



**DERİN ÖĞRENME İLE YÜKSEK
ÇÖZÜNÜRLÜKLÜ HAVA GÖRÜNTÜLERİNDE
YOLLARIN TESPİT EDİLMESİ**

Figen ÖNÜN
Yüksek Lisans Tezi

Ekobilişim Anabilim Dalı
Danışman: Doç. Dr. Galip AYDIN

AĞUSTOS-2019

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME İLE YÜKSEK ÇÖZÜNÜRLÜKLÜ HAVA
GÖRÜNTÜLERİNDE YOLLARIN TESPİT EDİLMESİ

YÜKSEK LİSANS TEZİ

Figen ÖNÜN
(152141103)

Tezin Enstitüye Verildiği Tarih: 1 Temmuz 2019

Tezin Savunulduğu Tarih: 9 Ağustos 2019

Tez Danışmanı: Doç. Dr. Galip AYDIN (F.Ü)

Diğer Jüri Üyeleri: Dr. Öğr. Üyesi Betül AY(F.Ü)

Dr. Öğr. Üyesi Nuh ALPASLAN(B.Ü)

AĞUSTOS - 2019

ÖNSÖZ

Yapay Zekâ; çeşitli çalışma alanlarını kapsayacak şekilde gerçek hayat ile ilişkilendirilerek birçok alanda çalışmalara konu olmuştur. Her geçen gün artan bu çalışmalarda insan iş gücünü azaltacak, zamandan tasarruf sağlayacak bununla beraber dinamik ve otomatize edilmiş yapıdaki çalışmaların devam etmesi yapay zekâdaki gelişimin de devam ettiğini göstermektedir. Yapay zekâ alanında yapılan çalışmalar sonucunda ortaya çıkan teknolojilerden biri olan Derin Öğrenme bu tez çalışmasına büyük katkı sağlamıştır.

Bu tez çalışmasında bilgi ve deneyimi ile çalışma süresince desteğini esirgemeyen değerli tez danışmanım ve hocam **Doç. Dr. Galip AYDIN**'a ve hayatım boyunca aldığım kararlar konusunda hep yanımda olan benden desteğini esirgemeyen aileme çok teşekkür ederim.

Figen ÖNÜN

ELAZIĞ-2018

İÇİNDEKİLER

Sayfa No

ÖNSÖZ	I
İÇİNDEKİLER.....	II
ÖZET	III
SUMMARY	IV
ŞEKİLLER LİSTESİ	V
TABLolar LİSTESİ	VII
KISALTMALAR LİSTESİ	VIII
1. GİRİŞ.....	1
2. LİTERATÜR ÖZETİ.....	5
3. DERİN ÖĞRENME	9
3.1. MAKİNE ÖĞRENMESİ	11
3.2. YAPAY SİNİR AĞLARI	13
3.3. DERİN ÖĞRENME MİMARİLERİ	16
3.3.1. Deep Neural Network-DNN	18
3.3.2. Deep Belief Network	18
3.3.3. Deep Auto-Encoders.....	18
3.3.4. Boltzmann Machines-BM.....	18
3.3.5. Restricted Boltzmann Machines-RBM.....	18
3.3.6. Recurrent Neural Networks-RNNs	19
3.3.7. Recognizing Recurrent Neural Networks	19
3.3.8. Convolution Neural Network-CNN.....	19
3.3.9. Fully Convolution Neural Network-(FCNN)	32
3.4. DERİN ÖĞRENME İLE SINIFLANDIRMA	34
4. DERİN ÖĞRENME AĞ YAPILARI	36
4.1. ALEXNET AĞ YAPISI	36
4.2. VGGNET AĞ YAPISI	37
4.3. U-NET AĞ YAPISI.....	38
5. UYGULAMALAR.....	40
5.1. GÖRÜNTÜ VERİ SETLERİ.....	40
5.1.1. Derin Öğrenme Unet Modeli İle Toronto Veri Seti Eğitimi.....	42
5.1.2. Derin Öğrenme Unet Modeli İle Paris Veri Seti Eğitimi	44
5.1.3. Derin Öğrenme Unet Modeli İle Vegas Veri Seti Eğitimi	48
5.1.4. Derin Öğrenme Unet Modeli İle Shanghai Veri Seti Eğitimi.....	52
5.1.5. Derin Öğrenme Unet Modeli İle Khartoum Veri Seti Eğitimi	54
5.1.6. Derin Öğrenme Unet Modeli İle Tüm Veri Setleri Eğitimi.....	58
5.2. TÜRKİYE GÖRÜNTÜ VERİ SETLERİ	60
5.2.1. Derin Öğrenme Unet Modeli İle Elazığ Veri Seti Eğitimi	62
5.2.2. Malatya Uydu Görüntülerine Ait Veri Setinin Hazırlanması ve Unet Modeli ile Eğitimi	64
5.3. TAHMİNLEME SONUCUNDA OLUŞAN GÖRÜNTÜDEKİ YOLLARIN GEOGRAPHİC VERİLERİNİN ÇIKARILMASI	71
6. SONUÇ	74
KAYNAKLAR.....	76
ÖZGEÇMİŞ	82

ÖZET

Derin öğrenme, önceden bilinen veri setleri üzerinde çok katmanlı yapay sinir ağı ile modelleme yapar. Bilinmeyen veriler için özelliklerin öğrenimi yardımıyla analiz ve sınıflandırma yapılmasına olanak sağlayan makine öğrenmesi algoritmalarını temel alan bir yaklaşımdır. Yapay zekâ teknolojisinin hayatımızda kullandığımız cihazlarda kullanılabilirliğinin artması sonucunda derin öğrenme ortaya çıkmış ve bu anlamda yapılan çalışmalar gün geçtikçe artmaya devam etmiştir.

Derin öğrenme teknolojileri sayesinde nesne tanıma ve bu nesnelerin analizi sonucunda sınıflandırma çalışmaları gün geçtikçe hız kazanmıştır. En temel derin öğrenme algoritmaları konvolüsyonel sinir ağları ve çok katmanlı yapay sinir ağları olmuştur. Analiz çalışmalarında özellikle görüntü işleme alanında derin öğrenme algoritmaları yaygın bir şekilde kullanılmıştır. Bu tez çalışmasında, derin öğrenme alanında önemli bir yeri olan yüksek çözünürlüklü hava görüntülerinin Evrişimsel Sinir Ağları ile analiz edilmesi ve yolların tespiti konusu işlenecektir. Bu çalışma ile büyük boyutlu hava görüntülerinin işlenebilir resim boyutuna gelmesi için bölümlenerek işlenmesi ve veri seti elde edilmesi sağlanmıştır. Hazırlanan bu veri seti eğitilerek daha sonra test amaçlı ayrılan verilerin Derin Öğrenme Unet ağ modeli ile istenilen yolların tespiti gerçekleştirilmiştir. Türkiye için derin öğrenme çalışmalarında kullanılabilecek veri seti oluşturma yöntemleri sunulmuştur. Eğitilen sinir ağının yaptığı tahminleme sonucunda görüntüdeki yol ağının HAVERSINE ve BEARING denklemleri kullanılarak geliştirilen algoritma ile coğrafik verileri çıkarılmıştır. Bu çalışma sonucunda yüksek çözünürlüklü uydu görüntülerinde yolların tespitinde derin öğrenmenin başarılı sonuçlar verdiği gözlemlenmiştir.

Anahtar Kelimeler: Derin Öğrenme, Konvolüsyonel Sinir Ağları, Yüksek Çözünürlüklü Hava Görüntüleri Analizi, Görüntü Segmentasyonu, CBS, Derin Öğrenme Unet Ağ Modeli, Yol Algılama

SUMMARY

ROAD DETECTION IN AERIAL IMAGES WITH DEEP LEARNING

Deep learning is an approach based on machine learning algorithms that allows modeling and classification by means of learning the properties for unknown data by modeling with the multi-layer artificial neural network on previously known data sets. As a result of the increase in the usability of artificial intelligence technology in the devices we use in our lives, deep learning has emerged and the studies conducted in this sense have continued to increase day by day.

Thanks to deep learning technologies, object recognition and analysis of these objects has accelerated the classification studies. The most basic deep learning algorithms were convulsive neural networks and multilayer artificial neural networks. Especially in the field of image processing, deep learning algorithms have been widely used in analysis studies. In this thesis, the analysis of high resolution aerial images with an important place in the field of deep learning by using Convolutional Neural Networks (CNNs) method and determination of roads will be discussed. With this study, it is ensured that the large-size aerial images can be processed and processed in order to achieve the workable image size. This data set was prepared and then the test was carried out with various deep learning network models. Creating data sets can be used in methods of deep learning activities for Turkey is presented. As a result of the estimation of the trained neural network, the geographic data of the road network in the image was obtained by using the algorithm developed by using HAVERSINE and BEARING formulas. As a result of this study, it has been observed that deep learning has yielded successful results in the detection of roads in high resolution satellite imagery.

Key Words: Deep Learning, Convolutional Neural Networks, High Definition Aerial Images Analysis, Image Segmentation, GIS, Deep Learning Unet Neural Model, Road Detection

ŞEKİLLER LİSTESİ

Şekil 3.1 Yapay Sinir Hücresi	14
Şekil 3.2 Yapay Sinir Ağı Yapısı	15
Şekil 3.3 Derin Ağlara ait Çeşitli Katman ve Model Yapısı	17
Şekil 3.4 Konvolüsyon İşlemi	20
Şekil 3.5 CNN’de Nöron Yapısı	21
Şekil 3.6 CNN Mimarisi.....	21
Şekil 3.7 Padding Örneği	23
Şekil 3.8 Pooling Örneği	24
Şekil 3.9 Dropout Kullanılmamış ve Kullanılmış Ağ Yapıları	25
Şekil 3.10 ReLU Fonksiyon Grafiği	26
Şekil 3.11 Sigmoid Fonksiyon Grafiği.....	27
Şekil 3.12 Softmax Fonksiyon Grafiği.....	27
Şekil 3.13 Yaygın Kullanılan Aktivasyon Fonksiyonları	28
Şekil 3.14 Tam Bağlı Katmanları Dönüştürerek Isı Haritaların Oluşturulması	33
Şekil 3.15 FCN Model Katmanları	34
Şekil 3.16 CNN İkili Sınıflandırma Çalışma Yapısı	35
Şekil 3.17 CNN Çoklu Sınıflandırma Örneği	35
Şekil 4.1 AlexNET Ağ Yapısı.....	37
Şekil 4.2 VGGNET Ağ Yapıları	38
Şekil 4.3 U-Net Ağ Yapısı	39
Şekil 5.1 8 Bitlik Uydu Görüntüsü.....	40
Şekil 5.2 8 Bitlik Uydu Görüntüsü(Solda) ve 8 Bitlik Uydu Görüntüsü Maskesi(Sağda)..	41
Şekil 5.3 8 Bitlik Resim(Solda) ve Bu Resmin Maskesi(Sağda)	42
Şekil 5.4 Hata Fonksiyon Değeri	43
Şekil 5.5 Model Doğruluğu	43
Şekil 5.6 Test Verisi(Solda) ve Tahmini(Sağda)	44
Şekil 5.7 Hata Fonksiyon Değeri	45
Şekil 5.8 Model Doğruluğu	45
Şekil 5.9 8 Bitlik Resim(Solda) ve Bu Resmin Maskesi(Sağda)	46
Şekil 5.10 Test Verisi(Solda) ve Tahmini(Sağda)	46
Şekil 5.11 Hata Fonksiyon Değeri	47
Şekil 5.12 Model Doğruluğu	47
Şekil 5.13 Test Verisi(Solda) ve Tahmini(Sağda)	48
Şekil 5.14 Hata Fonksiyon Değeri	49
Şekil 5.15 Model Doğruluğu	49
Şekil 5.16 Test Verisi(Solda) ve Tahmini(Sağda)	50
Şekil 5.17 Hata Fonksiyonu Değeri	51
Şekil 5.18 Model Doğruluğu	51
Şekil 5.19 Test Verisi(Solda) ve Tahmini(Sağda)	52
Şekil 5.20 Hata Fonksiyonu Değeri	53

Şekil 5.21 Model Doğruluğu	53
Şekil 5.22 Test Verisi(Solda) ve Tahmini(Sağda)	54
Şekil 5.23 Hata Fonksiyonu Değeri	55
Şekil 5.24 Model Doğruluğu	55
Şekil 5.25 Test Verisi(Solda) ve Tahmini(Sağda)	56
Şekil 5.26 Hata Fonksiyonu Değeri	57
Şekil 5.27 Model Doğruluğu	57
Şekil 5.28 Test Verisi(Solda) ve Tahmini(Sağda)	58
Şekil 5.29 Hata Fonksiyonu Değeri	59
Şekil 5.30 Model Doğruluğu	59
Şekil 5.31 Test Verisi(Solda) ve Tahmini(Sağda)	60
Şekil 5.32 Elazığ Uydu Görüntüsü(Solda) ve Photoshop ile Oluşturulan Maske(Sağda) ..	61
Şekil 5.33 Hata Fonksiyonu Değeri	61
Şekil 5.34 Model Doğruluğu	62
Şekil 5.35 Test Verisi(Solda) ve Tahmini(Sağda)	62
Şekil 5.36 Hata Fonksiyonu Değeri	63
Şekil 5.37 Model Doğruluğu	63
Şekil 5.38 Test Verisi(Solda) ve Tahmini(Sağda)	64
Şekil 5.39 Geoserver	65
Şekil 5.40 Yol Verilerinin WMS Katmanı Üzerindeki Görüntüsü	65
Şekil 5.41 WMS ve Google Maps Katmanlarının Birleştirilmesi.....	66
Şekil 5.42 Yol Verilerinin Maskesi.....	66
Şekil 5.43 Görüntü Kesiti(Solda) ve Görüntü Kesitinin Maskesi(Sağda)	67
Şekil 5.44 Hata Fonksiyonu Değeri	68
Şekil 5.45 Model Doğruluğu	68
Şekil 5.46 Test Verisi(Solda) ve Tahmini(Sağda)	69
Şekil 5.47 Hata Fonksiyonu Değeri	70
Şekil 5.48 Model Doğruluğu	70
Şekil 5.49 Test Verisi(Solda) ve Tahmini(Sağda)	71
Şekil 5.50 Dört noktasının enlem ve boylam değeri bilinen Test Verisi(Solda) ve Tahmini(Sağda)	72
Şekil 5.51 Dört noktasının enlem ve boylam değeri bilinen Tahmini Veri (Solda) ve Geodata çıkarılan Görüntü (Sağda)	73

TABLÖLAR LİSTESİ

Tablo 5.1 Model Eğitim ve Loss Değeri.....	42
Tablo 5.2 Segmentasyon Sonuçları.....	44
Tablo 5.3 Model Eğitim ve Loss Değeri.....	44
Tablo 5.4 Model Eğitim ve Loss Değeri.....	47
Tablo 5.5 Segmentasyon Sonuçları.....	48
Tablo 5.6 Model Eğitim ve Loss Değeri.....	49
Tablo 5.7 Model Eğitim ve Loss Değeri.....	50
Tablo 5.8 Segmentasyon Sonuçları.....	52
Tablo 5.9 Model Eğitim ve Loss Değeri.....	52
Tablo 5.10 Segmentasyon Sonuçları.....	54
Tablo 5.11 Model Eğitim ve Loss Değeri.....	54
Tablo 5.12 Model Eğitim ve Loss Değeri.....	56
Tablo 5.13 Segmentasyon Sonuçları.....	58
Tablo 5.14 Model Eğitim ve Loss Değeri.....	58
Tablo 5.15 Model Eğitim ve Loss Değeri.....	61
Tablo 5.16 Model Eğitim ve Loss Değeri.....	63
Tablo 5.17 Segmentasyon Sonuçları.....	64
Tablo 5.18 Model Eğitim ve Loss Değeri.....	67
Tablo 5.19 Model Eğitim ve Loss Değeri.....	69
Tablo 5.20 Segmentasyon Sonuçları.....	71

KISALTMALAR LİSTESİ

AI	:Artificial Intelligence
AE	:Auto-Encoders
BM	:Boltzmann Machine
CNNs	:Convolutional Neural Networks
CNV	:Choroidal Neo Vascularization
DL	:Deep Learning
DBM	:Deep Boltzmann Machine
DBN	:Deep Belief Network
DNN	:Deep Neural Network
FCNNs	:Fully Convolutional Neural Networks
GMMs	:Gaussian Mixture Models
GPU	:Graphics Processing Unit
ILSVRC	:ImageNET Large Scale Visual Recognition Competition
K-NN	:K Nearest Neighbours
MaxEnt	:Maximum Entropy
MIL	:Multiple Instance Learning
MLPs	:Multi Layer Perceptrons
RBM	:Restricted Boltzmann Machine
RNN	:Recurrent Neural Network
rRNN	:Recognizing Recurrent Neural Network
SIFT	:Scale-Invariant Feature Transform
SVM	:Support Vector Machine
YSA	:Yapay Sinir Ağları

1. GİRİŞ

Dünya nüfusundaki artış beraberinde birçok sorunu da getirmektedir. Kentsel sorunlar, trafik sorunları, sağlık sorunları, kirlilik vb. Bu sorunları ele aldığımızda ciddi anlamda büyük veri kaynaklarına erişmek mümkündür. Bu veri kaynaklarını kullanmak için yürütülen çalışmaların çoğunlukla mekânsal verilerin kullanımıyla ilgili olması, CBS teknolojisinin büyük bir hızla gelişmesine ve kullanılmasına olanak sağlamaktadır. CBS'nin gün geçtikçe önemli hale gelmesi ve çok sayıda disiplinin CBS teknolojisinden yararlanmaya başlamasıyla uygulama çeşitliliği de artmıştır [1]. CBS bilinenin aksine sadece harita işlemleri ile harita çizmekten ibaret olmayıp aynı zamanda, verilerin bilgisayar ortamında saklanarak, bilinen kullanılmış yöntemlere göre daha hızlı ve dinamik bir şekilde değerlendirilmesine olanak sağlamaktadır. CBS yardımıyla var olan verilerden yeni veriler elde edilebilmekte ve bu verilere bağlı olarak farklı alanlarda çalışmalar yürütülebilmektedir. Elde edilen CBS alanındaki zengin verilerin işlenmesinde birçok yöntem mevcut olup en çok kullanılan ve en doğru sonuçlar veren yöntem makine öğrenmesi yöntemleridir.

