablo 5.2. Ağ parametrelerinin değerleri

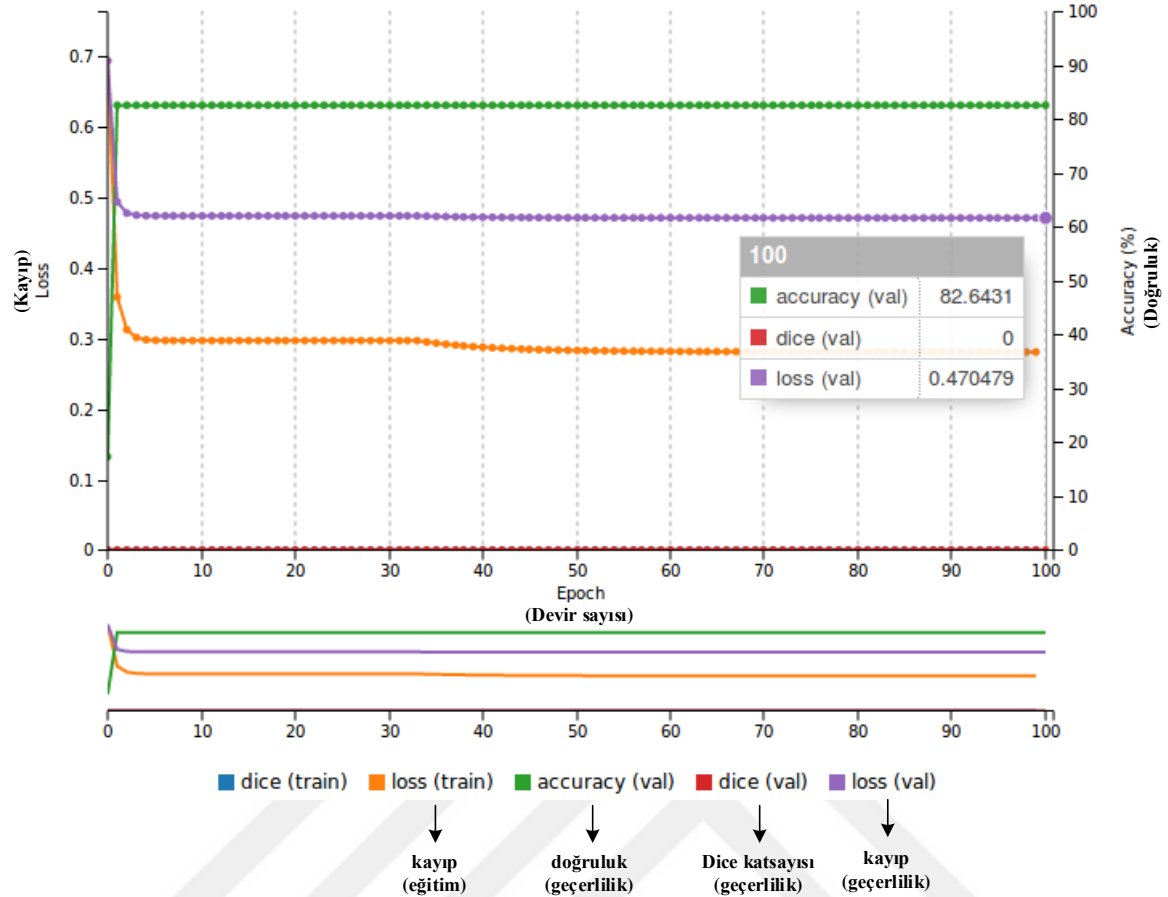
Parametre	Değeri
Temel öğrenme oranı	0.0001
Momentum katsayısı	0.9
Ağırlık bozunumu	10^{-6}
Yığın boyutu	1
Gama	0.1
Maksimum iterasyon sayısı	200000

Çözücü tipi olasılıksal gradyan inişi, çözücü modu GPU olarak seçilmiştir. Eğitim devir sayısı da 100 olarak belirlenmiştir. Devir sayısı, eğitim veri setinin ağı bir kez gösterildiği iterasyon sayısı olarak tanımlandığı için 1 devir sayısının tamamlanma işlemi, (5.3) ile ifade edilebilir.

$$\text{Eğitim görüntülerinin sayısı} = \text{Yığın boyutu} \times \text{İterasyon sayısı} \quad (5.3)$$

(5.3) dikkate alındığında 2000 eğitim görüntüsü için 1 devir sayısı, 1 yığın boyutu ile 2000 iterasyonda tamamlanmakta ve Tablo 5.2’de ifade edildiği gibi 100 devir sayısı için maksimum iterasyon sayısının 200000 olduğu görülmektedir.

İlk olarak DIGITS’te ağırlıkların rastgele başlatılmasıyla FCN-AlexNet modelinin eğitim işlemi gerçekleştirilmiş ve Şekil 5.3’teki sonuçlar elde edilmiştir.



Şekil 5.3. Ağırlıkların rastgele başlatılmasıyla FCN-AlexNet'in eğitimi sırasında eğitim-geçerlilik kaybı, geçerlilik doğruluğu ve Dice katsayısı

Şekil 5.3'te görüldüğü gibi ağın performansı çok kötüdür. Geçerlilik doğruluğunun, yaklaşık %82'de durağan bir noktaya geldiği görülse de geçerlilik Dice katsayısının tüm eğitim devir sayılarında 0 olduğu görülmüştür. Bu durum, geçerlilik setindeki piksellerin yaklaşık %82'sinin doğru olarak etiketlenmiş olmasına rağmen bunların tamamının arka plan pikselleri olduğu anlamına gelmektedir. Eğitilen model, bir test görüntüsü üzerinde test edildiğinde tüm görüntüyü arka plan olarak sınıflandıracaktır. Ağ, lezyonu temsil eden hiçbir pikseli öğrenememiştir. Sınıf dengesizliğinden dolayı ağın geçerlilik doğruluğu yüksek çıkmıştır.

Eğitim setine uygun olmayan bir ağı iyileştirebilmek için en önemli yollardan biri öğrenim transferidir. Derin bir ağ tarafından öğrenilen bilgiler, başka bir ağın performansını arttırabilmek için kullanılabilir ve bu işlem bilgisayarla görme uygulamaları için oldukça başarılıdır. Bu sebeple tez çalışması kapsamında gerçekleştirilen uygulamalar için gerekli modeller eğitilirken öğrenim transferinden yararlanılmıştır.

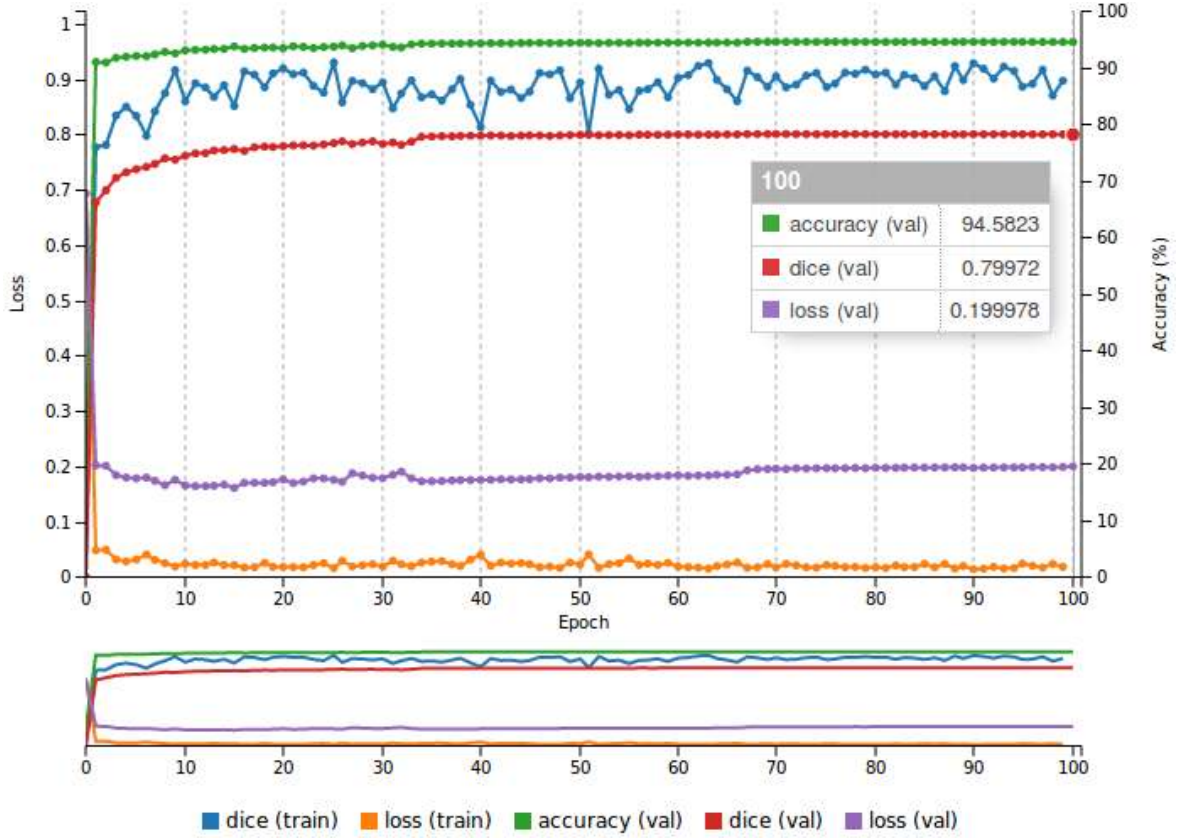
Makine öğrenmesi ve bilgisayarla görme alanlarındaki son gelişmeler, ilk olarak yaygın ölçütlerin kullanımıyla sağlanır. Bir modeli eğitmek için rastgele olarak başlatılmış ağırlıklardan başlanması zorunlu değildir. Öğrenim transferi, bir ağı başka bir veri setinde eğitilerek öğrendiği bilgilerin başka bir ağı performansını arttırmak için yeniden kullanılması işlemidir [66].

Çizgi, köşe, kenar ve doku gibi düşük seviyeli özniteliklerin birçoğunun KSA'lar vasıtasıyla herhangi bir veri setine hemen uygulanabileceğinden dolayı öğrenim transferi bilgisayarla görme alanında sıkça tercih edilmektedir. Bir KSA'nın büyük ve tıbbi olmayan çeşitli bir görüntü veri seti üzerinde eğitilmesi durumunda modelin çok çeşitli uzaysal ve yoğunluk yapılarını öğrenmiş olma ihtimalleri vardır. Bu yapıların birçoğu tıbbi görüntülerde mevcut olan anatomik özniteliklerde de bulunabilir.

Yüksek varyanslı standart veri setleri üzerinde eğitilen ve test edilen modeller başarılarını genellikle güçlü özniteliklere borçludur [66]. Öğrenim transferi, ImageNet gibi büyük bir veri seti üzerinde eğitilen, oldukça genelleştirilebilir ağırlıkları öğrenen bir modeli kullanmaya ve kullanılacak ağı durumuna uygun olacak şekilde ince bir ayar yapılmasına izin vermektedir.

Görüntü bölütleme işleminde piksel seviyesinde bir sınıflandırma yapıldığından öğrenimi ImageNet gibi görüntü sınıflandırma veri setlerinden transfer etmek çok anlamlıdır. Bu işlem, Caffe kütüphanesi tarafından desteklenmektedir. AlexNet'in fc6'sında ağırlıkların 4096×9216 biçiminde olduğu, FCN-AlexNet'in conv6'sında ise ağırlıkların $4096 \times 256 \times 6 \times 6$ biçiminde olduğu hatırlandığında ağırlık sayılarının eşit olduğu görülebilir. Ancak, biçimleri farklı olduğu için Caffe ağırlıkları otomatik olarak FCN-AlexNet'e taşıyamamaktadır. Bu ağırlıkları taşıma işlemi, Python komut dizisi yardımıyla ağırlıkları tam bağlı katmanlardan konvolüsyonel karşılıklarına aktararak gerçekleştirilebilir.

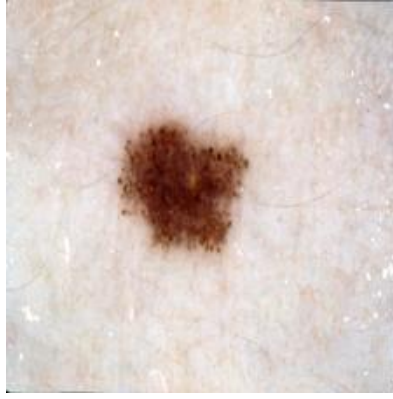
Gerçekleştirilen tıbbi görüntü bölütleme uygulaması için ImageNet veri seti üzerinde eğitilmiş AlexNet modeli, yukarıda değinilen yöntemler ile FCN-AlexNet için uygun hale getirilmiştir. Böylece elde edilen önceden eğitilmiş model de kullanılarak FCN-AlexNet modelinin eğitim işlemi gerçekleştirilmiş ve Şekil 5.4'teki sonuçlar elde edilmiştir.



Şekil 5.4. Önceden eğitilmiş modelin kullanılmasıyla FCN-AlexNet'in eğitimi sırasında eğitim-geçerlilik kaybı, geçerlilik doğruluğu ve Dice katsayısı

Şekil 5.4'te görüldüğü gibi modelin, önceden eğitilmiş FCN-AlexNet modeli yardımıyla geçerlilik Dice katsayısı hızlıca yükselmiş ve 40. devirden sonra %79 seviyelerinde durağanlığa ulaşmıştır. 100. devirde model, %79.972 geçerlilik Dice katsayısı ile %94.5823 geçerlilik doğruluğuna ulaşmış ve eğitim işlemi tamamlanmıştır. Bu durum, 100 devir sonunda elde edilen modelin geçerlilik setindeki piksellerin %94.5823'sinin doğru etiketlenerek %79.972'sinin kesin referans ile etkin bir biçimde örtüştüğü anlamına gelmektedir. Böylece, öğrenim transferi kullanılarak eğitilen modelin, ağırlıkları rastgele başlatılan FCN-AlexNet modeline göre oldukça yüksek başarımlarına sahip olduğu görülmüştür. Ayrıca, ağır eğitim setine uygun olmayışını ifade eden eğitim kaybının geçerlilik kaybıyla paralel olduğu ve kayıpların düşük değerlerde olduğu görülmüştür.