Yaşadığımız şehirler sürekli genişliyor, büyüyor ve değişiyor. Bu değişim yeni altyapılar, yeni binalar ve mahalleleri beraberinde getiriyor. Tüm bu değişimlerin izlenmesi ve takip edilmesi ve güncellenmesi her zaman ciddi bir emek ve dikkat gerektiriyor. Haritalarımızı eski usul yöntemlerle güncellemek yerine yeni uydu görüntüleri alabilseydik ve tüm haritalarımızı çabucak güncellemek için Derin Öğrenmeyi kullansaydık, bu alanda çalışan insanlar için işlemlerin ne kadar rahat olacağını gözlemleyebilirdik. Örneğin; harita ya da yüksek çözünürlüklü hava görüntülerinde Nesne algılama, yüksek çözünürlüklü görüntü verilerinde yol bulma, yeryüzü şekilleri tanımlama, arazi örtüsü bilgisine ulaşmanın en önemli adımlarından biridir.

Makine öğrenimindeki Derin Sinir Ağları, hava görüntülerindeki piksel tabanlı otomatik nesne algılamasında kullanılan ve doğru sonuçlar veren mimarilerden biridir. Derin öğrenme ağları genellikle birden çok gizli katman ile sunulmaktadır. Gizli katman, ağın eğitim verilerinin dâhili soyut temsilini depoladığı yerdir. Derin öğrenmede, gizli katmanlar, geleneksel sinir ağlarıyla karşılaştırıldığında tamamen farklı bir biçimde hesaplanır. Daha belirgin olarak, derin bir ağdaki her katman, denetiminin öğrenme

algoritmasıyla önceden eğitilmiş olup, girişin veya önceki katmanın çıktısının doğrusal olmayan bir dönüşümü ve girişinden daha soyut özelliklerin yakalanmasıyla sonuçlanır. Sonra son eğitim aşamasında, derin mimari gradyan tabanlı optimizasyonla denetlenen bir eğitim kriterine göre ayarlanır. Derin öğrenme kavramı, alt düzey özellikler bileşiminin uygulanmasıyla özellikleri daha üst seviyede eğitmek üzere tasarlanmıştır. Bengio ve LeCun'ın önerdiği gibi, derin öğrenen ağlar, bir dizi işlem kademesi aracılığıyla alt düzey özellikler ile üst düzey kavramlara yönelik soyutlamaları otomatik olarak keşfedebilir [2]. Burası, alt düzey soyutlamaların verilerin parçaları ile daha sıkı ilişkilendirildiği ve üst düzey soyutlamalar, gerçek ve anlamlı kavramlarla doğrudan bağlantılı olduğu yerdir. Böyle derin bir mimari kullanmanın bir avantajı, farklı bir soyutlama seviyesinin çok sayıda özellikten küçük bir alt küme altına odaklanmasıdır. Öğrenilecek bilgiler tek bir nöron tabakasında bulunmamakla birlikte, çok katmanlı olarak dağıtılmaktadır. Böyle dağıtılmış bir temsil, derin öğrenme ağlarının öğrenme için daha güçlü bir kapasiteye sahip olmasını sağlar ve geleneksel makine-öğrenme yöntemlerine kıyasla çok daha iyi genellemeler üretebilir. Ayrıca, çok seviyeli ve verilerin dağıtılmış bir temsili üzerine kurulu bir mimari, derin öğrenme ağlarının farklı problem alanlarında paylaşılabilen ara temsilleri öğrenmesine olanak tanır. Bu, ara tasvirler olarak öğrendiği bilginin ortak bir havuzdaki alt düzey ara özelliklerin birleştirilmesiyle yeni yüksek seviye özelliklerin öğrenilebileceği tekrar kullanılabilir olduğu anlamına gelir [2].

Makine öğrenimindeki derin ağlar, otomatik nesne algılamada yetenekler göstermiştir. Özellikle, nesne algılamada başarılı sonuçlar veren Evrişimsel Sinir Ağı kullanılarak trafik işareti tespiti [3], biyolojik görüntü segmentasyonu [4], yüz tanıma, yaya ve insan vücudu bulma [5,6] alanlarında çalışmalar mevcuttur. Büyük veri setleri üzerinde çalışmaya uygun olan bu yöntem tespit edilen nesnelerin kalitesini de arttırmaktadır. Bu çalışmalar, kullanılan alanlarda ciddi anlamda sonuçlar vermiş ve uygulanan işlemleri otomatikleştirmiştir.

Derin öğrenmenin esası, üst düzey özellikler veya faktörlerin alt düzeydeki özelliklerden tanımlandığı gözlem verilerinin aşamalı olarak özelliklerini veya temsillerini hesaplamaktır. Derin öğrenme yöntemleri ailesi, sinir ağları, aşamalı olasılık modelleri ve denetlenmeyen ve denetlenen özellik öğrenme algoritmalarını kapsamaktadır ve giderek daha zengin hale gelmiştir [7].

Derin öğrenme otomotiv, harita, cbs, fen bilimleri, tıp, pazarlama ve eğlence gibi alanlarda yaygın bir şekilde kullanılmaya başlanmıştır. CBS alanında sürekli olarak artan veri yoğunluğundan, bu verilerin analiz ve saklanması kaynaklı olarak bilgisayar destekli sistemlere gün geçtikçe daha çok ihtiyaç duymaktadır. Derin öğrenme algoritmaları, CBS alanında uzun süreçlere yayılan harita çizimi, yeni yolların haritalara eklenmesi, verilerin analizinde kullanılabilir bir teknolojidir. Son yıllarda, dünya gözlemi için yüksek çözünürlüklü uzaktan algılamayı analiz etmek, birçok alan için önemli bir rol oynamıştır ve yol çıkarma, çok fazla dikkat gösterilen anlamlı tekniklerden biridir [8].

Bu tezin amacı yüksek çözünürlüklü hava görüntülerinde yol segmentasyonu yapan derin öğrenme tabanlı bir sistem gerçekleştirmektir. Yol segmentasyonunda derin öğrenmenin başarılı sonuçlar verip vermediği araştırılmıştır. Bu çalışma ile büyük boyutlu ve yüksek çözünürlüklü hava görüntülerinin işlenebilir resim boyutuna gelmesi için bölümlenerek işlenmesi ve veri setlerinin elde edilmesi sağlanmıştır. Hazırlanan bu veri setleri eğitilerek daha sonra test amaçlı ayrılan verilerin çeşitli derin öğrenme ağ modelleri ile istenilen yolların tespiti gerçekleştirilmiştir. Bu veri setlerinden önceden etiketlenmiş bir miktar veri alınıp oluşturulan ağ eğitip geri kalan test verisinde başarısı ölçülmüştür. Bunu gerçekleştirmek için derin öğrenme yöntemlerinden (FCNN) yöntemi kullanılarak başarılı sonuçlar alınmıştır. Resim segmentasyonunda başarılı sonuçlar veren bir FCNN ağı olan U-NET kullanılmıştır.

Genel başlıklarıyla bu tezin;

2.bölümünde, tez çalışması kapsamında literatür çalışması olarak, hava görüntüleri kullanılarak yapılmış sınıflandırma ve segmentasyon işlemleri için geliştirilen yöntemler incelenmiştir.

3.bölümünde, Derin öğrenme anlatılmaktadır. Çalışmanın konusu kapsamındaki kavramlara değinilmiş, tez çalışması süreci içerisinde yapılan algoritma tasarım ve geliştirme süreçleri öncesinde yeter düzeyde bir teorik altyapı kazanması sağlanmıştır.

4.bölümde, Derin öğrenme ağ yapıları ve modelleri anlatılmıştır.

5.bölümde, Derin öğrenmede ile eğitilen veri setleri ve sonuçları anlatılmıştır.

6.bölümde ise çalışmanın sonuçları ve başarı değerleri yönteminin olumlu ve olumsuz yanları ifade edilip ne gibi gelişme yöntemleri ile daha iyi sonuçlar alınabileceği anlatılmıştır.



2. LİTERATÜR ÖZETİ

Coğrafi bilgi sistemlerinde (CBS) hava görüntülerinin analiz edilmesi için derin öğrenmeye dayanan yöntemler giderek artmaktadır. Çeşitli bilimsel alanlarda analiz ve sınıflandırmaya yardımcı olan bu yöntem birçok kişi tarafından kullanılarak deneyler yapılmıştır. Yüksek çözünürlüklü uydu görüntülerinin analizi trafik yönetimi, şehir ve bölge planlamasında ve ya yol çalışmaları ile ilgili alanlarda üzerinde çalışma yapılması için önemli bir analiz konusudur.

CBS alanında görüntü işleme yöntemlerinin yanı sıra derin öğrenme yöntemlerinin kullanımı hem performans hem de daha doğru sonuç verme konusunda hız kazanmıştır. CBS alanında derin öğrenme yöntemlerinin kullanım alanları; görüntü sınıflandırma, nesne algılama, segmentasyon vb.dir.

Ale ve arkadaşları [9], yol görüntülerini analiz ederek yol hasarını etkin bir şekilde saptamak için bir yöntem önermişlerdir. Akıllı telefonlardan elde edilen görüntülerde yol hasarının olup olmadığı bilgisi için RetinaNet modeli kullanılarak eğitim gerçekleştirmişlerdir. Sonuç olarak, yol hasarı tespiti için gelişmiş sınıflandırma oranını ortaya koymuşlardır.

Zhang, Lei ve ark. [10], kaldırım çatlaklarının otomatik tespiti için derin öğrenme temelli bir yöntem önermişlerdir. Akıllı telefon ile elde edilen 3264×2448 boyutundaki 500 görüntüden oluşan veri seti üzerinde denetimli bir derin evrimsel sinir ağı olan ConvNet modelini eğitmişlerdir. Sonuç olarak, kaldırım çatlaklarının tespiti için gelişmiş sınıflandırma oranını ortaya koymuşlardır.

Wang, Yanbo J. ve ark. [11], yol hasarı tespiti için basit ama etkili bir nesne algılama modeli önermişlerdir. Yol hasarı tespiti için Faster-RCNN ve SSD modellerini eğitmişlerdir. Uygulama sonucunda alınan başarı değeri IEEE BigData Yol Hasar Tespiti çalışmalarında iyi sonuçlar elde etmiştir.

Kulkarni ve arkadaşları [12], trafik ışıklarının güvenilir bir şekilde algılanması ve tanınması için derin bir sinir ağı temelli model önermişlerdir. Hindistan Trafik Sinyallerine göre farklı trafik sinyalleri görüntüleri içeren veri kümesi üzerinde evrimsel sinir ağı olan

R-CNN modelini eğitmişlerdir. Sonuç olarak, trafik ışığını doğru sınıf tipiyle tespit ederek başarılı sonuçlar ortaya koymuşlardır.

Zhang ve arkadaşları [13], internetten indirilen 150 yol görüntüsünde karayolu aracı tespiti için derin sinir ağı (DNN) yöntemi önermişlerdir. Sonuç olarak, karayolu aracı tespiti için başarılı sonuçlar ortaya koymuşlardır.

Buslaev, Alexander ve ark. [14], evrimli bir sinir ağına dayanan otomatik yol çıkarma için bir yaklaşım önermişlerdir. 6226 tane 1024x1024 boyutunda RGB girişli veri setini ResNet-34 ve U-Net Modelleri ile eğitmişlerdir. Tüm eğitimi gerçekleştirmek için yalnızca

bir GTX 1080 veya 1080ti ekran kartı kullanmaya izin veren ve oldukça hızlı tahminlerde bulunan orta derecede bellek kullanmışlardır.

Li, Jun ve ark. [15], hedefin varlığını ve hedefin bölgeye göre geometrik niteliklerini (yer ve oryantasyon) eşzamanlı olarak tespit eden çok görevli, derin bir evrimsel sinir ağı (DCNN) geliştirmişlerdir. Yol sınırlarını otomatik olarak algılamayı hedeflemişlerdir.

Horita, Luiz Ricardo T., ve Valdir Grassi [16], yol işaretlemesi tespiti ve segmentasyonu için enkoder kod çözücü mimarisine sahip, derinlemesine bir Tam Dönüştürmeli Sinir Ağı (FCNN) temelli alternatif bir yaklaşım önermişlerdir. Yol işaretleri trafik yönlendirmede kullanılan ana işaretlerden biri olduğundan, onları algılamak özerk araçlar için çok önemli bir görevdir. Sonuç olarak, önerilen yaklaşımın herhangi bir yol işaretleme tipini yüksek doğrulukta saptayabildiğini ve böylelikle pürüzsüz bir segmentasyona neden olduğunu ortaya koymuşlardır.

Cisek, Daniel ve ark. [17], hava görüntülerinden park yeri çıkarma işlemini otomatikleştirmek için bir yöntem sunmuşlardır. New York State GIS Clearinghouse veri seti üzerinde evrimsel sinir ağı (CNN) ve çeşitli morfolojik analiz araçlarının kombinasyonunu kullanarak 4000 adımlı eğitim gerçekleştirmişlerdir. Sonuç olarak, otomatik park yeri tespiti için gelişmiş sınıflandırmayı ortaya koymuşlardır.

Xia, Wei ve ark. [8], standart yol bölgesini yüksek doğrulukla elde etmek için yol çıkarmaya yönelik bir yaklaşım önermişlerdir. Taşımacılık endüstrisi tarafından yapılan yol tasarımı ve inşaat özellikleri kullanılarak, yol nesnelerini farklı sınıflara ayırmışlardır. Sınıflara ayrılan görüntüler üzerinde derin evrimsel ağ ile segmentasyon yapmışlardır.

Jan, Zohaib ve ark. [18], video verilerini analiz edebilen ve yol güvenliği ile koşullarını değerlendirebilen yeni bir derin öğrenme yaklaşımı önermişlerdir. Avustralya Yol Değerlendirme Programı (AusRAP) için yol özelliklerini otomatik olarak tanımlamak için bir evrişimli sinir ağı tabanlı segmentasyon ve sınıflandırma tekniği ve mesafeye yönelik yeni bir yakınlık ölçüm tekniği geliştirmişlerdir. Sonuç olarak, eğitilen sinir ağı Queensland ulaşım ve ana departmanı tarafından toplanan yol video verileri üzerinde değerlendirilmiş ve yollar tespit edilmiştir.

Yadav, Shashank ve ark. [19], bir şartlı rasgele alan (CRF) çerçevesinden önce dayalı, derin evrişimli sinir ağlarının (CNN'ler) ve renkli çizgiler modelinin kombinasyonuna dayanan yeni bir teknik önermişlerdir. CNN yol dokusunu öğrenirken, renk çizgileri modeli değişen aydınlatma koşullarına uyum sağlamasına izin vermiştir. Sonuç olarak, geliştirdikleri tekniklerinin işaretli yol bölümlene problemde son teknoloji segmentasyon tekniklerinden daha iyi performans gösterdiğini göstermişlerdir.

Singh ve arkadaşları [20], hava görüntülerinden yol çıkarımı yapmak için derin bir evrişim ağı yöntemi önermişlerdir. Massachusetts yol veri setini U-Net modeli ile eğitmişlerdir. Farklı katman sayılarına sahip aktivasyon fonksiyonları kullanmışlardır. Kullanılan aktivasyon fonksiyonlarından LeakyReLU en iyi hassas ölçümleri gösterirken, ELU en iyi hatırlama önlemlerini göstermiştir. Bu iki U-net modelini iki aşamalı bir CNN modeli olarak birleştirmek, tartışılan diğer modelleri doğruluk, hassasiyet, hatırlama ve F-skoru bakımından daha iyi performans gösterdiği sonucunu çıkarmışlardır.

Zhang ve arkadaşları [21], hava görüntülerinden karayolu çıkarmak için bir segmentasyon yöntemi önermişlerdir. Evrişimsel bir sinir ağı olan U-Net'in güçlü yönlerini birleştiren semantik bir segmentasyon sinir ağı kullanmışlardır. Massachusetts yol verileri ile eğitilen ağ U-Net ve derin ResNet ağları ile sonuçları kıyaslanmıştır. Sonuç olarak, önerdikleri yöntemin tüm karşılaştırma yöntemlerinden daha iyi performans gösterdiklerini sunmuşlardır.

Proulx-Bourque ve arkadaşları [22], topografik özelliklerin yüksek çözünürlüklü optik uydu görüntülerinden çıkarılması için evrişimli bir sinir ağının kullanılmasını sunmuşlardır. Yol, bina ve su yolları çıkarımı için U-Net modeli ile eğitim yapmışlardır. Eğitilen model, ticari ve endüstriyel binalar ve eğitim verilerinde temsil edilen asfaltsız yollar hariç, genellikle doğru tahminler ürettiğini sunmuşlardır. Geliştirdikleri uygulama derin öğrenme

kullanarak topografik özellik çıkarımı için ve ulusal ölçeğe genişleme için bir eğitim veri tabanının geliştirilmesi için perspektif sağlamışlardır.