100. devirde elde edilen model örnek test görüntüleri ile test edildiğinde lezyon bölütleme, Şekil 5.5, Şekil 5.6 ve Şekil 5.7'deki gibi gerçekleşmiştir.



(a)



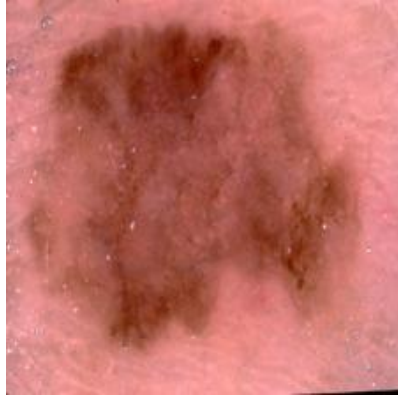
(b)



□ lezyon

(c)

Şekil 5.5. Önceden eğitilmiş modeli kullanarak eğitilen FCN-AlexNet ile örnek bir test görüntüsü için lezyon bölütleme işlemi-1: (a) Test görüntüsü, (b) Kesin referans görüntüsü, (c) Bölütleme çıkarımı



(a)

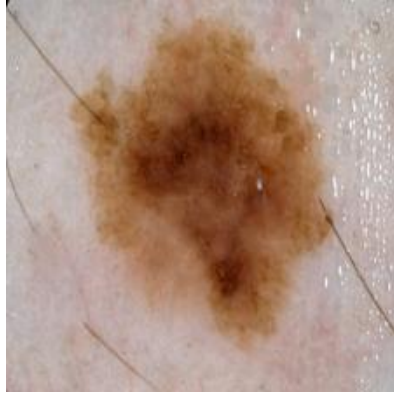


(b)



(c)

Şekil 5.6. Önceden eğitilmiş modeli kullanarak eğitilen FCN-AlexNet ile örnek bir test görüntüsü için lezyon bölütleme işlemi-2: (a) Test görüntüsü, (b) Kesin referans görüntüsü, (c) Bölütleme çıkarımı



(a)



(b)

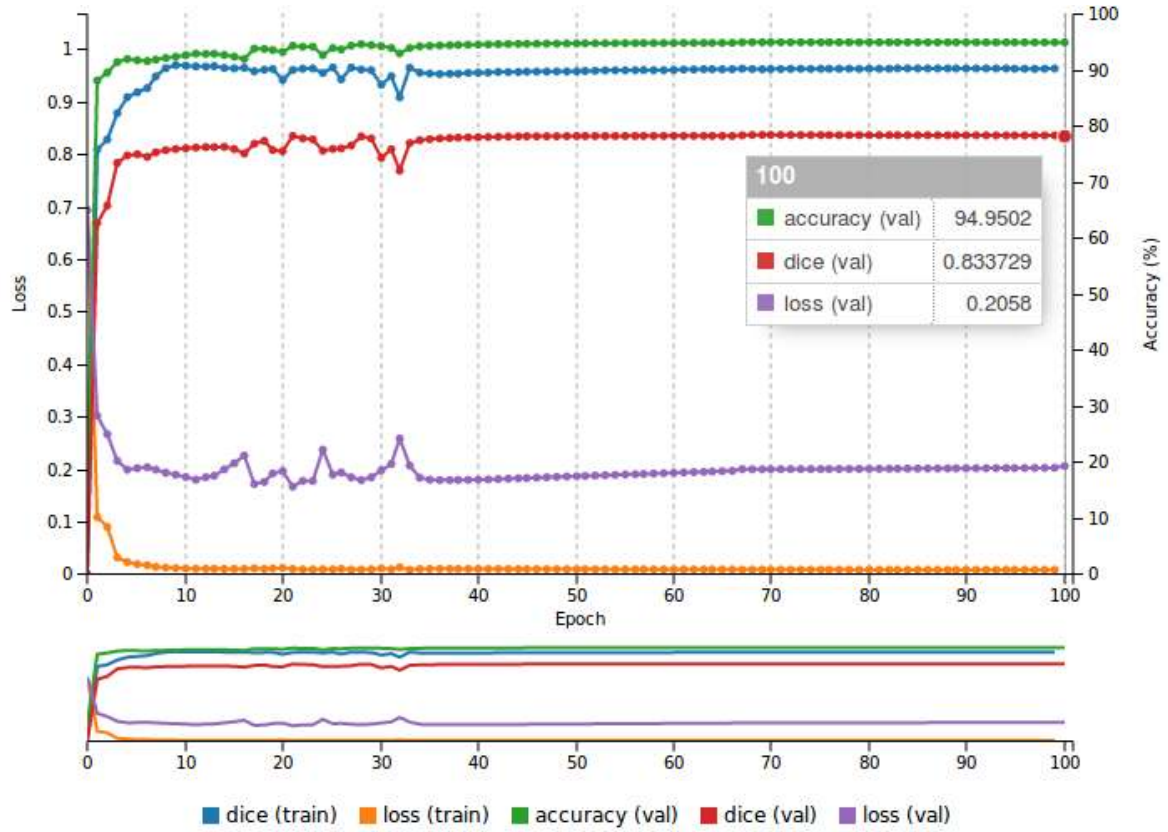


□ lezyon

(c)

Şekil 5.7. Önceden eğitilmiş modeli kullanarak eğitilen FCN-AlexNet ile örnek bir test görüntüsü için lezyon bölütleme işlemi-3: (a) Test görüntüsü, (b) Kesin referans görüntüsü, (c) Bölütleme çıkarımı

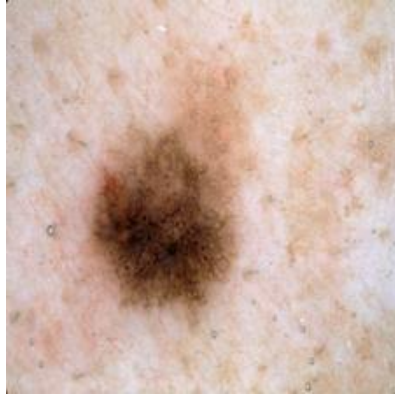
Şekil 5.5, Şekil 5.6 ve Şekil 5.7 gözlemlendiğinde bölütleme çıkarımları yeterliymiş gibi görünse de lezyon dış hatlarının kaba bir biçimde bölütlendiği görülmektedir. Bölütleme modelinin hassasiyetini iyileştirebilmek için daha ince ayarda bölütlemeler yapabilen FCN-8s ağı kullanılmıştır. Öğrenim transferi için PASCAL VOC veri setinde önceden eğitilmiş FCN-8s modeli kullanılarak FCN-8s ağının eğitim işlemleri gerçekleştirilmiş ve Şekil 5.8'deki sonuçlar elde edilmiştir.