Huang ve arkadaşları [23], yüksek çözünürlüklü hava görüntülerinden çok ölçekli yapısal özelliklere ve destek vektör makinelerine (SVM'ler) dayanan yeni bir yol algılama sistemi önermişlerdir. Sonuç olarak iki IKONOS multispektral veri seti üzerinde deneyler yapılmış ve deney sonuçlarının önerdikleri yöntemi doğruladığını göstermişlerdir.

Mnih ve arkadaşları [24], geniş bir çerçeveye bakan milyonlarca eğitilebilir ağırlığa sahip bir sinir ağı kullanarak yolları tespit etmeyi önermişlerdir. Ağ büyük miktarda veri ile eğitilmiştir. Tahmini performans, son zamanlarda geliştirilen denetimsiz öğrenme yöntemleri kullanılarak ve çıktı etiketlerinin yerel mekânsal tutarlılığından yararlanarak özellik detektörleri başlatılarak büyük ölçüde iyileştirilebileceğini göstermişlerdir. Sonuç olarak, geliştirdikleri metodun kentsel veri seti üzerinde başarılı sonuçlar verdiğini sunmuşlardır.

Cheng, Guangliang ve ark. [25], yol tespiti ve merkez hattı çıkarma görevleriyle aynı anda başa çıkabilmek için yeni bir derin model, yani basamaklandırılmış uçtan uca bir evrimsel sinir ağı olan CasNet'i önermişlerdir. Google Earth'ten çok yüksek çözünürlüklü 224 adet görüntü üzerinde manuel etiketleme işlemi yapılmış ve ağ eğitilmiştir. Sonuç olarak, eğitilen CasNet ağının diğer karşılaştırma yöntemlerinden neredeyse 25 kat daha hızlı olduğunu sunmuşlardır.

Saito ve arkadaşları [26], evrimsel bir sinir ağı (CNN) tabanlı bina ve yol çıkarma sistemi önermişlerdir. Girdi olarak ham görüntüdeki piksel değerlerini alıp ve öngörülen üç kanallı etiket görüntülerini (bina-yol-arka plan) çıkarmışlardır. CNN'leri kullanarak, hem özellik çıkarıcıları hem de sınıflandırıcıları otomatik olarak oluşturmuşlardır. Aynı anda birden fazla nesneyi çıkarmak için tek bir CNN'yi verimli bir şekilde eğitmek için yeni bir teknik önermişlerdir. Sonuç olarak, tahminleme performansını iyileştirdiklerini sunmuşlardır.

Alshehhi ve arkadaşları [27], hiyerarşik grafik tabanlı görüntü segmentasyonuna dayalı yüksek çözünürlüklü görüntülerden yollar çıkarmak için yeni bir yöntem sunmuşlardır. Önerilen yöntemi ibraz etmek ve diğer benzer yaklaşımlarla karşılaştırmak için yüksek çözünürlüklü görüntülerin üç zorlu veri kümesi üzerinde deneyler

yapmışlardır. Sonuç olarak, kentsel alanlarda kara yolu çıkarma için önerilen yöntemin üstünlüğünü ve üstün performansını göstermişlerdir.

Liu, Bo ve ark. [28], yönelimli matematiksel morfolojiye (MM) ve önceki bilgilere dayanan yüksek çözünürlüklü gri tonlamalı görüntülerden ana yolları çıkarmak için yeni bir yöntem sunmuşlardır. Önerilen yöntem yol özellik çıkarımı için diğer yaygın kullanılan tekniklerle karşılaştırmak için iki ZiYuan-3 görüntü ve bir QuickBird yüksek çözünürlüklü gri tonlamalı görüntü üzerinde deneyler yapmışlardır. Sonuç olarak, kentsel ana yol özellik çıkarımı için önerilen yöntemin geçerliliğini ve daha iyi performansını göstermişlerdir.

Sujatha ve arkadaşları [29], karayolu merkez hattının yüksek çözünürlüklü uydu görüntüsünden otomatik olarak çıkarılması için bir yöntem önermişlerdir. Önerilen çalışmada yol bölgesinin bölümlenmesi, bağlı yol bileşenini bölümlenmiş yol bölgesinden çıkarmak için kullanılan bağlı bileşen esaslı işlemler ve morfolojik işlem kullanılarak istenmeyen yol dışı piksellerin kaldırılması yapmışlardır. Sonuç olarak, önerilen çalışmanın performans ölçümleri diğer yöntemlerle karşılaştırılmış ve bu karşılaştırma önerilen yöntemin diğer yöntemlerden çok daha iyi sonuçlar verdiğini kanıtlamışlardır.

Cheng, Guangliang ve ark. [30], yol merkez çizgisi çıkarma için yarı denetimli segmentasyon ve çok ölçekli filtreleme (MF) ve çok yönlü maksimum baskılama (M-NMS) (MF ve M-NMS) esas alınarak oluşturulan merkez hattı çıkarma yöntemi sunmuşlardır. Genel bir veri setindeki deneysel sonuçlar, önerilen yöntemin elde edildiğini göstermişlerdir.

Song ve arkadaşları [31], sınıflandırma ve görüntü segmentasyonundan türetilmiş nesne özellikleri için piksel spektral bilgisini kullanan yol çıkarma için benzersiz bir yaklaşım geliştirmişlerdir. Destek vektör makinesi (SVM) ve Gaussian maksimum olasılığından (GML) yalnızca görüntüyü iki kategori grubuna sınıflandırmak için kullanılmıştır. Destek vektör makinesi (SVM) Gaussian maksimum olasılığından (GML) daha yüksek doğruluk elde ettiğini sunmuşlardır. Sonuç olarak, önerilen yaklaşımın hem kırsal hem de kentsel alan özelliklerinden oluşan görüntülerle iyi çalıştığını göstermişlerdir.

3. DERİN ÖĞRENME

Derin öğrenme, makine öğreniminin belirli bir alt alanıdır, verilerin giderek artan anlamlı temsillerin ardışık katmanlarını öğrenmeye vurgu yapan yeni bir öğrenme şeklidir. Tarihsel olarak, derin öğrenme yapay sinir ağı alanında yapılan araştırmalar ve geliştirmeler sonucunda ortaya çıkmıştır. Derin öğrenmenin mimarisi insan beyni gibi çalışan ve karar verebilen yapılar oluşturmayı amaçlar. Derin öğrenme yalnızca sayısal verilerin işlenmesi üzerine değil, piksel, text ve ses verilerinin işlenmesini de sağlayabilir.

İnsanın beyninin çalışma yapısından ilham alan derin öğrenme algoritmaları, beyin neokorteksinin veriyi karmaşık bir hiyerarşiden geçirdiğini gösteren sinir bilimi bulgularını kullanır [32]. Nöral ağ terimi, nörobiyolojiye bir atıftır, ancak derin öğrenmedeki bazı merkezi kavramlar, kısmen, bizim beyin anlayışımızdan esinlenerek geliştirilmiş olsa da, derin öğrenme modelleri beyin modelleri değildir. Derin öğrenme yöntemleri, temellerini sinir ağlarından alır. Derin öğrenmedeki 'derin' tabiri, yaklaşımın ulaştığı her türlü derinlemesine bir referans değildir. Aksine birbirinin üzerine yığılan katmanları ifade eder. Derin bir öğrenme mimarisi, çoğu zaman onlarca hatta yüzlerce birbirini izleyen temsil katmanları içerir ve bunların hepsi otomatik olarak eğitim verilerine maruz kalmaktan öğrenilir. Derin öğrenme, verilerden temsilleri öğrenmek için matematiksel bir çerçevedir. Örneğin, bir görüntüyü ele aldığımızda piksellerden oluştuğunu biliriz. Derin bir model, piksellerden oluşan bu resmin köşelerini, içerisindeki nesnelerini veya nesneleri üst üste yığılmış katmanlardan yayılan olarak öğrenir. Derin öğrenme yaklaşımları, görüntü sınıflandırması, konuşma tanıma, görüntü işleme, doğal dil işleme vb. konularında kullanılmaya uygundur ve bunlar sıkı mimarilerle kıyaslandıklarında çok daha güçlüdürler [33].

Derin öğrenme makine öğreniminin oldukça eski bir alt alanı olmasına rağmen, insanlara doğal ve sezgisel görünen, ancak makineler için uzun süredir zor olan becerileri içeren problemleri görme ve duyma gibi algısal problemler üzerinde kayda değer sonuçlarla, sahada bir devrimden kısa bir şey elde etmemiştir. Son zamanlarda bu alanda yapılan çalışmalarda devresel ve biyolojik karmaşıklığı yüksek olan yapılardaki derin öğrenme kullanımının artması, özellikle problemin baya zor ve karmaşık matematiksel işlemleri analiz etme de diğer mimarilere göre daha yüksek oranlı doğru sonuçlar verdiği

çalıřmalarda gözlemlenmiřtir. Temel yapısı gereęi aslında bir yapay sinir aęı olan derin öğrenmenin derin olarak adlandırılmasının sebebi bu YSA'nın mimarisinden dolaydır. Çalışmalar sonucunca ortaya çıkan ilk yapay sinir aęı mimari sadece birkaç katmandan oluşuyordu. Günümüzde ise katman sayıları kullanılan model yapısına göre deęişmekte ve gittikçe artmaktadır. Yakın zamana kadar yapay sinir aęı eğitmenin çok zor olduęu düşünölüyordu. Bunun temel sebebi olarak donanımsal eksiklikler olduęu düşünölmüřtür. Teknolojinin gelişimi ile zor olduęu düşünölen eğitimin temel eksiklięi giderilmiş ve modelin katman sayısı atılarak derin aę yapıları oluşturulmaya başlanmıřtır. Deneysel açıdan, derin aęlar genellikle bir veya iki gizli katmana sahip sinirsel aęlardan daha iyi veya daha kötü deęildir [34]. Derin öğrenme dięer yapılara kıyasla: karmařık yapıdaki yapıları çözümleme, az çevre (kullanıcı) müdahalesi ve yüksek seviye soyutlamaları analiz etmede, çok büyük örnekleri ve etiketsiz verileri deęerlendirebilmede dięer yöntemlere göre daha iyi başarılar göstermektedir.

3.1. Makine Öğrenmesi

Makine Öğrenmesi de yapay zekânın bir alt yöntemidir. Elimizde var olan verilerle kendi kendine eğitme, gözlem ve analiz ile çıkarım yapabilme özellięi vardır. Makine öğrenmesi bilgisayarın insan öğrenme yapısına benzer “öğrenme” işlemini sağlayacak algoritma ve yöntemlerinin geliştirilmesi üzerine çalışan bir alandır [35]. Bir makine öğrenme sistemi, açıkça programlanmış olmaktan ziyade eğitilmiştir. Bir görevle ilgili birçok örnek sunulmuş ve bu örneklerde, sistemin, görevi otomatikleřtirmek için kurallar oluşturmasına olanak tanıyan istatistiksel yapıyı bulmuřtur. Örneęin, fotoğraf albümünüzde etiketleme görevini otomatik hale getirmek isterseniz, insan tarafından zaten etiketlenmiş birçok resim örneęine sahip bir makine öğrenme sistemi sunabilir ve sistem belirli resimleri belirli etiketlerle ilişkilendirmek için istatistiksel kurallar öğrenir. Makine öğrenmesinde öğrenme yöntemlerini iki ana başlıkta gözetimli ve gözetimsiz öğrenme olarak sınıflandırılabiliriz.

Pratik makine öğreniminin çoęunluęu gözetimli öğrenmesini kullanır. Gözetimli makine öğrenmesi giriş deęerine karşılık alınan çıkıř deęeri için fonksiyon üreten bir makine öğrenmesi teknięidir. Amaç, eşleme fonksiyonunu yaklaşık olarak tahmin etmektir ki, yeni giriş verisi olduęunda, bu veriler için çıkıř deęişkenleri tahmin edilebilir. Eğitim verisi hem girdilerden hem çıktılardan oluşur. Gözetimli öğrenme problemleri için fonksiyonlar, eğri uydurma (regression) veya sınıflandırma (classification) algoritmaları ile

belirlenebilir. Ayırık olan sınıflar arasında uzak ya da yakın gibi bir durum söz konusu değildir. Regresyon, öğrendiklerine göre gerçekçi tahminleme yapar. Örneğin, sınıflandırma algoritmalarında çıktı verisi kırmızı ve ya beyazdır, hastalıklı ve hastalıklı değildir. Regresyon algoritmalarında çıktı verisi para birimi, ağırlık yaşı vb. şeklinde gerçek değerlerdir.

İşaretlenmemiş verilerden yola çıkarak bilinmeyen bir durumu tahminlemek için bir fonksiyon kullanan makine öğrenmesi Gözetimsiz öğrenme tekniğiyle sağlanır. Gözetimsiz öğrenmenin amacı, veriler hakkında daha fazla bilgi edinmek için verilerin altta yatan yapısını veya dağıtımını modellemektir. Gözetimsiz öğrenmede doğru sonuç yoktur. Bu algoritmalar, verilerdeki değişik durumu keşfetmek ve sergilemek için kendi kendilerine öğrenir. Gözetimsiz öğrenme Kümeleme (Clustering) ve Boyut Azaltımı (Dimensionality Reduction) olarak iki alt başlığa ayrılabilir. Kümeleme, verilerdeki homojen gruplamaları keşfetmek için kullanılır. Örneğin; bir marketteki meyve reyonunda bütün yeşil meyveleri bulmaktır. Sınıflamaya yapısına benzemesine rağmen farklı olarak aynı kümede değil farklı kümelemede yakınlık arar. Boyut azaltımı ise, gözlemlerin var olan özellik değerini en aza indirgeyip ve özünü çıkartır, en iyi şekilde öğrenme gerçekleşmesini amaçlar.

Gözetimsiz ve gözetimli öğrenme dışında Yarı Gözetimli Öğrenme, Yapay Sinir Ağları, Ödüllü Öğrenme, Genetik algoritmalarla da öğrenme yapabiliriz. Makine öğrenmesinde kullanılan terimler şu şekildedir;

- **Eğitim Verisi:** Davranışlı öğrenmede fonksiyona giriş parametresi olarak kullanılır ve eğitim tamamlandıktan sonra çıkış vektörleri oluşturur.
- **Aşırı Beslenme:** Verinin fazlasıyla (aşırı) büyük ya da küçük olması durumuna aşırı beslenme denir. Gürültü aşırı beslenmeye sebep olabilecek bir duruma örnek verilebilir.
- **Çapraz Doğrulama:** Eğitilmek için kullanılacak olan veriler kendi içerisinde kümelendirilir ve bu kümelerden bir tanesi alt küme diğerleri ise doğrulama yapma amacıyla seçilir. Yapılan bu işlemler çapraz bir şekilde alt kümeler üzerinde tekrar yapılır.
- **Gürültü:** Konuyla ilgisi olmayan anlamlandırılmayan verilerdir. Makine öğrenmesinde gürültüyü azaltmak öncelikli işlemlerden biridir.

3.2. Yapay Sinir Ağları

Yapay sinir ağları, makine öğrenmede kullanılan temel araçlardan biridir. Adlarının “nöral” kısmı öne sürdüğü için, insanların öğrendiği yolu çoğaltmayı amaçlayan beyin ilham sistemleridir. Sinir ağları, giriş ve çıkış katmanlarının yanı sıra girdiyi çıktı katmanının kullanabileceği bir şeye dönüştüren birimlerden oluşan gizli bir katmandan oluşur. 1900’lerden bu yana yapay zekanın alt kolu olarak araştırmacılar tarafından ilgi duyulan bir hal almış ve 1970’li yıllardan sonra önemi artan bu alanda çalışmalar yapılmıştır. 1943’de W. McCulloch ve W.Pitts tarafından önerilen ilk yapay sinir ağı modeli, insan beyninin hesaplama yeteneğinden etkilenecek, basit bir sinir ağı modeli geliştirmişlerdir [54].Yapay sinir ağları sınıflandırma, tahminleme, kümeleme, fonksiyon yaklaşımı, kontrol, optimizasyon, desen uyumu ve arama çalışmalarında kullanılmaktadır.

Tıp, Elektronik, Güvenlik ve Kredi Uygulamalarında, Borsa Tahminlerinde, sigortacılıkta, Üretim alanında, Görüntü Sıkıştırma, ürün kalite ve kontrolünde, eğlence alanında reklam analizinde, karakter tanımda, müşteri değerlendirilmesinde, endüstri alanında, askeri alanda silah hedef tespitinde vb. birçok alanda kullanılmaktadır.

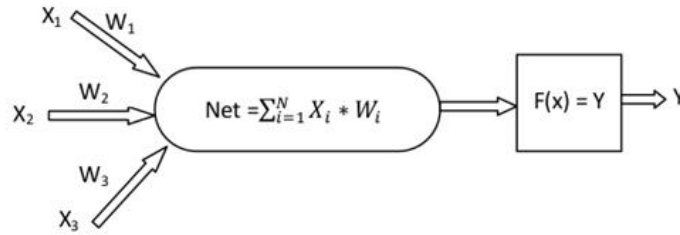
Yapay Sinir Ağları öğrenme algoritmalarına, öğrenme zamanına ve yapılarına göre aşağıdaki gibi sınıflandırılmaktadır.

- **Öğrenme Algoritmalarına Göre Yapay Sinir Ağları**
Danışmalı Öğrenme, Danışmansız Öğrenme, Destekli Öğrenme
- **Öğrenme Zamanına Göre Yapay Sinir Ağları**
Dinamik Öğrenme, Statik Öğrenme
- **Yapılarına Göre Yapay Sinir Ağları**
Geri Beslemeli Ağlar, İleri Beslemeli Ağlar

Beynin nöral yapısına benzetilen yapay sinir ağı nispeten kaba elektronik bir model olarak tanımlanabilir. Beyin temelde deneyimden öğrenir. Yapay sinir hücreleri verileri girdi olarak aldıkları ve işledikten sonra çıktı olarak verdikleri yapılar içermektedir. Bir yapay sinir hücresinin bölümleri:

- Girdiler (X)
- Çıktılar (Y)
- Ağırlık (W)
- Toplama (Σ)
- Birleştirme Fonksiyonu (Net)

Aşağıda Şekil 3.1’de basit bir yapay sinir hücresi modeli yapısı gösterilmiştir.



Şekil 3.1 Yapay Sinir Hücresi

$$NET = \sum_{i=1}^N X^i * W^i \quad (3.1)$$

Yapay sinir ağları giriş ile çıkış parametreleri arasındaki fark aşağıdaki hata fonksiyonu ile elde edilir[54,55].(d_j : hedeflenen sonuç, o_j : gerçekleşen sonuç)

$$E_p = \frac{1}{2} \sum_j (d_j^p - o_j^p)^2 \quad (3.2)$$

Hata fonksiyonunun farkı kullanılarak ağırlıklar yeniden düzenlenir.

$$\Delta^p w_{ji} = -\left(\frac{\partial E^p}{\partial w_{ji}}\right) \quad (3.3)$$

Burda “öğrenme oranı” n sabiti herhangi bir değer olacak şekilde seçilir. Yeniden ağırlık revizesi;

$$w_{ij}(t + 1) \cong w_{ij}(t) + n\delta_j i_i \quad (3.4)$$

Denklem 3.4.’deki gibi yazılır. Burada, $w_{ij}(t)$ ağırlık, i_i : i düğümünün sonucunda çıkan bir değer olabileceği gibi bir giriş, δ_j : j düğümünün hata değeri olabilir. Bir çıkış düğümü için hata değeri (δ_j);

$$\delta_j \cong o_j(1 - o_j)(d_j - o_j) \quad (3.5)$$

Şeklinde elde edilir. j düğümü bir gizli düğüm olmak üzere hata terimi (δ_j);

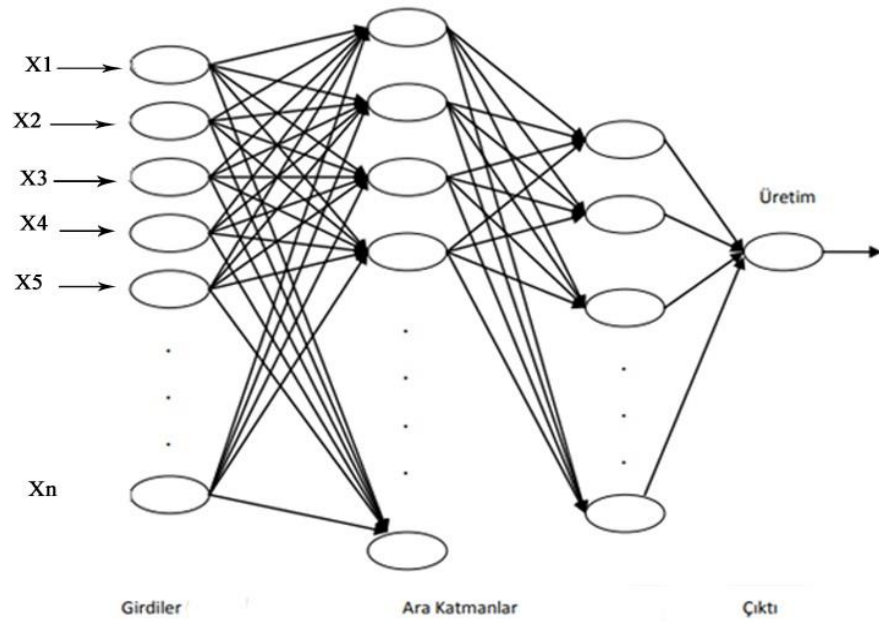
$$\delta_j = o_j(1 - o_j) \sum_k \delta_k w_{jk} \quad (3.6)$$

Şeklinde düzenlenir. Bir “moment” teriminin (α) eklenmesi ile ağırlık değişimleri üzerine etki yapabilir [37].

$$w_{ij}(t + 1) \cong w_{ij}(t) + n(d_j - o_j)i_i + \alpha(w_{ij}(t) - w_{ij}(t - 1)) \quad (3.7)$$

Yapay sinir ağı yapısı Şekil 3.2’de gösterilmiştir. Yapay sinir ağları 3 katmandan oluşur. Bunlar:

- Girdi katmanı(ları),
- Gizli(ara) katman(lar),
- Çıktı katmanı(ları).



Şekil 3.2 Yapay Sinir Ağı Yapısı

- **Girdi Katmanı(ları):** Eğitilecek verilerin dışardan giriş parametresi olarak verildiği katmandır. Genelde bu katmanda veride herhangi bir değişiklik yapılmaz ve kendinden bir sonraki katmanlara aktarılır.

- **Gizli(Ara) Katman:** Bu katmanda hesaplama karmaşıklığını ve zamanını olumsuz ya da olumlu etkileyebilecek durumlar söz konusudur. Ara katmanlar model yapısına bağlı olarak yapay sinir ağlarında birden fazla olabileceği gibi hiç olmaya da bilir. Birden fazla gizli(ara) katman içeren ağlarda katmanlar içerisinde nöron sayıları da birbirinden farklılık gösterebilir. Bu durum karmaşık oranını arttırdığı gibi daha zor görünen problemlerin çözümünde kullanılabilir.

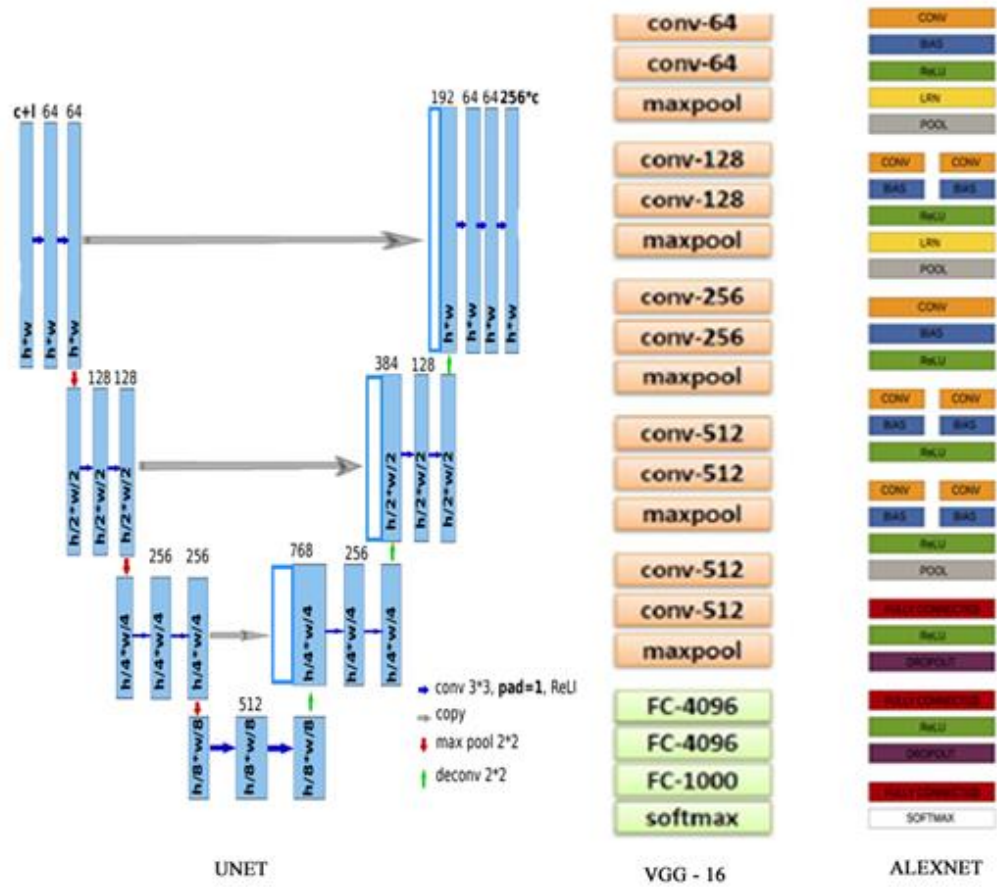
- **Çıkış Katmanı:** Bu katman ara katmanlardan aldığı bilgilerin çıktılarını verir.

3.3. Derin Öğrenme Mimarileri

Derin Öğrenme, insan beyninin çalışma şeklinden esinlenerek ortaya çıkmış makine öğreniminin alt dalı olan öğrenme şeklidir. En önemli özelliği ham veriden otomatik özellik çıkarabilmesidir. Derin öğrenme yöntemleri ailesi; sinir ağları, aşamalı olasılık modelleri ve denetlenmeyen ve denetlenen özellik öğrenme algoritmalarını kapsamaktadır ve giderek daha zengin hale gelmektedir [7].

.Derin öğrenme temelinde yapay sinir ağlarını baz olarak ortaya çıkmıştır. Ancak yapısal olarak farklılıkları da vardır. Yapay Sinir Ağlarının hesaplama yapısını temel alan derin öğrenme; klasik YSA yaklaşımlarından alınan sonuçlardan daha başarılı sonuçlar veren bir yöntemdir. Yakın zamana kadar, birçok makine öğrenimi ve sinyal işleme tekniği sığ yapılmış mimarilerden oluşmaktaydı. Bu mimariler yapısı gereği bir veya iki kat doğrusal olmayan özellik dönüşümü içermektedir. Bu yapıdaki sığ mimarilere örnek olarak Conditional Random Fields (CRFs), maksimum entropi (MaxEnt) modelleri, Gaussian Mixture Models (GMMs), doğrusal veya doğrusal olmayan dinamik sistemler, Kernel Regression, Multi Layer Perceptrons (MLPs), SVM'ler, Logistic Regression ve tek bir gizli katman içeren Extreme Learning Machines (ELMs) [38-41]. Sığ mimariler, basit veya iyi sınırlandırılmış problemleri çözmede başarı göstermiştir ancak sınırlı modelleme ve temsil gücü; insan konuşması, doğal ses ve dil işleme, görüntü ve görsel sahneleri içeren daha karmaşık gerçek dünya uygulamalarıyla uğraşırken zorluklara neden olmuştur [42].

Bununla birlikte, gerçek hayata uyarlanması zorlaşan insan bilgi işleme sistemi örneğin; ses veya görüntü gibi karmaşık yapıları çıkarmada sığ mimarilerin yetersiz kaldığı bunun için derin mimarilerin gerektiği görülmektedir. Örneğin, insanlarda görme sistemi katmanlı hiyerarşik yapılardan oluşmaktadır [43]. Bu yapı benzerliği modellemede ve çözümlemede derin mimarilerin daha kolay yapılacağını göstermiştir. Deep Neural Networks (DNN), birçok gizli katmana sahip ileri besleme ağları (Feed-Forward Neural Networks) veya MLP'ler, derin bir mimariye sahip modellerin ilk örneklerindendir. Şekil 3.3'de bu derin ağlara ait çeşitli katman ve model yapısı gösterilmiştir.



Şekil 3.3 Derin Ağlara ait Çeşitli Katman ve Model Yapısı

Yapay sinir ağlarında katman sayılarının farklılık göstermesi farklı türde derin öğrenme mimarilerinin ortaya çıkmasını sağlamıştır. Bu bölümde en sık kullanılan derin öğrenme mimarileri anlatılacaktır.

3.3.1. Deep Neural Network-DNN

Derin bir sinir ağı (DNN), girdi ve çıktı katmanları arasında çoklu katmanlar içeren bir yapay sinir ağıdır. Ekstra katmanlar, alt katmanlardan özelliklerin kompozisyonunu mümkün kılar, potansiyel olarak benzer karmaşık bir ağı kıyasla daha az ünite ile karmaşık verileri modellemeyi sağlar.

3.3.2. Deep Belief Network

Derin İnanç Ağları (DBN) Geofry HINTON tarafından tanıtılmıştır. Sınırlı Boltzmann Makinelerinin (Restricted Boltzmann Machines-RBM) yığını şeklinde tanımlanmıştır. Gizli değişkenlerin katmanlarından oluşan olasılık üzerine çalışan üretken bir modeldir. Her RBM katmanı hem kendinden önce gelen hem de kendinden sonra gelen katmanlara bağlıdır. Fakat hiçbir katmanın düğümleri birbirleriyle iletişim kurmazlar. Derin inanç ağları mimarileri genel olarak görüntü üretme ve tanıma alanlarında uygulanmaktadır. Salakhutdinov ve Hinton, dokümanlara DBN kullanarak büyük bir belge veri setinde kelime-sayı vektörleri elde etmişlerdir [44].

3.3.3. Deep Auto-Encoders

Denetimsiz öğrenme için kullanılan Diabola ağı olarak adlandırılan Oto-Kodlayıcılar (Auto Encoders-AE) özel bir yapay sinir ağı mimarisidir [45]. AE, bir veri kümesi üzerinde boyut indirgemeyi amaçlar. AE, kabaca girdi verisinin indirgenmiş yapısından en iyi özelliklerin öğrenilmesi hedefleyen bir ileri beslemeli sinir ağıdır [46].

3.3.4. Boltzmann Machines-BM

Boltzmann Makineleri, stokastik tekrarlayan sinir ağı modelidir. Bu yapı, iç temsilleri öğrenebilen ilk sinir ağlarından biridir.

3.3.5. Restricted Boltzmann Machines-RBM

Sınırlı Boltzmann Makineleri giriş veri setinde olasılık dağılımına göre öğrenebilen üretken ve rastgele bir yapay sinir ağıdır. RBM'ler, Boltzmann makinelerinin bir çeşididir. Gizli ve görünür olacak şekilde aralarında simetrik bir bağlantı olan iki düğümlü graflardan oluşur. Bir graf içindeki düğümlerde kendi aralarında bağlantı bulunmaz [44].

3.3.6. Recurrent Neural Networks-RNNs

Basit tekrarlayan bir sinir ağı (SRN), düğümler arasındaki bağlantıların bir dizi boyunca yönlendirilen bir döngü oluşturduğu yapay sinir ağı sınıfıdır. Bu döngü sayesinde, bir zaman dizisi için zamansal dinamik davranış sergilemesini sağlar. İleriye dönük sinir ağlarından farklı olarak, RNN'ler kendi iç durumlarını kullanarak girdilerin dizilerini işlerler. RNN, ses veya metin gibi verilerin modellenmesi için çok etkilidir ancak son zamanlarda kullanılmamıştır. Çünkü uzun ardışık yapıları yakalamak ve eğitmek zordur [7].

3.3.7. Recognizing Recurrent Neural Networks

Özyinelemeli tekrarlayan sinir ağları (rRNN); geleneksel bir RNN'nin sahip olmadığı, Dinamik uyarıların ve sağlamlığın ilk koşullara ve gürültüye hızlı çözümülenmesi örnek verilebilir. Ayrıca, dinamik girdiler için bir tahmin kodlaması uygular. RNN'lerin Bayes inversiyonunun hem bir beyin fonksiyonu modeli hem de bir makine öğrenme aracı olarak faydalı olabileceği önerilir. İnsan kinematiğinin çevrimiçi kod çözme (tanıma) için bir uygulama ile rRNN'nin kullanımını göstermektedir. Ayrıca dinamik girişler için tahmini kodlama şeması uygulanmasında fırsat verir [47].

3.3.8. Convolution Neural Network-CNN

Konvolüsyonel Sinir Ağları (Convolutional Neural Network-CNN ve ya ConvNet) çok katmanlı algılayıcıların (Multi Layer Perception-MLP) bir çeşididir. Hayvanların görüş mekanizmasından esinlenilmiştir. Canlıların görme merkezindeki hücreleri tüm görüntüyü içine alacak şekilde alt bölgelere ayrılmıştır. Bir nöronun kendi uyarı bölgesinden uyarılara verdiği cevap şeklinde ortaya çıkmıştır [48]. CNN algoritmaları ses ve görüntü işleme alanları başta olmak üzere biyomedikal, doğal dil işleme (Neuro Linguistic Programming-NLP) gibi birçok farklı alanda uygulanmaktadır. Özellikle de görüntü işleme çalışmalarında bugüne kadar birçok sonuç elde edilmiştir. Çişeran yaptığı çalışmada MNIST veri seti ile çalışarak , CNN ile hata oranı %2'lere kadar düşürmeyi başarmışlardır [49].