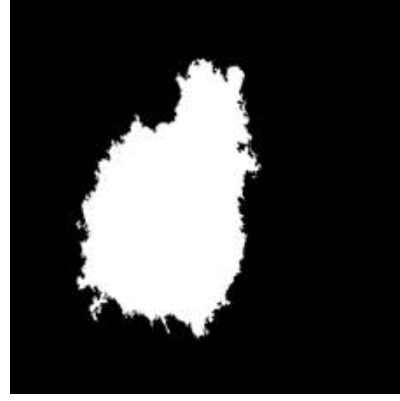
Şekil 5.8. Önceden eğitilmiş modelin kullanılmasıyla FCN-8s'nin eğitimi sırasında eğitim-geçerlilik kaybı, geçerlilik doğruluğu ve Dice katsayısı

Şekil 5.8'de görüldüğü gibi modelin, önceden eğitilmiş FCN-8s modeli yardımıyla geçerlilik Dice katsayısı hızlıca yükselip %80 seviyelerine çıkmıştır ve daha sonra yaklaşık %83'te durağanlığa ulaşmıştır. 100. devirde model, %83.3729 geçerlilik Dice katsayısı ile %94.9502 geçerlilik doğruluğuna ulaşmış ve eğitim işlemi tamamlanmıştır. Modelin geçerlilik doğruluğunun FCN-AlexNet modelinkinden çok az bir farkla yüksek olmasının yanında geçerlilik Dice katsayısının da belli oranda artışı modelin üstünlüğünü ön plana çıkarmıştır. Ayrıca, eğitim kaybının geçerlilik kaybıyla paralel olduğu ve kayıpların düşük değerlerde olduğu görülmüştür.

100. devirde elde edilen model örnek test görüntüleri ile test edildiğinde lezyon bölütleme, Şekil 5.9, Şekil 5.10 ve Şekil 5.11'deki gibi gerçekleşmiştir.



(a)



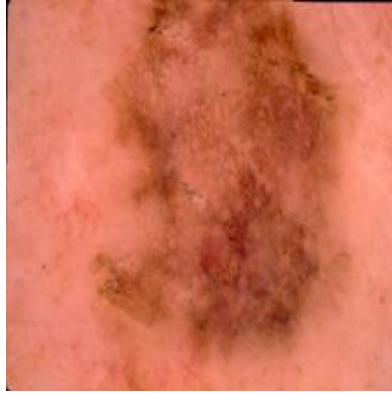
(b)



□ lezyon

(c)

Şekil 5.9. Önceden eğitilmiş modeli kullanarak eğitilen FCN-8s ile örnek bir test görüntüsü için lezyon bölütleme işlemi-1: (a) Test görüntüsü, (b) Kesin referans görüntüsü, (c) Bölütleme çıkarımı



(a)



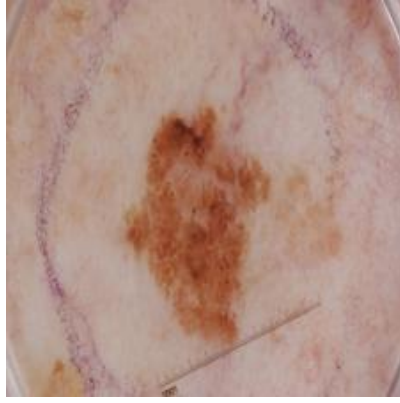
(b)



□ lezyon

(c)

Şekil 5.10. Önceden eğitilmiş modeli kullanarak eğitilen FCN-8s ile örnek bir test görüntüsü için lezyon bölütleme işlemi-2: (a) Test görüntüsü, (b) Kesin referans görüntüsü, (c) Bölütleme çıkarımı



(a)



(b)

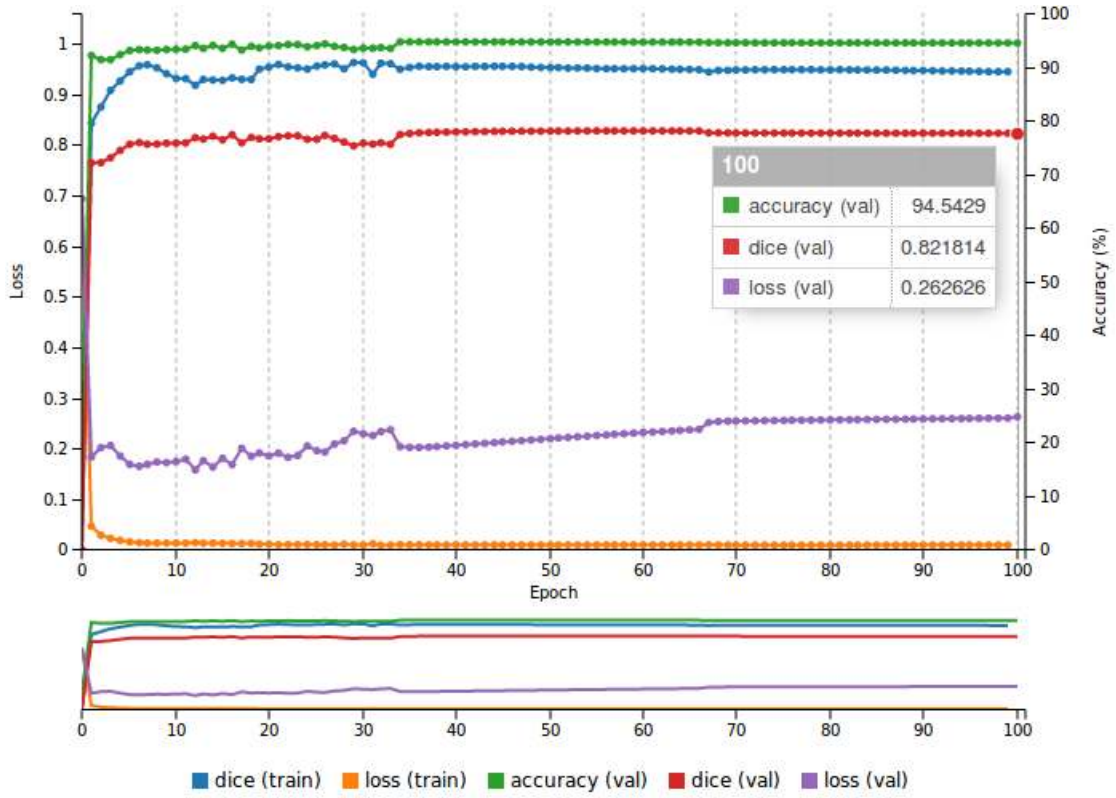


(c)

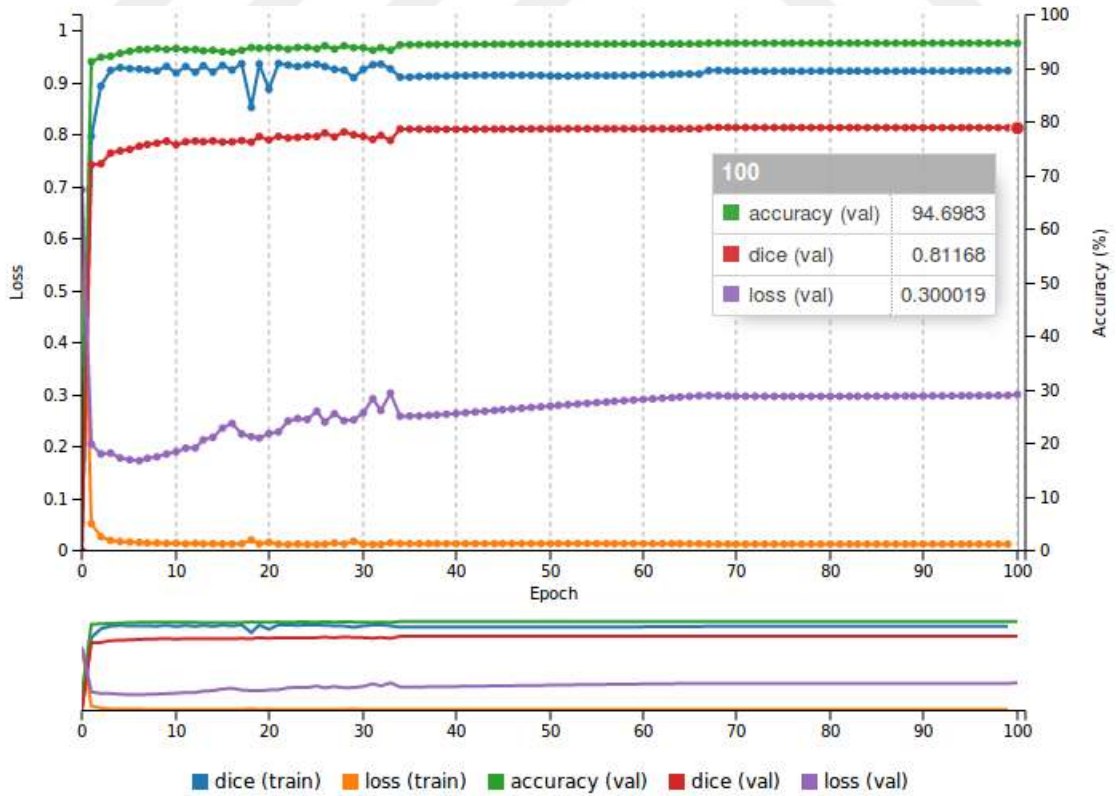
Şekil 5.11. Önceden eğitilmiş modeli kullanarak eğitilen FCN-8s ile örnek bir test görüntüsü için lezyon bölütleme işlemi-3: (a) Test görüntüsü, (b) Kesin referans görüntüsü, (c) Bölütleme çıkarımı

Şekil 5.9, Şekil 5.10 ve Şekil 5.11 gözlemlendiğinde FCN-8s mimarisinin, 8×8 piksel bloklarına kadar daha ince ayarda tahminler yapabilmesi sayesinde 32×32 piksel bloklarında tahminler yapan FCN-AlexNet'e göre daha keskin lezyon dış hatlarıyla bölütleme sağladığı görülmüştür.

Deneyisel çalışmalar için FCN-16s ve FCN-32s ağlarının da eğitimleri gerçekleştirilmiş olup sırasıyla Şekil 5.12 ve Şekil 5.13'teki sonuçlar elde edilmiştir.



Şekil 5.12. Önceden eğitilmiş modelin kullanılmasıyla FCN-16s'nin eğitimi sırasında eğitim-geçerlilik kaybı,geçerlilik doğruluğu ve Dice katsayısı

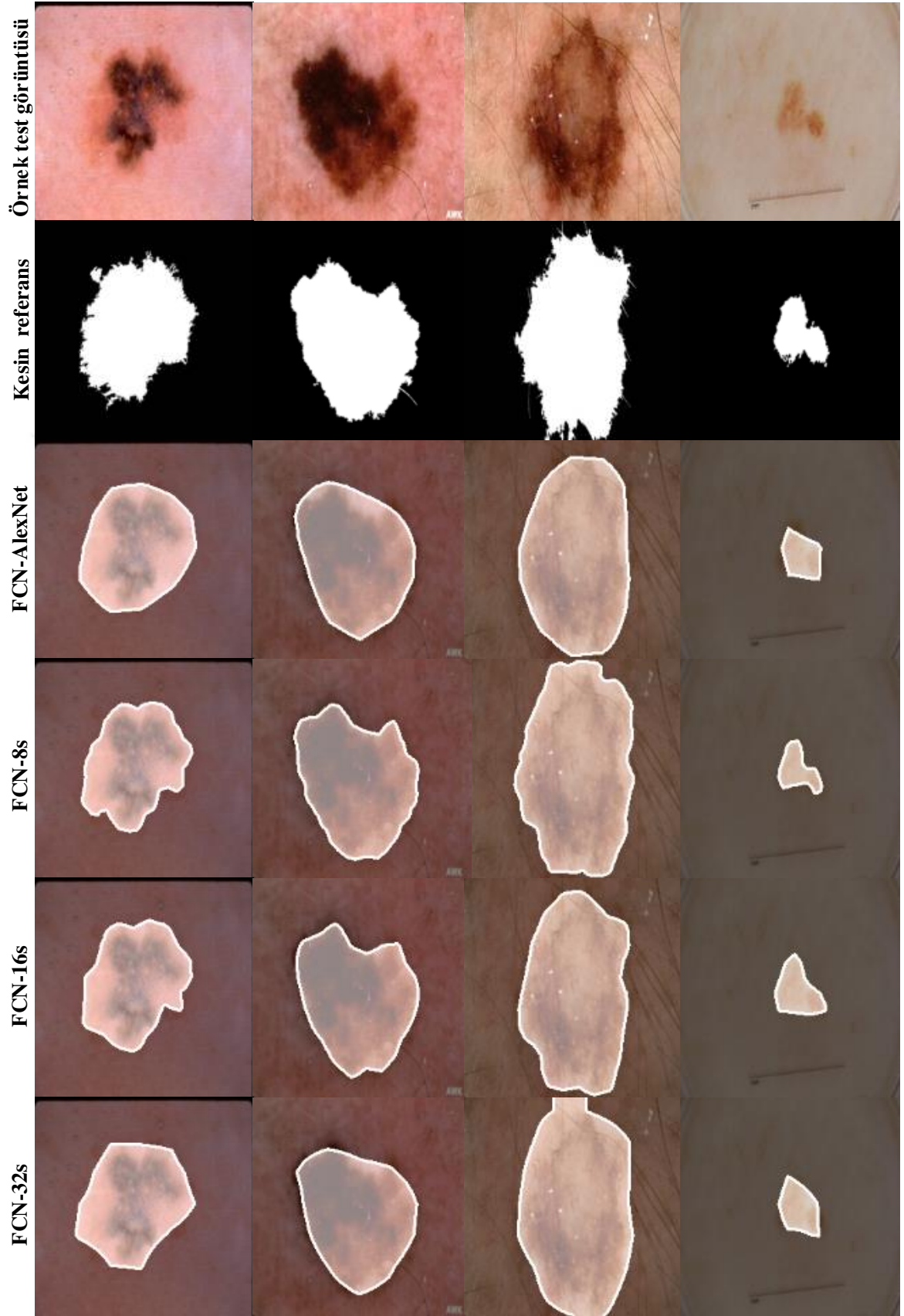


Şekil 5.13. Önceden eğitilmiş modelin kullanılmasıyla FCN-32s'nin eğitimi sırasında eğitim-geçerlilik kaybı, geçerlilik doğruluğu ve Dice katsayısı

FCN-16s ve FCN-32s modellerinin geçerlilik doğruluklarının Şekil 5.12 ve Şekil 5.13'te görüldüğü gibi FCN-8s'ye benzer şekilde %94 seviyelerinde durağanlığa ulaşmıştır. Geçerlilik Dice katsayılarının ise FCN-8s'ye kıyasla daha düşük olduğu görülmüştür.