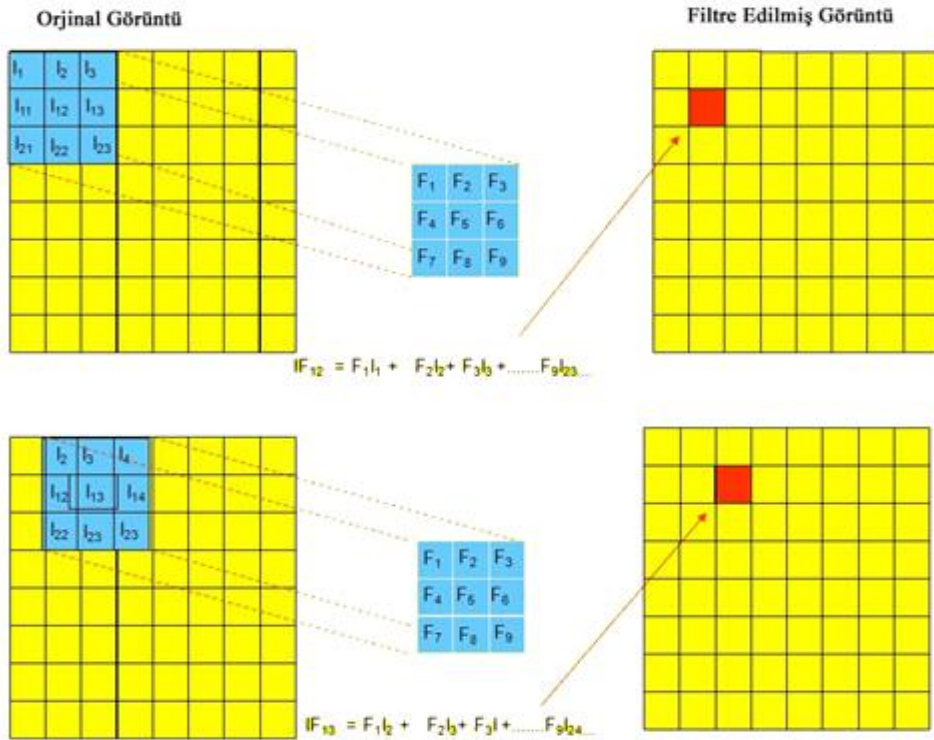
Konvolüsyon İşlemi: Amaç girdi verisinden özellik çıkarmaktır. İki fonksiyonun matematiksel işlemi Konvolüsyon işlemidir. Bu matematiksel işlem bir nöronun uyarı alanındaki uyarılara verdiği cevap şeklinde gösterilebilir. CNN yapısındaki konvolüsyon

işlemi, filtre olarak adlandırılan kernel (çekirdek) fonksiyonunu ana veri (görüntü) üzerinde her seferinde bir eleman kaydırılmasıdır. Her kaydırma işlemi, elemanların filtre matrisi ile birim çarpımının toplamı şeklindedir. Tüm görüntü işleme boyunca pencere kaydırılarak özellik haritası olarak adlandırılan konvolüsyon işleminin çıktısı üretilir. f ana veri(resim), h kernel olmak üzere konvolüsyon işlemi Şekil 3.4’de ve denklem 3.8’de gösterilmiştir.

$$f * h = \sum_k \sum_l f(k, l) h(i - k, j - l) \quad (3.8)$$

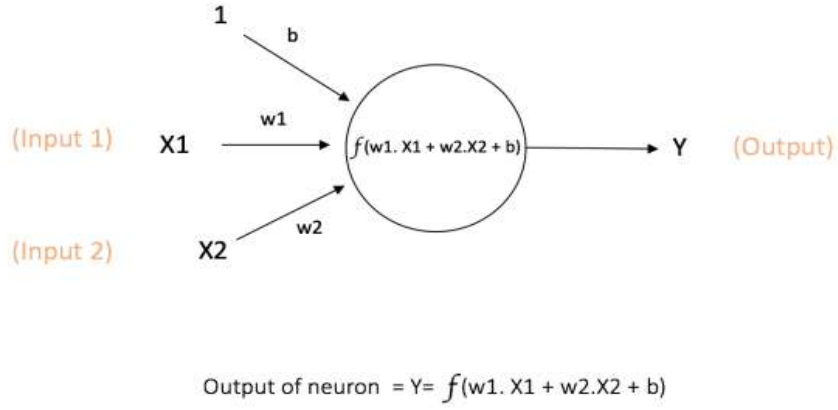
$$f = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \quad h = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

$$f * h = f_1 h_9 + f_2 h_8 + f_3 h_7 + f_4 h_6 + f_5 h_5 + f_6 h_4 + f_7 h_3 + f_8 h_2 + f_9 h_1$$



Şekil 3.4 Konvolüsyon İşlemi

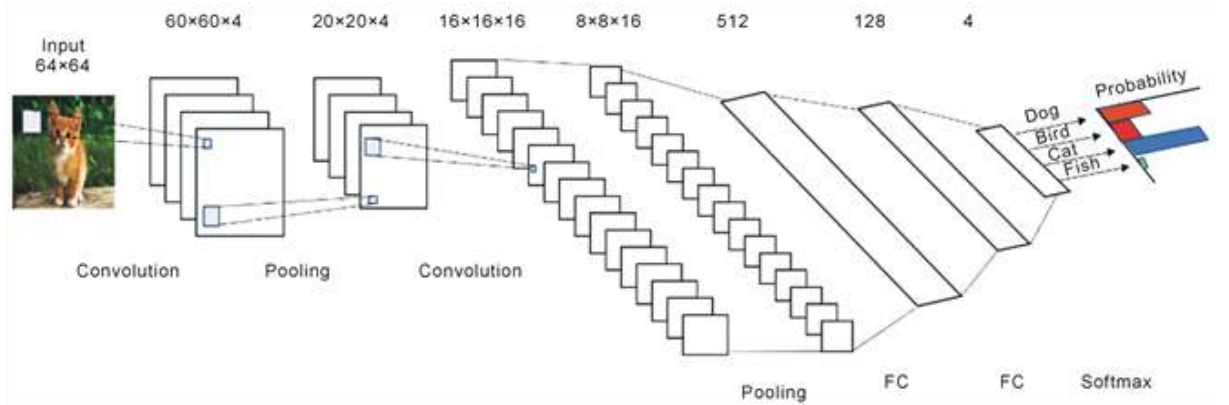
Yapay bir nöron girdiler, eğitilmiş bir girdi ve çıktıdan oluşur. Çıkış fonksiyonu herhangi bir aktivasyon fonksiyonu olabilir ama genel olarak sigmoid ya da Rectified Linear Unit (ReLU) tercih edilir. Şekil 3.5’te bu nöronun konvolüsyonel yapısı ve f fonksiyonu denklem 3.9’da gösterilmiştir.



Şekil 3.5 CNN’de Nöron Yapısı

$$f = \sum_i w_i x_i + b \quad (3.9)$$

f fonksiyonu açıldığında CNN’in, bir veya daha fazla konvolüsyonel katmandan, alt örnekleme (submapling) katmanı ve bunun arkasından çok katmanlı standart bir sinir ağı benzeri bir veya birden fazla tamamen bağlı katmandan oluştuğu öğrenilmektedir ve bu yapı Şekil 3.6’de gösterildiği gibidir.



Şekil 3.6 CNN Mimarisi

CNN mimarisinde konvolüsyon işlemi yapılırken üç tür tasarım yapısı vardır: kernel boyutu, kernel sayısı ve adım. Bu yapılar girdi olarak alınıp işlem gördükten sonra çıkış verisinin şekli ve hafıza kullanımı üzerinde etkilidirler.

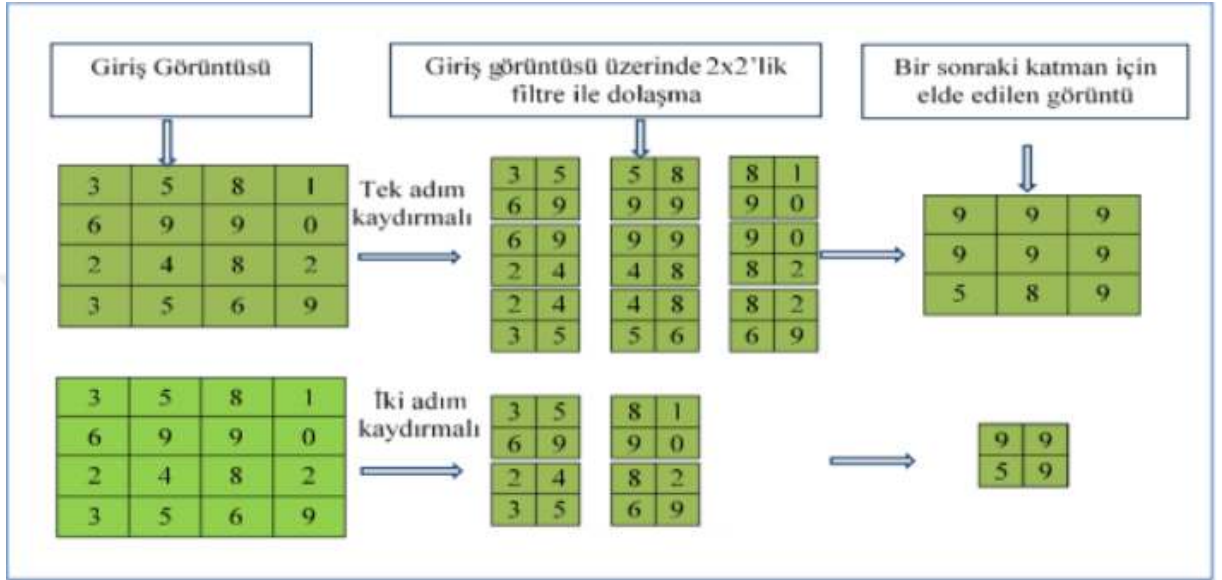
Kernel boyutu, konvolüsyon işlem katmanlarında her bir nöronun çıktı verisini belirler. Kernel değerleri modelin ağırlıklarını (b) denklem 3.9.'da temsil eder. Örneğin, kernel boyutu 3×3 ise, çıktısı giriş katmanından 9 farklı düğümle bağlantılıdır. Genellikle, kernel boyutu giriş veri boyutu ile aynıdır. Giriş tek kanallı gri tonlamalı bir görüntü ise, kernelde 1 kanallıdır. Giriş 3 kanallı RGB renkli görüntüyse, kernel 3 boyutlu bir yapı olarak seçilir. Çekirdek boyutu çok küçükse (örneğin 2×2) yeterli özellikleri ayıklayamayacaktır. Genel olarak 3×3 ve 5×5 kernel kullanılır.

Kernel sayısı çok önemli bir tasarım parametresidir. Çünkü odaklanılacak farklı özelliklerin sayısını belirler. Kernel sayısı küçükse, ağ verilerindeki bazı parametreleri kaçırabilir. Öte yandan kernel sayısı yinelenen (2 kat) ya da tekrarlayacak kadar çok olmamalıdır. Tekrarlanan filtrelere ek olarak çok sayıda filtre bellek sorunu getirir çünkü her bir ilişkili görüntü bilgisayarın belleğinde yer kaplar.

Her kernel giriş görüntüsü üzerinde kaydırıldığında (adım), bir özellik haritası oluşturur. Bu çıkış görüntüleri, tüm kernellerin çıktılarını oluşturduktan sonra birleştirilir. Giriş görüntüsü 2 boyutluysa çıktıları 3 boyutlu tensör olacaktır. Giriş görüntüsü 3 boyutlu veri ise çıktıları 4 boyutlu tensör olacaktır. Bu ekstra boyut birçok filtrenin kullanılmasından kaynaklanmaktadır. Tensörün çıktı derinliği ile kernel(filtre) derinliği aynı olmalıdır. Örneğin 30×30 ebadındaki bir giriş görüntüsü ve 3×3 boyutunda bir kernel düşünün. Konvolüsyon sonucu ortaya çıkan boyut 28×28 'dir. Yine de, giriş tensörüne uygulanan birden fazla kernel vardır. Sonuç olarak, çıkış tensörü boyutu $K \times 28 \times 28$ olur. Bir sonraki konvolüsyon işlemi için, kernel boyutu $K \times N \times M$ olur. CNN model gösteriminde, bir boyut genellikle $K @ N \times M$ şeklinde gösterilir.

Striding işlemi, kernel (filtrenin) giriş hacminin etrafında nasıl hareket ettiğini kontrol eder. Genellikle, kernel bir seferde giriş görüntüsü 1 adım (düğüm) boyutu boyunca kayar. Yine de, çıkış tensörü boyutunu kontrol etmek için, adım numarasını (kaymasını) değiştirebilir. Adım miktarı 2 alınırsa, kernel 2 düğüm kaydırarak bir sonraki pencereye gider. Örneğin, 64×64 boyutunda bir görüntü ve 3×3 boyutunda bir kernel düşünün; eğer adım 1 ise, çıktı görüntüsü 62 olacaktır. Adım 2 ise, çıktı görüntüsü 31×31 olacaktır. Konvolüsyon işleminde, çıktı tensörlerinin büyüklüğünü giriş hacmi ile aynı tutmak için giriş tensörüne padding (dolgu) eklenebilir. Padding büyüklüğü genellikle kernelin boyutuna göre belirlenir. Örneğin, kernel boyutu 3×3 ise görüntü etrafına 1 piksel yeterlidir. Kernel boyutu 5×5 ise, görüntü etrafında 2 piksel kullanılabilir. Padding işlemi

işleminde pooling boyutu ve stride (adım) olmak üzere iki önemli parametre vardır. Bu katmanda belli başlı filtreler vardır. Bu filtreler görüntü üzerinde adım atma parametresine göre görüntüyü oluşturan piksellerin maksimum değerlerini maksimum havuzlama (Max pooling) ve ya değerlerin ortalamasını ortalama havuzlama (Average pooling) ile alarak işlem yapar. Şekil 3.8’de pooling örneği gösterildiği gibidir.



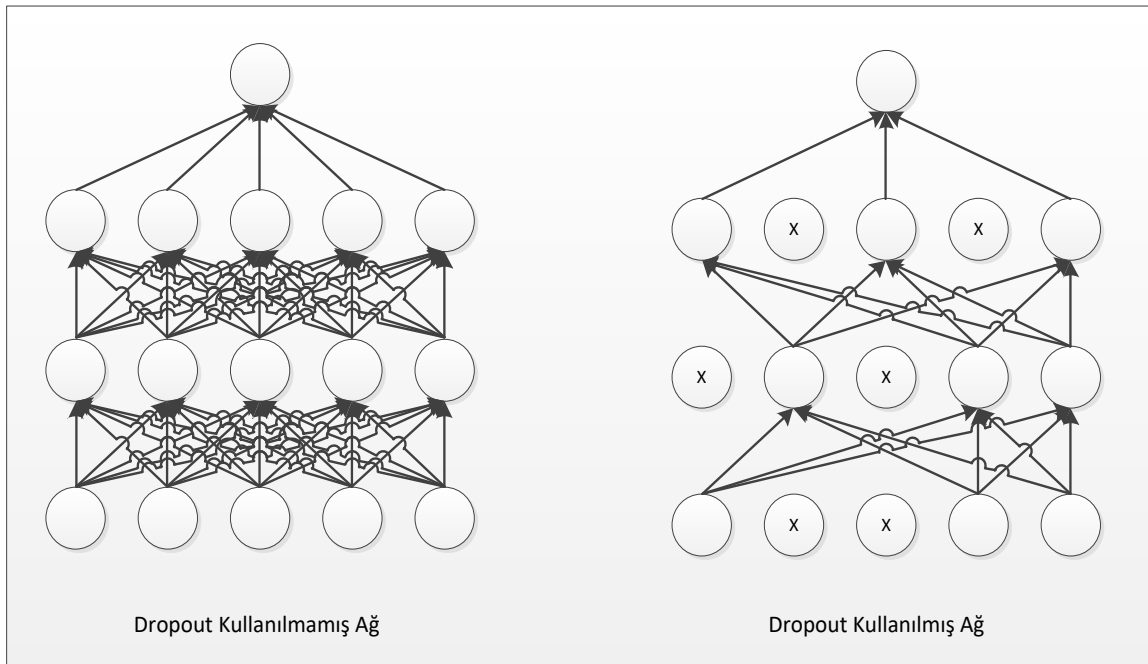
Şekil 3.8 Pooling Örneği

Tam Bağlı Katman (Fully Connected Layer): Tam bağlı katman havuzlama ve Relu katmanlarının ardından gelmektedir. Tam Bağlı katman kendinden önce gelen katmanın tüm özelliklerine bağlıdır. Konvolüsyon ve havuz katmanları dikdörtgen şekilli çıktılar üretir. Bu ağırlık matrisi çıktılar, vektör matrisine dönüştürülür. Örneğin son katmanın ürettiği matris boyutu $25 \times 25 \times 224 = 140000 \times 1$ ve tam bağlı katmandaki matris boyutu 4096×1 şeklinde seçildiğini varsayalım. Totalde 140000×4096 ağırlık matrisi oluşur. Her bir 140000 sinir ağı 4096 sinir ağı ile bağlı demektir. Tam bağlı katmandan önceki katman yüksek seviyeli özellikleri temsil eder. Bu katman yardımıyla, bu yüksek seviyeli özellikler gizli katmanların ağırlıkları ile çarpılabilir.

Hiper Parametre (Hyper Parameter) Ayarlamaları ve Seçimleri: Verinin eğitilmesi ile öğrenen makine öğrenmesi modelleri oluşturulurken, bazı parametreler vardır ki tasarımcının kararına bağlıdır. Model oluşturulurken kullanılacak olan tasarım, teknik ve algoritmalara tasarımcı karar verir. Örneğin, K-NN sınıflandırması için k değerine modeli tasarlayacak kişi karar verir. Benzer şekilde derin öğrenmede modeli tasarlarlarken kullanılacak kernel fonksiyonuna, nöron sayısına, modeldeki katman sayısına veya

seyreltme (dropout) değerine ihtiyaç olup olmayacağına modeli tasarlayan kişi karar vermelidir ve hangisinin daha iyi sonuç vereceği kesin değildir. Veri setine göre değişiklik gösteren bu parametrelere hiper parametre denir. Hiper parametrelerin seçimi genellikle tasarımcının sezgisine ve tecrübesine göre değişmektedir. Hiper parametreler model hiper parametreler ve iyileştirici (optimizer) hiper parametreler olarak ikiye ayrılmaktadır. modelin yapısında yer alan değişkenlere model hiper parametre denir. Aktivasyon fonksiyonları, dropout, katman sayısı ve modele özgü hiper parametrelerdir. Modelin yapısından ziyade eğitim ve optimizasyon yapılarında olan değişkenlere ise İyileştirici (optimizer) hiper parametre denir. Öğrenme oranı (learning rate), mini-batch boyutu ve epoch (devir sayısı) içermektedir.

- **Seyreltme Katmanı (Dropout Layer):** Katmanlarda eşik değerinin altında kalan düğümler seyreltilerek ağıın başarımlarının arttığı gözlemlenmiştir. Zayıf bilgilerin çıkarılması öğrenimi artırır. Ağıın modifiye edilmesi maliyet fonksiyonunu (cost function) değiştirmez. Büyük veriler içeren ağlarda eğitim yapıldığında bazen bu verilerden dolayı ağ ezberleme yapar. Ağıın bu ezberlemeyi engellemek için önceden belirlenmiş ağdaki bazı nöronları kaldırma işlemidir [50]. Genelde %50 (0,5) ile %20 (0,2) arasında nöron kaldırılır ve kaldırılan nöronlar ağa etki etmez. Aşağıda Şekil 3.9’da Dropout kullanılmış ve kullanılmamış ağ gösterilmiştir.

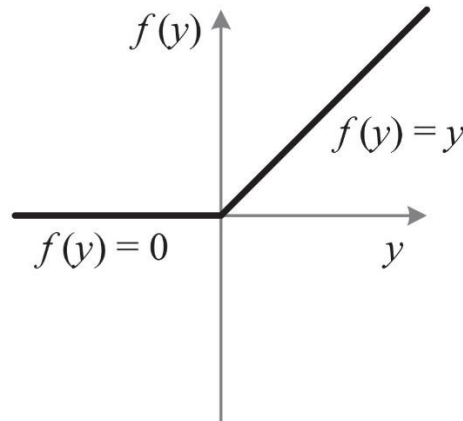


Şekil 3.9 Dropout Kullanılmamış ve Kullanılmış Ağ Yapıları [51]

Aktivasyon Fonksiyonları (Activation Function): Aktivasyon fonksiyonları çok katmanlı ağlarda modele non-linearite katar. Gizli katmanlarda $y = f(x, w)$ şeklinde matris çarpımı yapılan lineer fonksiyon da tüm nöronların ağırlıkları hesaplandıktan sonra çıktıyı lineer olmayan değere dönüştürür. Gizli katmanlarda geri türev alındığından gizli katmanların çıkış değeri bazı aktivasyon fonksiyonları kullanılarak normalize edilir. Aktivasyon fonksiyonları gradient descent ile geri dönüp düzeltme yapmak için de kullanılabilir. Bu tarz kullanımdaki amaç kolay türev alabilmektir. Yaygın aktivasyon fonksiyonları: Sigmoid, Softmax, ReLU, TanH, SoftPlus, ELU, PReLU, Swish'dir. En yaygın kullanılan ReLU, Softmax ve Sigmoid'dur.

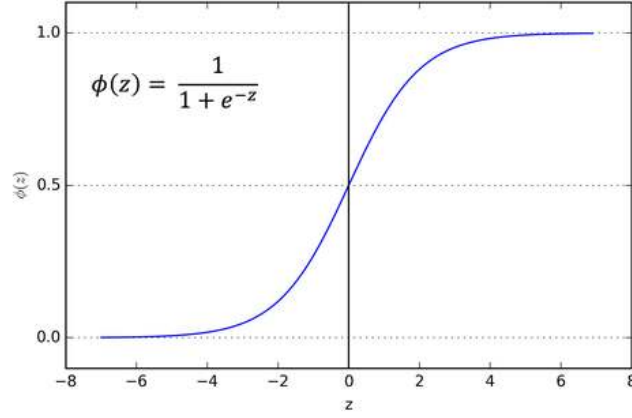
- **Derin Öğrenme Mimarilerinde Kullanılan Aktivasyon Fonksiyonları:** Bu fonksiyonlar çok katmanlı ağlarda modele doğrusal olmama (non-linearite) katar. Gizli katmanlarda $y = f(x, w)$ şeklinde matris çarpımı yapılan lineer fonksiyon da tüm nöronların ağırlıkları hesaplandıktan sonra çıktıyı lineer olmayan değere dönüştürür. Gizli katmanlarda geri türev alındığından gizli katmanların çıkış değeri bazı aktivasyon fonksiyonları kullanılarak normalize edilir. Aktivasyon fonksiyonları gradient descent ile geri dönüp düzeltme yapmak için de kullanılabilir. Bu tarz kullanımdaki amaç kolay türev alabilmektir. Yaygın aktivasyon fonksiyonları: Sigmoid, Softmax, ReLU, TanH, SoftPlus, ELU, PReLU, Swish'dir. En yaygın kullanılan ReLU, Softmax ve Sigmoid'dur.

- **ReLU:** Yapay sinir ağları bağlamında doğrultucu (Rectified Linear Unit-ReLU) doğrusal olmayan bir aktivasyon işlevidir. Negatif giriş parametreleri için 0 değerini alırken, x pozitif giriş parametreleri için x değerini almaktadır. ReLU fonksiyon grafiği Şekil 3.10'da gösterilmiştir.



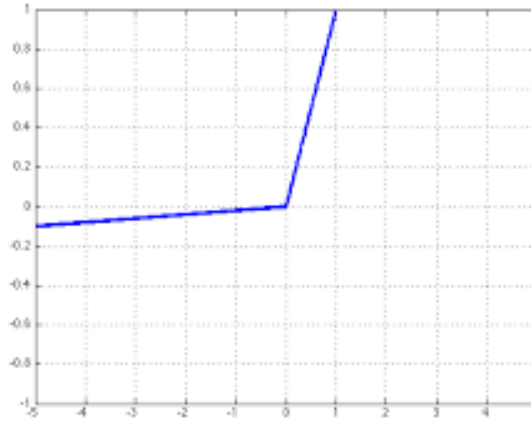
Şekil 3.10 ReLU Fonksiyon Grafiği

- **Sigmoid:** Sigmoid aktivasyon fonksiyonu sürekli ve türev alınabilir bir fonksiyondur. Bu fonksiyon giriş parametre değeri 0 ile 1 arasındadır. İkili sınıflandırma için kullanılır ve genelde son katmanda bulunur. Sigmoid fonksiyon grafiği Şekil 3.11’de gösterilmiştir.



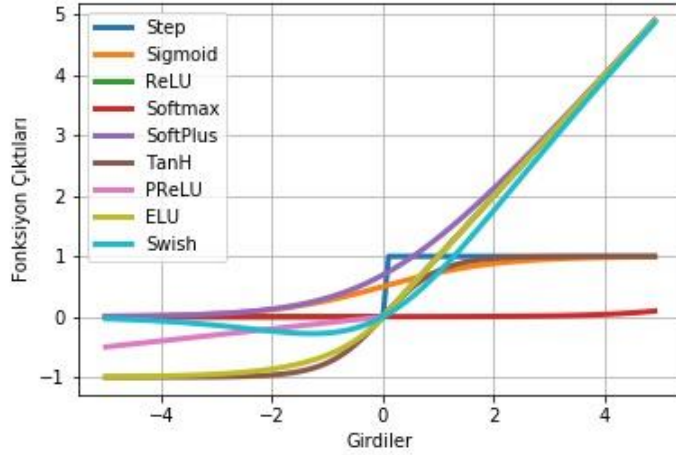
Şekil 3.11 Sigmoid Fonksiyon Grafiği

- **Softmax:** Çoklu sınıflandırma problemlerinde kullanılan bir aktivasyon fonksiyonudur. Verilen her bir giriş parametresinin bir sınıfa ait olma olasılığını gösteren $[0,1]$ arasında değerler üretmektedir.



Şekil 3.12 Softmax Fonksiyon Grafiği

Yaygın olarak kullanılan aktivasyon fonksiyonlarına ait grafik Şekil 3.13’de gösterilmiştir.



Şekil 3.13 Yaygın Kullanılan Aktivasyon Fonksiyonları[47]

- Mini-Batch (Batch Size) Boyutu:** Derin öğrenme uygulamalarında aynı anda tüm veriyi nöronlardan a-ışleterek ağı eğitmek zamansal ve donanımsal kapasite bakımından oldukça maliyeti yüksek bir iştir. Sebep olarak öğrenmenin her iterasyon adımımda geri yayılım (back propagation) işlemi yapılarak ağıda geriye dönük şekilde gradyan (gradient descent) hesaplama işlemi yapılır ve ağırlık (w) değeri bu şekilde yeniden güncellenir. Hesaplama işlemi yapılırken üzerinde çalışılan veri setindeki sayı ne kadar fazlaysa hesaplama değeri de o derece artmaktadır. Ağı eğitim performansını artırmak, bellek ve zaman açısından maliyeti düşürmek için tüm veriyi almak yerine veriyi belirli gruplar halinde almak ve küçük gruplar halinde işlem yapmak daha uygundur. Bu yöntem mini-batch denmektedir. Yani tasarlanan modelin eğitim ve test verilerinin modelden geçerken aynı anda kaç veriyi işleyeceği anlamına gelmektedir. Ağı alınan küçük gruptaki veriler rastgele seçilmektedir. Oluşturulan model mini-batch'lerin her birinde ileri besleme (feedforward) işlemi yaparak hata değeri hesaplanır. Bu işlem sonrasında mini-batch'lerin her birinde geri yayılım işlemi ile gradyan işlem hesaplaması gerçekleştirilerek ağırlıklar güncellenir. Bu ağırlık değerlerini hesaplamak için gradient descent kullanır. Genel olarak mini-batch boyutları hepsi aynı olup %1-%5 arasında ideal olarak seçilmektedir. %1'den küçük seçilirse gürültü çok fazla olmakta, %5'ten fazla seçilirse modelin eğitim değerleri düşmektedir.
- Learning Rate (Öğrenme Hızı):** Derin öğrenme de parametreler güncellenirken geri yayılım (backpropagation) işlemi ile yapılır. Geri yayılım işleminde parametre güncelleme işlemi geriye doğru türev (chain rule) alınıp farkın çıkarılması ve bu

fark parametre değerinin öğrenme hızı (learning rate) parametresi ile çarpılması, sonucunda çıkan değerin ağırlık değerinden çıkarılması ve yeni ağırlık değerinin yeniden hesaplanmasıyla olmaktadır. Öğrenme hızı parametrik değeri sabit bir katsayı olarak belirlenebilir veya belirli düzeyde artış veya azalış gösteren şekilde ayarlanabilir olup 0,1 ile 0,000001 arasında seçilmektedir. Öğrenim hızı seçilirken çok yüksek seçilirse modelin eğitim verisinde çok etkilenmesine neden olmaktadır. Benzer durumda küçük değerin olması modelin yavaş ilerlemesine ve öğrenimin çok uzun sürmesine neden olmaktadır. Bu yüzden genel olarak hedef ilk önce büyük değeri kullanılıp eğitimi artırmak ve istenilen hata değerine yaklaştıkça öğrenme hızını düşürmektir. Momentum değerine bağlı olarak belirlenebilir.

- **Epoch (Eğitim Tur Sayısı):** Mini-batch tüm veriyi tek seferde eğitmeyip belli parçalar halinde eğitime sunar. Her parçanın ağırlığı eğitimin sonunda geri yayılıma göre hesaplanıp güncellenir. Epoch'da ise her iterasyon adımı tüm veri setinin geriye ve ileriye doğru (backward and forward) yöntemleri ile model tekrar eğitilip ağırlık güncellemeleri yapılır. Epoch sayısı az seçilirse model eksik öğrenir ya da çok seçilirse model aşırı öğrenir. Bu epoch sayısı tahmini olarak belirlenmekteydi fakat tensorflow desteğiyle earlystopping özelliği ile başarımla istenilen seviyeye gelince otomatik olarak durdurulmaktadır.
- **Momentum (Momentum Katsayısı):** Öğrenme sırasında gürültü üreten yöntemler çeşitli yöntemlerle normalize edilip bu gürültüye sebep olan olgular azaltılmaktadır. Gürültü azaltan normalizasyon yöntemi momentum katsayısı ile yapılmaktadır. Stochastic gradient descent momentum ile kullanıldığında hızı artırmaktadır. Bu yöntemde üretilen yeni değeri olduğu gibi alınmaz, onun yerine üretilen bir önceki değeri momentum katsayısı oranında alınarak çıkacak yeni değeri hesaplanır. Bu sayede grafikteki salınımlar ve gürültü değerinin azalması sağlanır. Bu da diğerlerine göre daha hızlı bir yöntem oluşturmuş olur. Genelde kullanılan momentum veya momentum beta katsayısı 0,9 kabul edilmiş olup parametre aralığı 0,8 ile 0,99'dur.
- **Optimizasyon Yönteminin Seçimi:** Optimizasyon; bir uygulama sürecindeki işgücü, zaman, hammadde, donanım vb. gibi temel parametrelerin en verimli ve etkili şekilde kullanılmasını amaçlayan sonucunda katma değeri olan bir çıktı elde etmeyi sağlayan bir teknolojidir. Derin öğrenme uygulamalarında temelde bir optimizasyon problemi bulunmaktadır. Derin öğrenme alanında yapılan

alıřmalarda farklı optimizasyon yöntemleri kullanılmaktadır. Bunlara stochastic gradient descent, adamax, adadelta, adagrad, adam örnek olarak verilebilir. Optimizasyon fonksiyonu derin öğrenmede, doğrusal olmayan problemlerin çözümünde optimum değeri bulmak için kullanılır. Bu optimizasyon fonksiyonları arasında başarımları ve hız farklılıkları vardır. Yani birinin diğerine göre artışı veya eksisi vardır denilebilir.

İki türü bulunan Gradient Descent'in; Numeric Gradient Descent yaklaşık sonuç vermekte yavaş çalışmakta olup uygulaması kolaydır. Analytic Gradient Descent ise hızlı çalışmakta ve tam sonuç vermekte olup hataya eğilimlidir. Bu yüzden Analytic versiyonu ile hesaplama, Numeric'te ise kontrol edilmektedir.

Derin öğrenmede standart tanımlı olarak kullanılan yöntem ise Stochastic gradient descent olup diğer yöntemlere göre daha yavaş çalışmakta ve resim tanıma problemlerinde yeterli verimi sağlamamaktadır.

AdaGrad seyrek parametreler için büyük güncellemeler yaparken sık parametreler için daha küçük güncellemeler yapar. Bu nedenle resim tanıma gibi seyrek verilerde tercih edilir. AdaGrad öğrenme hızı (learning rate) algoritmaya bağlı olarak zamanla düşmektedir bu yüzden sistem bir süre sonra öğrenmeyi bırakır. AdaGrad'ın bu öğrenme hızı sıkıntısını RMSprop ve AdaDelta hız düşüşünü çözmüşlerdir. Adam ise AdaDelta'ya benzer bir algoritmadır. AdaDeltadan farkı hiper parametre değişiklikleri olarak learning rate yanı sıra momentum değişikliklerini de önbellekte saklar bir bakıma RMSprop ile momentumu birleştirir.

Lojistik Regresyon: Lojistik regresyon, sınıflandırma problemleri için kullanılan doğrusal ve olasılıksal çalışan bir sınıflandırıcıdır. Giriş parametrelerini tutan bir dizinin veya vektörün, rastgele bir sınıfa ait olma ihtimali Denklem 3.11'de gösterilmiştir. x girdisinin i . sınıfına ait olma olasılığı $\sigma(z)$ 'dir.

y =Sınıflandırıcı Sonucu, i =Sınıf Numarası, x =Girdi Değeri,

w =Ağırlıkları, b =bias, z =Regresyon Değeri

$$P(y=i | x, w, b) = \sigma(z)$$

$$z = w \cdot x + b \quad (3.11)$$

Lojistik regresyon modelde sınıflandırma gerçekleştirmek için öncelikle hiper parametreleri ve giriş verilerini tanımlar. Önceden tanımlanan modelde hata fonksiyonunu kullanır. Amacı hata fonksiyonun en aza indirmek için modeli eğitmektir. Burada kullanılan hata fonksiyonu sınıflanma türüne göre değişmektedir. Denklem 3.12 ve 3.13’de bu fonksiyonlar tanımlanmıştır.

İkili sınıflandırma hata fonksiyonu(Sigmoid fonksiyonuna bağlı olarak);

$$E(w, b) = -\frac{1}{m} \sum_i^m (y_i \cdot \log(\hat{y}) + (1 - y_i) \cdot \log(1 - \hat{y})) \quad (3.12)$$

Çoklu sınıflandırma hata fonksiyonu (Softmax fonksiyonuna bağlı olarak);

$$E(w, b) = -\frac{1}{m} \sum_i^m \sum_j^n (y_{ij} \cdot \log(\hat{y}_{ij})) , \hat{y} = \sigma(z) \quad (3.13)$$

Ağırlık Algoritması (Weight Algorithm): Ağırlık fonksiyonu (Weight Algorithm-w), hata fonksiyonu ile beraber hesaplanır. Ağırlık fonksiyonunun hesaplama adımları aşağıdaki gibidir.