FCN-AlexNet, FCN-8s, FCN-16s ve FCN-32s mimarileriyle elde edilen modellerin, aynı test görüntüleri üzerinde lezyon bölütleme çıkarımlarının karşılaştırmalı olarak gözlemlenebilmesi için Şekil 5.14 verilmiştir.

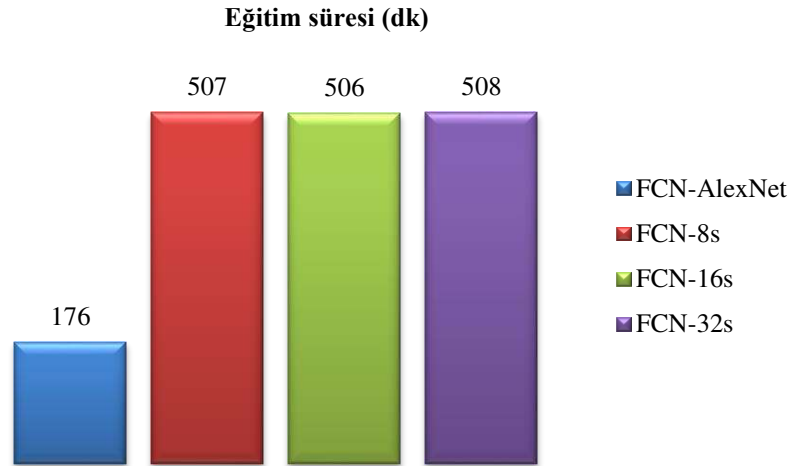




Şekil 5.14. Bazı örnek test görüntüleri için lezyon bölütleme çıkarımlarının kullanılan TKA mimarilerine göre karşılaştırılması

Örnek test ve kesin referans görüntüleri dikkate alınarak Şekil 5.14'teki lezyon bölütleme işlemleri analiz edildiğinde, FCN-AlexNet modeliyle gerçekleştirilen bölütleme işlemlerinde lezyon dış hatlarının kaba bir şekilde bölütlendiği gözlemlenmiştir. FCN-8s modeliyle ise FCN-AlexNet'in aksine lezyon dış hatları keskin bir şekilde bölütlenerek bölütleme çıkarımlarının kesin referans görüntülerine yakın olduğu görülmüştür. FCN-16s modelinin ise FCN-8s kadar keskin olmasa da lezyon dış hatlarını başarılı sayılabilecek derecede bölütleyebildiği gözlemlenmiştir. Son olarak, FCN-32s modelinin bölütleme çıkarımları incelendiğinde FCN-AlexNet'e çok yakın bölütlemelerin gerçekleştiği görülmüş ancak küçük farklarla daha kullanışlı bir model olabileceği kanısına varılmıştır.

Tez çalışması kapsamında gerçekleştirilen lezyon bölütleme uygulaması için eğitilen modellerin eğitim süreleri ise Şekil 5.15'te verilmiştir.



Şekil 5.15. Modellerin eğitim süreleri

Modellerin eğitim süreleri dikkate alındığında FCN-AlexNet'in eğitim süresi diğer üç modele göre bir hayli düşüktür. FCN-8s, FCN-16s ve FCN-32 modellerinin ise eğitim süreleri hemen hemen aynıdır.

6. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, derin öğrenme yaklaşımlarından TKA'lar ile en tehlikeli cilt kanseri türü olan melanomun tespitine katkıda bulunacak bir cilt lezyonu bölütleme uygulaması gerçekleştirilmiştir. Uygulama için FCN-AlexNet, FCN-8s, FCN-16s ve FCN-32s isimli dört popüler TKA mimarisinden yararlanılmıştır. Bu mimariler ve kullanılan ISIC 2017 veri seti dikkate alındığında, bu çalışma bilimsel yazında ilk kez gerçekleştirilmiştir.

Uygulama, DIGITS ortamında Caffe kütüphanesi vasıtasıyla gerçekleştirilmiş olup deneysel çalışmalar için kullanılan TKA mimarilerinin bölütleme performansları karşılaştırılmıştır. Uygulamada veri seti olarak cilt lezyon görüntülerinden oluşan ISIC 2017 veri seti kullanılmıştır. Deneysel çalışmalar, 4. nesil Intel i5 3.4 GHz işlemci, 8 GB RAM ve NVIDIA GTX Titan X Pascal ekran kartına sahip bir masaüstü bilgisayar üzerinde gerçekleştirilmiştir. Kullanılan ekran kartının CUDA desteği aracılığıyla uygulama için gerekli hesaplamalarda GPU tabanlı paralel hesaplama gücünden yararlanılmıştır. Böylece, ağların eğitimi için harcanan zaman büyük oranda azaltılmıştır.

Uygulama için öncelikle TKA'lar ayrı ayrı eğitilmiş ve eğitilen bu ağ modellerinin geçerlilik veri seti üzerindeki doğrulukları ve Dice katsayıları karşılaştırılmıştır. Deneysel çalışmalarda, bölütleme çıkarımları görselleştirilerek kullanılan TKA mimarilerinin nesneleri ne kadar keskinlikte bölütleyebildiği göz önünde bulundurulmuştur.

Önceden eğitilmiş modellerdeki ağırlıklar kullanılarak eğitilen FCN-AlexNet, FCN-8s, FCN-16 ve FCN-32s modelleri ile sırasıyla %94.5823, %94.9502, %94.5429 ve %94.6983'lük geçerlilik doğrulukları elde edilmiştir. Az sınıf etiketine sahip bölütleme uygulamalarında başarıyı ifade etmede en önemli metriklerden olan Dice katsayıları ise %79.972 (FCN-AlexNet), %83.3729 (FCN-8s), %82.1814 (FCN-16s) ve %81.168 (FCN-32s) olarak elde edilmiştir.

Elde edilen TKA modellerinin performansları, rakamsal olarak birbirine çok yakın görünse de FCN-AlexNet ve FCN-32s modeliyle gerçekleştirilen bölütleme işlemlerinde lezyon dış hatlarının kaba bir şekilde bölütlendiği görülmüştür. FCN-8s modeliyle ise FCN-AlexNet ve FCN-32s'in aksine lezyon dış hatları keskin bir şekilde bölütlenerek bölütleme çıkarımlarının kesin referans görüntülerine çok yakın olduğu görülmüştür. FCN-16s modelinin ise FCN-8s kadar keskin olmasa da nesne dış hatlarını başarılı sayılabilecek derecede bölütleyebildiği görülmüştür. Son olarak, FCN-32s modelinin bölütleme

ıkarımları incelendiğinde FCN-AlexNet'e ok yakın blütlemelerin gerekleştiiği grlmüş ancak kk farklarla da olsa daha kullanışlı bir model olabileceği gzlemlenmiştir.