1. Rastgele ağırlık (w) değeri seç ve eğitimi başlat.

2. Her giriş değeri ($x_1, x_2, \dots, x_n$) için gerçek etiket değeri y ve tahmini etiket çıktısı $\hat{y}$ için öğrenme katsayısı(learning rate) α bağlı olarak b bias, ağırlık w parametrelerini güncelle.

- Pozitif durum $y - \hat{y} = 1$ ise;
 - For $i = 1 \dots n$
 - If $\hat{y} = 0$ and $y = 1$:
 - $w_i = w_i + \alpha \cdot x_i$
 - $b_i = b_i + \alpha$
- Negatif durum $y - \hat{y} = -1$ ise;
 - For $i = 1 \dots n$
 - If $\hat{y} = 1$ and $y = 0$:
 - $w_i = w_i - \alpha \cdot x_i$
 - $b_i = b_i - \alpha$

- Sıfır durum $y - \hat{y} = 0$ ise;
- Doğru etiket bir değişiklik yapma

3. Yukarıdaki adımları t zaman süresince tekrarla.

Gradyan İnişi (Gradient Descent): Gradyan inişi (Gradient Descent), en uygun ağırlık (w) ve bias (b) değerlerini bulmaya çalışan bir optimizasyon algoritmasıdır. Gradyan inişi, makine öğrenmesi çalışmalarında çokça kullanılan bir yöntemdir. Hata oranını (E) en aza indirerek en düşük hata oranını bulmayı amaçlar. Hata oranını düşürmek için bu fonksiyonun türevini almaktadır. Elde edilen türev değeri negatif ise Gradyan yönünde güncelleme yapmaktadır. Gradyan inişi ile hata indirgeme adımları aşağıdaki gibidir.

1. Rasgele ağırlık (w) ve bias (b) ata ve eğitime başla.
2. Her bir giriş değeri ($x_1, x_2, \dots, x_n$) için öğrenme katsayısı (learning rate) α bağlı olarak b bias, ağırlık w parametrelerini güncelle (hata oranı (E_i)).

for $i=1 \dots n$

if $E_i \leq E_{i-1}$

$$w'_i = w_i - \alpha \frac{\partial E}{\partial w_i} = w_i - \alpha(y - \hat{y})x_i$$

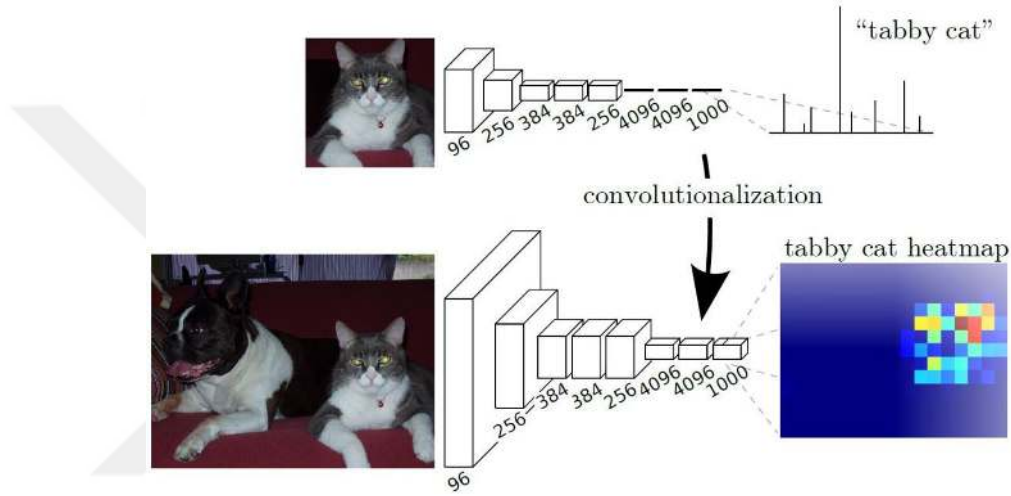
$$b' = b - \alpha \frac{\partial E}{\partial b} = b - \alpha(y - \hat{y})$$

3. Hatanın değeri minimuma inene kadar ($E_i \cong 0$) tekrar et.

3.3.9. Fully Convolution Neural Network-(FCNN)

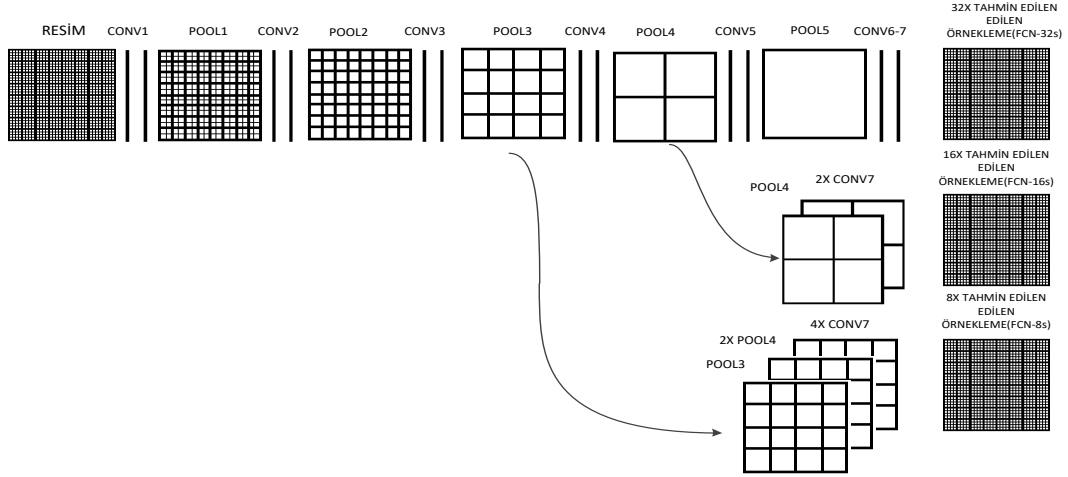
Jonathan Longi, Evan Shelhamer ve Trevor Darrell 2014 yılında “Fully Convolutional Networks for Semantic Segmentation” adlı çalışmayla tam konvolüsyonen sinir ağı fikri ortaya çıkmıştır[52]. Bu çalışmalarında piksel şekilli haritalamayı önerdiler. Tam konvolüsyonel Sinir ağı bir normal bir CNN’dir. Yapı olarak benzer olsalar da katman ve bazı açılardan farklılık göstermektedir. CNN genel yapıya bakıp sınıflandırma yapar. FCNN ise piksel haritalama ile çalıştığı için her bir pikselin neye ait olduğunu bulup sınıflandırma yapmaktadır. Bir FCNN bu çalışma yapısı sayesinde bütün görüntüleri işler ve olasılık haritaları oluşturur. Eğitim için uçtan uca uygulama yaptığı için yoğun olarak etiketlenmiş görüntüler gereklidir. Bir FCNN işlemi giriş olarak hangi boyutta görüntü

verisi alırsa aynı boyut pikselde çıktılar üretir. CNN'deki gibi geriyayılım kullanır ve hata oranı piksel şekillidir. CNN benzerliği nedeniyle mevcut sınıflandırılmış ağırları FCNN yapısına dönüştürülüp kullanılabilmesidir. Bundan dolayı FCNN önceden eğitilmiş CNN modellerinden (AlexNET, LeNET,VGG vb.) yararlanabilir. Bir CNN modelini FCNN'de kullanabilmek için CNN katmanlarındaki Tam Bağlı Katmanları (Full Connected Layer) 1x1 kernellere dönüştürmemiz gerekmektedir. Şekil 3.14'de bu dönüşüm işlemi gösterilmiştir. Bu sayede giriş boyutuyla aynı boyutta çıktı ısı haritaları oluşturulmaktadır.



Şekil 3.14 Tam Bağlı Katmanları Dönüştürerek ısı Haritaların Oluşturulması[52]

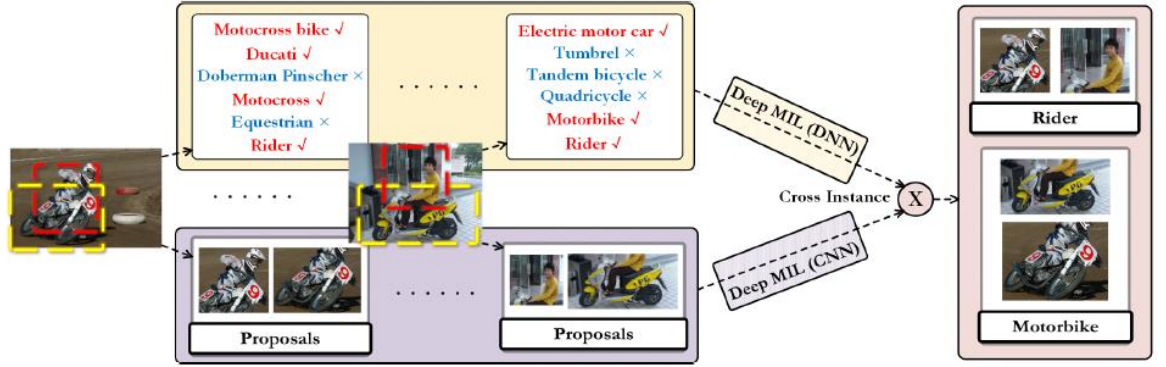
Bir görüntü sınıflandırmada tam bağlı katmanlar mekânsal verilere bakmadan yalnızca çerçeveyi göz önüne alıp bütün görüntüler için özellik vektörleri oluşturur. Tam konvolüsyon katmanları ise her piksel için özellik haritası oluşturur ve bu özellik haritasına bağlı olarak piksele dayalı sınıflandırma yapar. Orijinal görüntü boyutlarını geri yüklemek için bu özellik haritası tarafından dekonvolüsyon yapılır. Ağırlık baz alınarak yapılan dekonvolüsyon işlemi bir çift yönlü enterpolasyon görevi görür. Havuzlama (pooling) işlemi de sınıflandırma doğruluğunu artırırken mekansal bilgileri kısmen de olsa ihmal eder. Bu segmantasyonun mekansal doğruluğunu sınırlandırdığı için bir deavantajdır. Bu yüzden FCNN bu havuzlama işlemini tersine çevirmek için FCNN-32, FCNN-16 ve FCNN-8 fikrini öne sürmüştür. Kayma (Stride) yaparak çıktı haritasını giriş boyutuna çevirip, kayma miktarları 32, 16 ve 8 alınmıştır ve bu işlemler özellik çıkarımı için hepsi son katmanda değil ara katmanlarda yapılmıştır. Bu yapı Şekil 3.15'de gösterilmiştir [52].



Şekil 3.15 FCN Model Katmanları

3.4. Derin Öğrenme ile Sınıflandırma

Sınıflandırma çalışmalarında son yıllarda derin öğrenme teknolojisi ile sınıflandırma oldukça popüler hale gelmiştir. Görüntü analizi ve işleme alanında daha iyi performansla sonuç veren yöntemler aranmakta ve denenmektedir. Klasik makine öğrenmesi yöntemleri bu anlamda yetersiz kalmaktadır. CNN’de benzer makine öğrenmesi tabanlı olmasına rağmen klasik yöntemlere göre çok daha iyi sonuçlar verdiği gözlemlenmiştir. Son zamanlarda yapay zeka yöntemleri ile özellikle gerçek zamanlı görüntü işleme konusunda çalışmalar artmış, bu alanda otomatik sistemlerin optimize edilmesi bu sistemleri daha iyi çalışır hale getirmeyi sağlamıştır. Gerçek zamanlı görüntü işlemenin çalışma alanları sınırlı kalmayıp, coğrafi bilgi sistemleri alanından sağlık alanına, güvenlikten ulaşım sektörüne kadar birçok alanda kullanılmaya başlanmıştır. Derin öğrenme hem ikili sınıflandırma hem de çoklu sınıflandırma yapan diğer yöntemlere göre daha iyi sonuçlar vermiş olup SVM, K-NN gibi yöntemlerle yarışacak seviyede başarılar göstermiştir [53].



Şekil 3.16 CNN İkili Sınıflandırma Çalışma Yapısı [53]

CNN ikili sınıflandırma çalışma yapısında öncelikle resimlere ait öznitelikler çıkarılır. Çıkarılan öznitelikler ağda değerlendirilip olanı eğitir. Eğitilen ağ yeni gelen test görüntülerini ağdan geçirip görüntüde sürücü ve ya motosiklet var mı varsa ona göre sınıflandırma yapar. Bu eğitim sayesinde sonraki gelen eğitim görüntülerinde sürünün konumunu ve ya aracın markasını söyleyebilecektir. Genel bir yargı yapılacak olursa Şekil 3.16’de gösterildiği gibi ileri ağ yapılarında çoklu etiketleme yaparak resimdeki her bir ögenin etiketini söyleyebilir hale gelmektedir.



Şekil 3.17 CNN Çoklu Sınıflandırma Örneği [53]

Resim içerisindeki yer alan kelimeler, görüntünün anahtar kelimesi olarak kabul edilir. Şekil 3.17’de bazı örnekleri gösterilmiştir. Web sayfalarından çıkarılan anahtar kelimeler oldukça gürültülüdür. Ancak, anahtar kelimelerin çokluğu görüntü ile ilgisiz olsa da, bazı anahtar kelimeler aslında kategori etiketinden daha ayrıntılı ve daha bilgilendirici açıklamalar sağlar [53]. Örneğin uçağın modeli bize, insan etiketlerinden daha özel açıklamaları çıkarmak için fırsat sunmaktadır. Bu sayede geniş ölçekli anahtar kelimesi kütüphaneleri yapıp nesnelerin ayırt edilmesi daha kolay hale getirilir ve anahtar kelimenin tanımladığı nesneyi daha detaylı tanımlamasına olanak sağlanır.

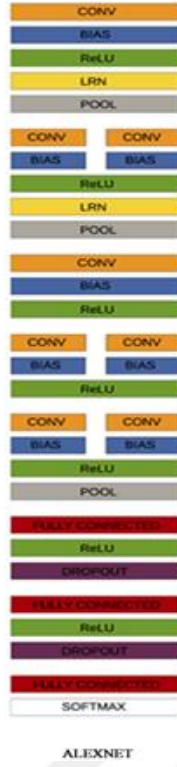
4. DERİN ÖĞRENME AĞ YAPILARI

Derin öğrenme veri seti eğitimi için sınıflandırma aşamasında en çok kullanılan derin öğrenme modellerinden AlexNET ve VGG16 model ağ yapıları anlatılmış olup, görüntü segmentasyonu için U-NET model ağ yapısına değinilmiştir.

4.1. AlexNET Ağ Yapısı

2012 yılında ILSRVRC yarışında grafik işlemci birimi (Graphics Processing Unit-GPU) destekli Compute Unified Device Architecture (CUDA) yapılmış konvolüsyonel sinir ağı modelidir. AlexNet, önceki ağ yapılarından daha iyi performans göstermiştir ve diğerlerindeki hata oranı % 26'dayken AlexNet'de bu oran % 15.3'e indirgenebilmiştir. AlexNet, Alex Krizhevsky, Geoffrey Hinton ve Ilya Sutskever'den oluşan SuperVision grubu tarafından tasarlanmıştır [54]. Bu yapıdaki ayarlamalar aşağıdaki gibi olmalıdır.

- Batch-size 128 alınmalıdır.
- Son katmana kadar “ReLU” aktivasyon fonksiyonu kullanılmalıdır. Son katmanda ise “Softmax” kullanılmaktadır.
- Fully Connected katmanlarından önce isteğe bağlı olarak Dropout (0,5) kullanılabilir.
- İsteğe bağlı olarak Maxpooling katmanlarından sonra Normalization katmanı kullanılabilir.
- SGD momentum 0,9 ve Learning rate 0,01 alınmalıdır.



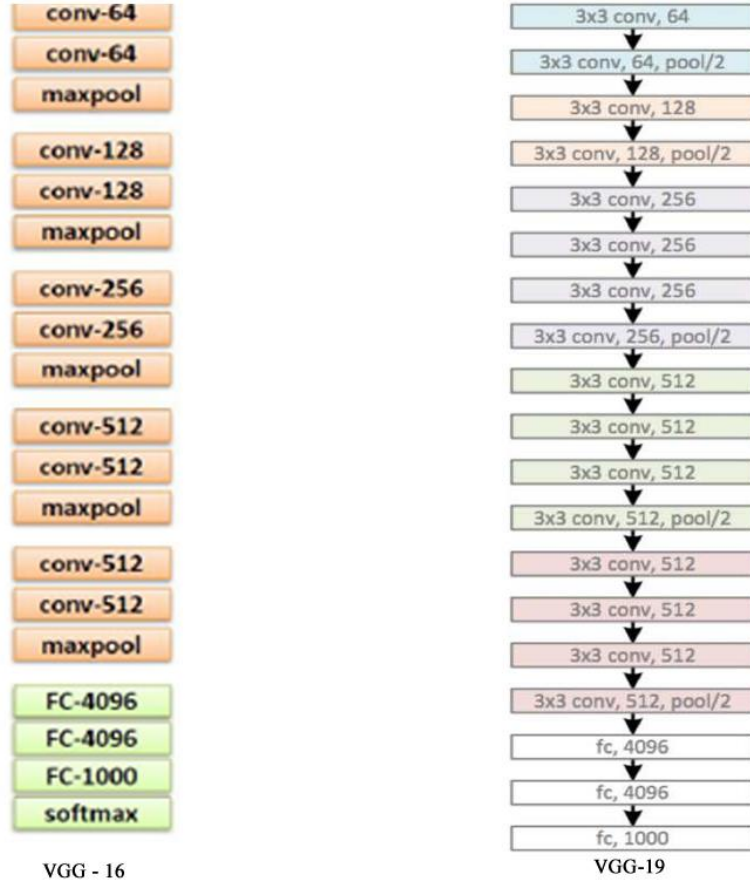
Şekil 4.1 AlexNET Ağ Yapısı

4.2. VGGNET Ağ Yapısı

VGGNET, 2014 yılında ILSRVRC yarışında grafik işlemci birimi (Graphics Processing Unit-GPU) destekli Compute Unified Device Architecture (CUDA) yapılmış konvolüsyonel sinir ağı modelidir. VGGNET, Karen Simonyan ve Andrew Zisserman oluşan Visual Geometry Group tarafından tasarlanmıştır [55]. AlexNET ağ yapısıyla ağ yapıları benzer olup ağırlık konfigürasyonu halka açıktır. Bu ağ % 7,5’luk hata oranı kadar başarı sağlamıştır. Bununla birlikte VGGNET 138 milyon parametreden oluştuğu için işlemesi zordur. Bu modelin yapısı aşağıda şekil 3.3.’deki gibidir. VGGNET ağının VGG-16 ve VGG-19 olmak üzere iki farklı yapısı vardır. VGG-16’da 16 katman bulunurken VGG-19’da 19 katman bulunmaktadır. VGGNET’in bazı ayarlamaları aşağıdaki gibidir:

- Her konvolüsyon işlemi giriş ve çıkış boyutları aynı olup kernel boyutu 3x3’tür.
- Son katmana kadar aktivasyon fonksiyonu “ReLU” son katmanın “Softmax”tır.
- VGG-16’da Fully Connected 14 ve 15 katmanlarından sonra VGG-19 ise Fully Connected 17 ve 18 katmanlarından sonra isteğe bağlı olarak Dropout (0,5) kullanılabilir.