TKA modellerinin eđitim sreleri karřılařtırıldığında ise FCN-AlexNet'in eđitimi iin harcanan srenin, birbirleriyle hemen hemen aynı eđitim srelerine sahip diđer TKA modellerinin yaklaşık te biri olduđu grlmüştür. Ancak, uygulama iin model eđitiminin bir defa gerekleřtirildiği gz nnde bulunduurulduğunda uygun modelin tercihinde ok nemli bir yere sahip olmadığı anlaşılmıştır. Bu yzden, gerekleřtirilen cilt lezyonu bltleme uygulaması iin kullanılan modellerden en uygun modelin FCN-8s olduđuna bir kere daha kanaat getirilmiştir.

Elde edilen deneysel sonular, cilt lezyonu bltleme uygulamalarında derin đrenme yaklaşımlarından TKA'ların elveriřli olduđunu gstermiştir. Ayrıca, TKA'ların tıbbi grnt bltleme bařta olmak zere birok piksel tabanlı uygulamalara hitap eden modellerdeki ađ yapıları olduđu anlaşılmıştır.

Tez alışması kapsamında yapılan deneysel alışmaların, bilimsel yazına ve tıbbi grnt bltleme ile ilgili alışmalar yapan arařtırmacılara katkı sađlayacağı dřnlmektedir. Bu alışmalardaki uygulamalara ynelik bazı neriler de yapılabilir. rneđin; eđitim veri setindeki lezyon grntlerinin sayısı veri arttırma yntemleriyle arttırılarak veya kullanılan ađ mimarilerinde deđiřiklik yapılarak modellerin performansları ykseltilebilir. Ayrıca, daha gl hesaplama gcne sahip GPU'ya sahip bir bilgisayar kullanılarak modellerin eđitim sreleri kısaltılabilir. Son olarak, eđitim ařaması iin devir sayısı arttırılarak ađı daha uzun sre eđitmek de ıkarım performansını ykseltmenin bir zm olabilir.

Gelecek alışmamızda, bu tez alışması ile bltlenmiş lezyon grntleri zerinde melanom tespit uygulamasının gerekleřtirilmesi dřnlmektedir.

KAYNAKLAR

- [1] Karpathy, A., Convolutional Neural Networks for Visual Recognition, *Course Notes*. <http://cs231n.github.io/convolutional-networks/>, 21 Aralık 2018.
- [2] **Wahba, M.**, 2009. An automated modified region growing technique for prostate segmentation in trans-rectal ultrasound images, *Master of Science Thesis*, University of Waterloo Institute of Science, Ontario, Canada.
- [3] **Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S. and Berg, A. C.**, 2015. Imagenet large scale visual recognition challenge, *International Journal of Computer Vision*, Springer Berlin, Heidelberg, **115(3)**, 211-252.
- [4] **Le, Q. V.**, 2013. Building high-level features using large scale unsupervised learning, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, 8595-8598.
- [5] **Taigman, Y., Yang, M., Ranzato, M. A. and Wolf, L.**, 2014. Deepface: Closing the gap to human-level performance in face verification, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colombus, Ohio, USA, 1701-1708.
- [6] **Deng, L. and Yu, D.**, 2014. Deep learning: methods and applications, *Foundations and Trends® in Signal Processing*, **7**, 197-387.
- [7] **Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C. and Chen, J.**, 2016. Deep speech 2: End-to-end speech recognition in english and mandarin, *International Conference on Machine Learning (ICML)*, New York, USA, 173-182.
- [8] <https://trends.google.com/trend/explore?q=deep%20learning>, Trend search of “Deep Learning” in Google. 2 Ocak 2019.
- [9] **Long, J., Shelhamer, E. and Darrell, T.**, 2015. Fully convolutional networks for semantic segmentation, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, Massachusetts, USA, 3431-3440.
- [10] **Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A. L.**, 2015. Semantic image segmentation with deep convolutional nets and fully connected crfs, *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 1-14.
- [11] **Simonyan, K. and Zisserman, A.**, 2015. Very deep convolutional networks for large-scale image recognition, *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 1-14.

- [12] **Ronneberger, O., Fischer, P. and Brox, T.,** 2015. U-net: Convolutional networks for biomedical image segmentation, *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Munich, Germany, 234-241.
- [13] **Yu, L., Chen, H., Dou, Q., Qin, J. and Heng, P. A.,** 2017. Automated melanoma recognition in dermoscopy images via very deep residual networks, *IEEE Transactions on Medical Imaging (T-MI)*, **36(4)**, 994-1004.
- [14] **Codella, N., Cai, J., Abedini, M., Garnavi, R., Halpern, A. and Smith, J. R.,** 2015. Deep learning, sparse coding, and SVM for melanoma recognition in dermoscopy images, *International Workshop on Machine Learning in Medical Imaging*, pp. 118-126, Springer, Cham, Switzerland.
- [15] **Codella, N. C., Nguyen, Q. B., Pankanti, S., Gutman, D. A., Helba, B., Halpern, A. C. and Smith, J. R.,** 2017. Deep learning ensembles for melanoma recognition in dermoscopy images, *IBM Journal of Research and Development*, **61(4-5)**, 5:1-5:15.
- [16] **Kawahara, J., Bentaieb, A. and Hamarneh, G.,** 2016 Deep features to classify skin lesions, *International Symposium on Biomedical Imaging (ISBI)*, Prague, Czech Republic, 1397-1400.
- [17] **Jin, X., Ye, H., Li, L. and Xia, Q.,** 2017. Image segmentation of liver CT based on fully convolutional network, *IEEE International Symposium on Computational Intelligence and Design (ISCID)*, **1**, 210-213.
- [18] **Brosch, T., Tang, L. Y., Yoo, Y., Li, D. K., Traboulsee, A. and Tam, R.,** 2016. Deep 3D convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation, *IEEE Transactions on Medical Imaging (T-MI)*, **35(5)**, 1229-1239.
- [19] **Akkus, Z., Galimzianova, A., Hoogi, A., Rubin, D. L. and Erickson, B. J.,** 2017. Deep learning for brain MRI segmentation: state of the art and future directions, *Journal of Digital Imaging (JDI)*, **30(4)**, 449-459.
- [20] **Milletari, F., Navab, N. and Ahmadi, S. A.,** 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation, *IEEE International Conference on 3D Vision (3DV)*, Stanford, California, USA, 565-571.
- [21] **Pereira, S., Pinto, A., Alves, V. and Silva, C. A.,** 2016. Brain tumor segmentation using convolutional neural networks in MRI images, *IEEE Transactions on Medical Imaging (T-MI)*, **35(5)**, 1240-1251.
- [22] **Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A. L.,** 2018. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **40(4)**, 834-848.

- [23] **Kline, T. L., Korfiatis, P., Edwards, M. E., Blais, J. D., Czerwicz, F. S., Harris, P. C. and Erickson, B. J.**, 2017. Performance of an artificial multi-observer deep neural network for fully automated segmentation of polycystic kidneys, *Journal of Digital Imaging (JDI)*, **30(4)**, 442-448.
- [24] **Mishra, R. and Daescu, O.**, 2017. Deep learning for skin lesion segmentation, *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Kansas City, MO, USA, 1189-1194.
- [25] **Yuan, Y., Chao, M. and Lo, Y. C.**, 2017. Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance, *IEEE Transactions on Medical Imaging (T-MI)*, **36(9)**, 1876-1886.
- [26] **Berseth, M.**, 2017. ISIC 2017-skin lesion analysis towards melanoma detection, *arXiv preprint arXiv:1703.00523*.
- [27] **Bi, L., Kim, J., Ahn, E. and Feng, D.**, 2017. Automatic skin lesion analysis using large scale dermoscopy images and deep residual networks, *arXiv preprint arXiv:1703.04197*.
- [28] **Vesal, S., Ravikumar, N. and Maier, A.**, 2018. SkinNet: A deep learning framework for skin lesion segmentation, *arXiv preprint arXiv:1806.09522*.
- [29] <https://www.oreilly.com/library/view/deep-learning/9781491924570/ch04.html>, Major Architectures of Deep Networks. 26 Ocak 2019.
- [30] **Gentleman, R., Huber, W. and Carey, V. J.**, 2008. Supervised machine learning, *Bioconductor Case Studies*, Springer, 121-136.
- [31] **Caruana, R. and Niculescu-Mizil, A.**, 2006. An empirical comparison of supervised learning algorithms, *International Conference on Machine Learning (ICML)*, Pittsburgh, Pennsylvania, USA, 161-168.
- [32] **Haykin, S. S.**, 2001. Neural networks: a comprehensive foundation, Tsinghua University Press.
- [33] **Zurada, J. M.**, 1992. Introduction to artificial neural systems, West Publishing Company, **8**.
- [34] http://www.nvidia.com/object/cuda_home_new.html, CUDA. 5 Şubat 2019.
- [35] <http://caffe.berkeleyvision.org/>, BVLC Caffe. 5 Şubat 2019.
- [36] <http://torch.ch/>, What is Torch?. 6 Şubat 2019.
- [37] <http://machinelearningmastery.com/introduction-python-deep-learning-library-theano/>, Introduction to the python deep learning library Theano. 8 Mart 2019.

- [38] <http://deeplearning.net/software/theano/>, Theano. 8 Mart 2019.
- [39] <https://www.tensorflow.org/>, About TensorFlow. 10 Mart 2019.
- [40] <https://keras.io/>, Keras: Deep Learning library for Theano and TensorFlow. 10 Mart 2019.
- [41] <https://developer.nvidia.com/cudnn>, NVIDIA CuDNN. 12 Mart 2019.
- [42] <https://devblogs.nvidia.com/parallelforall/digits-deep-learning-gpu-training-system/>, DIGITS: Deep Learning GPU Training System. 15 Mart 2019.
- [43] **LeCun, Y.**, 1989. Backpropagation applied to handwritten zip code recognition, *Neural Computation*, **1(4)**, 541-551.
- [44] **Shivaprakash, M.**, 2016. Semantic segmentation of satellite images using deep learning, *Master's Thesis*, Czech Technical University in Prague and Luleå University of Technology, Institute of Science, Prague, Czech Republic.
- [45] **Dumoulin, V. and Visin, F.**, 2016. A guide to convolution arithmetic for deep learning, *arXiv preprint arXiv:1603.07285*.
- [46] <https://leonardoaraujosantos.gitbooks.io/artificial-intelligence/content/convolution.html>, Convolution. 15 Mart 2019.
- [47] **Krizhevsky, A., Sutskever, I. and Hinton, G. E.**, 2012. Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems (NIPS)*, 1097-1105.
- [48] <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>, An Intuitive Explanation of Convolutional Neural Networks. 18 Mart 2019.
- [49] **Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.**, 2014. Dropout: A simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research (JMLR)*, **15(1)**, 1929-1958.
- [50] **Rojas, R.**, 2013. Neural networks: a systematic introduction, Springer Science and Business Media.
- [51] **Bengio, Y.**, 2012. Practical recommendations for gradient-based training of deep architectures, *Neural networks: Tricks of the trade*, Springer Berlin, Heidelberg, 437-478.
- [52] **Mishkin, D. and Matas, J.**, 2015. All you need is a good init, *arXiv preprint arXiv:1511.06422*.

- [53] **Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D. and Rabinovich, A.**, 2015. Going deeper with convolutions, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, Massachusetts, USA, 1-9.
- [54] **He, K., Zhang, X., Ren, S. and Sun, J.**, 2016. Deep residual learning for image recognition, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 770-778.
- [55] <https://github.com/NVIDIA/DIGITS/tree/master/examples/semantic-segmentation>, Using DIGITS to train a semantic segmentation neural network. 20 Mart 2019.
- [56] **Kaymak, C. and Ucar, A.**, 2019. A brief survey and an application of semantic image segmentation for autonomous driving, *Handbook of Deep Learning Applications*, pp. 161-200, Springer, Cham, Switzerland.
- [57] **Everingham, M., Van Gool, L., Williams, C. K., Winn, J. and Zisserman, A.**, 2010. The Pascal visual object classes (VOC) challenge, *International Journal of Computer Vision*, **88(2)**, 303-338.
- [58] **Jerant, A. F., Johnson, J. T., Demastes Sheridan, C. and Caffrey, T. J.**, 2000. Early detection and treatment of skin cancer, *American Family Physician (AFP)*, **62(2)**, 357-386.
- [59] <https://en.wikipedia.org/w/index.php?title=Melanoma&oldid=791942741>, Melanoma. 1 Nisan 2019.
- [60] **Kittler, H., Pehamberger, H., Wolff, K. and Binder, M.**, 2002. Diagnostic accuracy of dermoscopy, *The Lancet Oncology*, **3(3)**, 159-165.
- [61] **Codella, N. C., Gutman, D., Celebi, M. E., Helba, B., Marchetti, M. A., Dusza, S. W., Halpern, A.**, 2018. Skin lesion analysis toward melanoma detection: A challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), hosted by the International Skin Imaging Collaboration (ISIC), *IEEE 15th International Symposium on Biomedical Imaging (ISBI)*, Washington, D.C., USA, 168-172.
- [62] <https://challenge.kitware.com/#phase/5841916ccad3a51cc66c8db0>, Part 1: Lesion Segmentation, ISIC 2017: Skin Lesion Analysis Towards Melanoma Detection. 4 Nisan 2019.
- [63] **Dice, L. R.**, 1945. Measures of the amount of ecologic association between species, *Ecology*, **26(3)**, 297-302.
- [64] **Taha, A. A. and Hanbury, A.**, 2015. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool, *BMC Medical Imaging*, **15(1)**, 29.

- [65] **Popovic, A., De la Fuente, M., Engelhardt, M. and Radermacher, K.,** 2007. Statistical validation metric for accuracy assessment in medical image segmentation, *International Journal of Computer Assisted Radiology and Surgery (IJCARS)*, **2(3-4)**, 169-181.
- [66] **Yosinski, J., Clune, J., Bengio, Y. and Lipson, H.,** 2014. How transferable are features in deep neural networks?, *Advances in Neural Information Processing Systems (NIPS)*, 3320-3328.



ÖZGEÇMİŞ

Rüya KAYMAK, 1990 yılında Malatya’da doğdu. Orta öğrenimini Malaya Sümer Lisesi’nde (yabancı dil ağırlıklı) tamamladı. 2009 yılında girdiği Fırat Üniversitesi Mühendislik Fakültesi Mekatronik Mühendisliği Bölümü’nden 2013 yılında mezun oldu. 2014 yılında girdiği Fırat Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalı’nda yüksek lisans eğitimine başladı.