- MaxPooling katmanları standart olup kernel boyutu 2x2 ve strides 2'dir.
- SGD momentum 0,9 ve Learning rate 0,01 alınmalıdır.
- Batch-size 64,128 veya 256 alınabilmektedir.

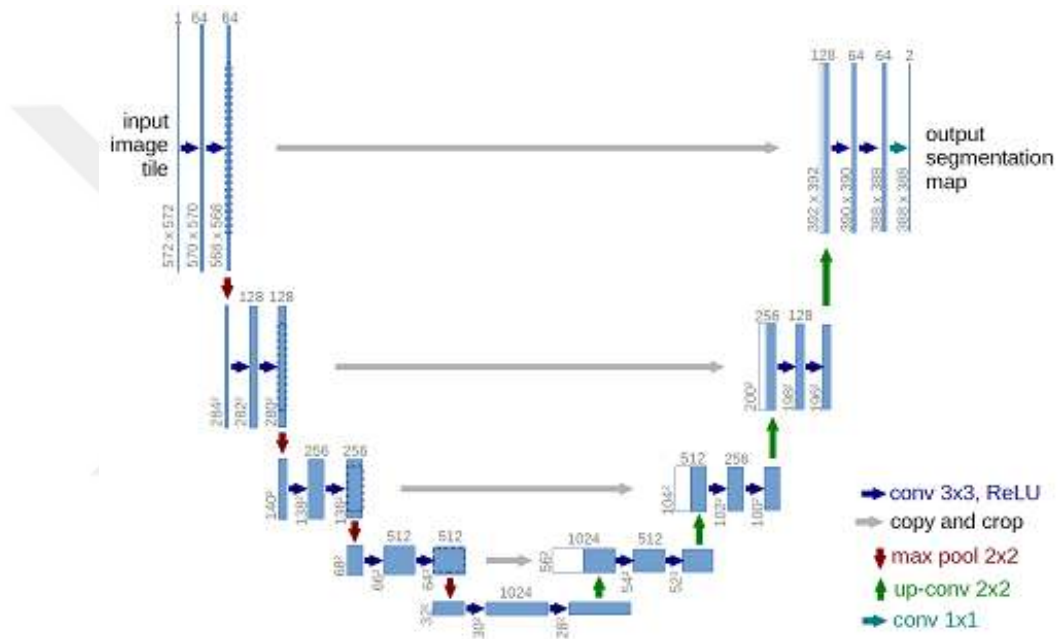


Şekil 4.2 VGGNET Ağ Yapıları

4.3. U-NET Ağ Yapısı

Görüntü segmentasyonu için kullanılan U-Net,2015 yılında Almanya Freiburg Üniversitesi Bilgisayar Bilimleri Bölümündeki Olaf Ronneberger, Philipp Fischer ve Thomas Brox tarafından biyomedikal bölümlere için geliştirilen bir evrimsel sinir ağıdır. Ağ tamamen konvolüsyonel ağa dayanmaktadır ve mimarisi daha az eğitim görüntüsüyle çalışmak ve daha hassas bölümlere sağlamak üzere değiştirilmiş ve genişletilmiştir. U-Net'teki önemli özelliklerden biri, ağın içerik bilgilerini daha yüksek çözünürlük katmanlarına yaymasına izin veren çok sayıda özellik kanalı olmasıdır. Ağ, her bir konvolüsyonun geçerli kısmını, tamamen bağlı herhangi bir katman olmadan kullanır.

Görüntünün sınır bölgesindeki pikselleri tahmin etmek için, eksik bağlam giriş görüntüsünü yansıtarak tahmin edilir. Bu döşeme stratejisi, ağı büyük görüntülere uygulamak için önemlidir. Ağ, bir anlaşıma yolundan ve ona “U” şeklindeki mimariyi veren geniş bir yoldan oluşur. Onaylanan yolu, her biri bir düzeltilmiş doğrusal birim (ReLU) ve bir maksimum havuzlama işlemi tarafından takip edilen konvolüsyon uygulamalarından oluşan tipik bir evrişim ağıdır. Kasılma sırasında, uzamsal bilgiler azaltılırken özellik bilgisi artırılır. Geniş yol, özellik ve mekansal bilgiyi, bir dizi evrişim ve birleştirme ile sözleşme yolundaki yüksek çözünürlüklü özelliklerle birleştirir.



Şekil 4.3 U-Net Ağ Yapısı

5. UYGULAMALAR

5.1. Görüntü Veri Setleri

Makine öğrenmesi üzerine yarışmaların düzenlendiği, açık kaynak şekilde verilerin paylaşıldığı Google LLC tarafından oluşturulan kaggle.com’da farklı amaçlara hizmet eden veri setleri paylaşılmaktadır. Uydu görüntüleri üzerine yapılan bir yarışmadan farklı şehirlere (Paris, Shanghai, Khaourtum, Vegas) ait uydu görüntüleri ve bu uydu görüntülerindeki yol özelliklerinin olduğu geojson dosyaları kullanıldı.

Kaggle.com’da indirilen görüntüler 16 bit RGB formatında görüntülerdir. 16 bitlik görüntüler QGIS benzeri uygulamalar ile görüntülenebilmektedir. 16 bitlik resimler python kodu yazılarak 8 bitlik formata çevrildi ve Şekil 5.1’deki gibidir.



Şekil 5.1 8 Bitlik Uydu Görüntüsü

8 bit formatına çevrilen uydu görüntüleri için etiket (maske) oluşturulması gerekmektedir. Görüntü ismi ve bu görüntü ismine karşılık gelen aynı isimli geojson uzantılı yol özelliklerini (Linestring, multilinestring) taşıyan dosyalar ile yazılan python kodu ile RGB formatındaki görüntünün etiketi (maske) oluşturuldu. Amacımız yapay sinir ağına yol bilgisi öğretmek olduğu için maskelenen görüntüde yol bilgileri dışında geri kalan tüm özelliklerin gösterilmemesi gerekmektedir ve Şekil 5.2’deki gibidir.



Şekil 5.2 8 Bitlik Uydu Görüntüsü(Solda) ve 8 Bitlik Uydu Görüntüsü Maskesi(Sağda)

Maskelenen veride ağın doğru öğrenilmesi için yol çizgilerinin rengi ve kalınlığı önemlidir. İnce çizgilerden oluşan yol özelliklerinin maskeleri ağ tarafından aşırı öğrenmeye sebep olacaktır.

Segmentasyonun değerlendirilme aşamasında kullanılan bazı metrikler ve tanımları şöyledir:

Dice katsayısı, temel gerçeklik (Y) görüntüsüne karşılık gelen öngörülen segmentasyon (X) görüntüsünün piksel bazında karşılaştırılmasıyla denklem 5.1'deki gibi hesaplanır. Özel olarak, bu metrik sadece segmentasyon modeli performansını değerlendirmek için kullanılır.

$$\frac{2 \times |X \cap Y|}{|X| + |Y|} \quad (5.1)$$

Jakard benzerlik katsayısı, hangi örneklerin ortak ve hangilerinin farklı olduğunu görmek için öngörülere (A) temel gerçeklik verileri ile (B) karşılaştırır ve denklem 5.2'deki gibi hesaplanır. Jakard katsayısı ne kadar yüksek olursa, iki alt grup o kadar birbirine benzerdir.

$$\text{Jaccard İndeksi}(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (5.2)$$

5.1.1. Derin Öğrenme Unet Modeli İle Toronto Veri Seti Eğitimi

Bu çalışmanın ikinci adımında Toronto şehri için paylaşılan 8 bitlik uydu görüntüleri ve maskeleri kullanıldı.

Şekil 5.3’de görüldüğü üzere 8 bite çevrilmiş giriş görüntüsü ve geojson verisi ile maskesi uygun formatta doğru çıkarılmış hali;



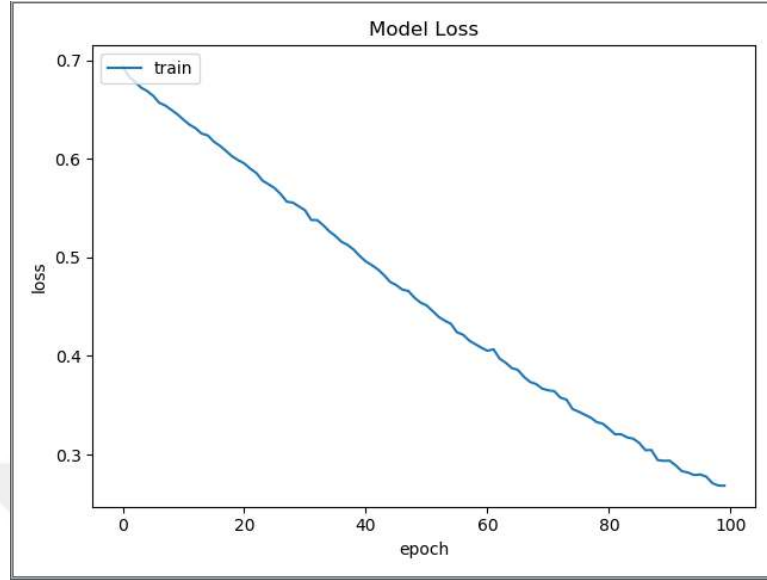
Şekil 5.3 8 Bitlik Resim(Solda) ve Bu Resmin Maskesi(Sağda)

1108 Adet tif dosyasından oluşan Toronto verileri için 100 epoch ve 50 batch size ile eğitim sonucu Tablo 5.1’deki gibidir.

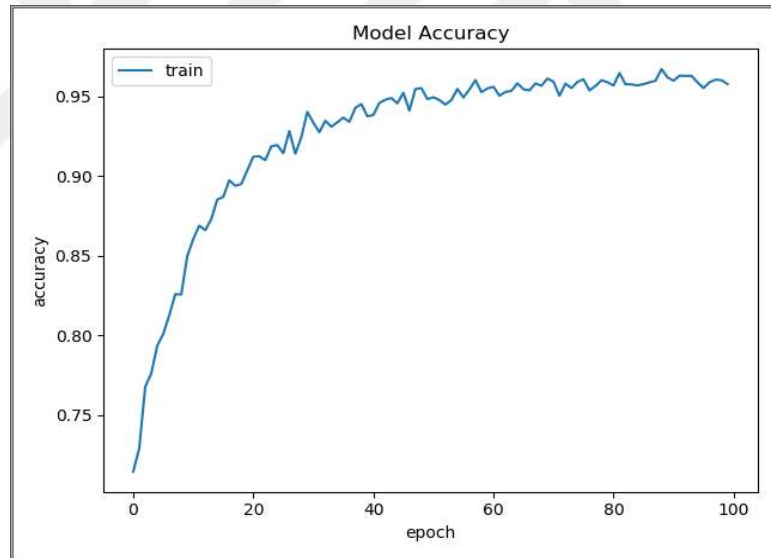
Tablo 5.1 Model Eğitim ve Loss Değeri

Veri seti	Training Accuracy	Val Loss	Epoch	Batch	Veri Kanalı
1108 adet Toronto verisi	%95	0.268	100	50	RGB

Toronto veri seti kullanılarak Unet modeli eğitilmesi sonucu elde edilen hata fonksiyonları Şekil 5.4 ve doğruluk sonuçları Şekil 5.5'deki gibidir.



Şekil 5.4 Hata Fonksiyon Değeri



Şekil 5.5 Model Doğruluğu

Epoch değeri arttırılarak yapılan eğitim sonucu daha başarılı olup hata değeri daha aza indirgenmiştir. RGB (3 Kanallı) girişli eğitim sonucunda tahminleme Şekil 5.6'daki gibidir.



Şekil 5.6 Test Verisi(Solda) ve Tahmini(Sağda)

Hava görüntülerinin segmentasyon sonucunda elde edilen ortalama ve en iyi değerlendirme metrikleri Tablo 5.2’deki gibidir.

Tablo 5.2 Segmentasyon Sonuçları

	Dice Katsayısı	Jaccard Benzerlik Katsayısı
Ortalama	0.76	0.83
En İyi	0.82	0.87

5.1.2. Derin Öğrenme Unet Modeli İle Paris Veri Seti Eğitimi

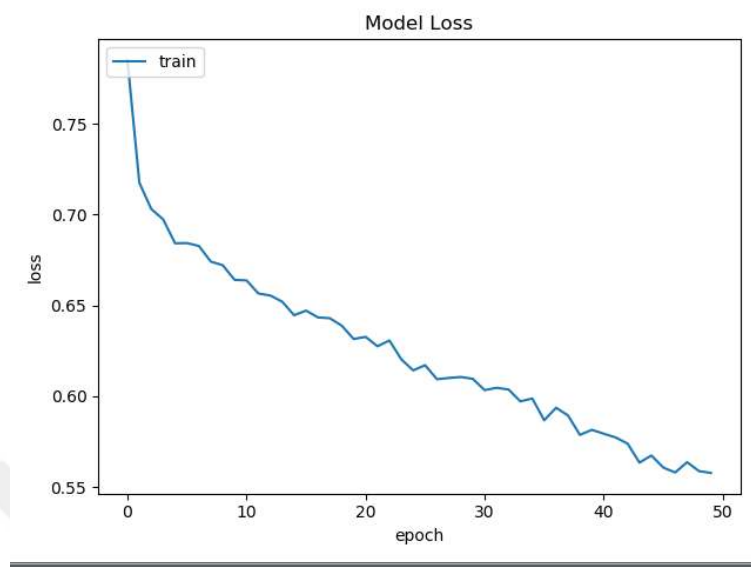
16 Bitlik Paris şehrine ait uydu görüntüleri içeren veriler öncelikle rgb formatında 8 bitlik verilere dönüştürüldü. 8 bitlik verilerin geojson verileri ile yol maskeleri çıkarıldı. Bu veriler üzerinde tekrar eğitim yapıldı.

1400 Adet tif dosyasından oluşan Paris verileri için 50 epoch ve 20 batch size ile eğitim sonucu Tablo 5.3’deki gibidir.

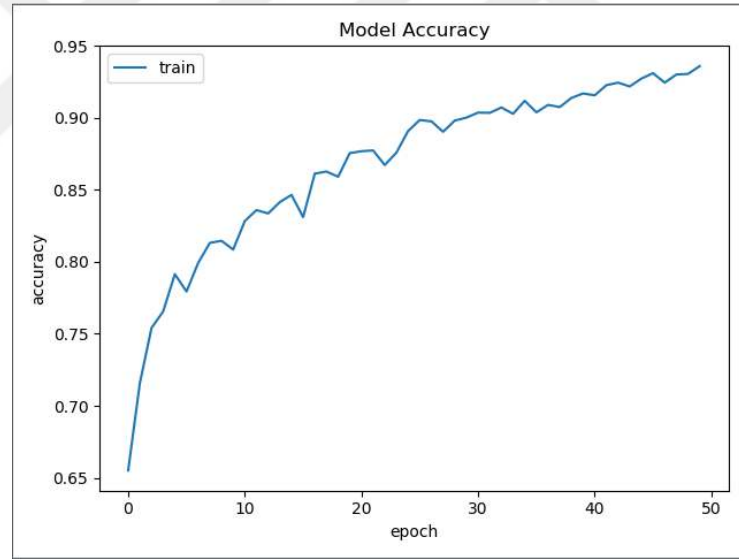
Tablo 5.3 Model Eğitim ve Loss Değeri

Veri seti	Training Accuracy	Val Loss	Epoch	Batch	Veri Kanalı
1400 adet Paris verisi	%93	0.557	50	20	RGB

Paris veri seti kullanılarak Unet modeli eğitilmesi sonucu elde edilen hata fonksiyonları Şekil 5.7 ve doğruluk sonuçları Şekil 5.8'deki gibidir.



Şekil 5.7 Hata Fonksiyon Değeri



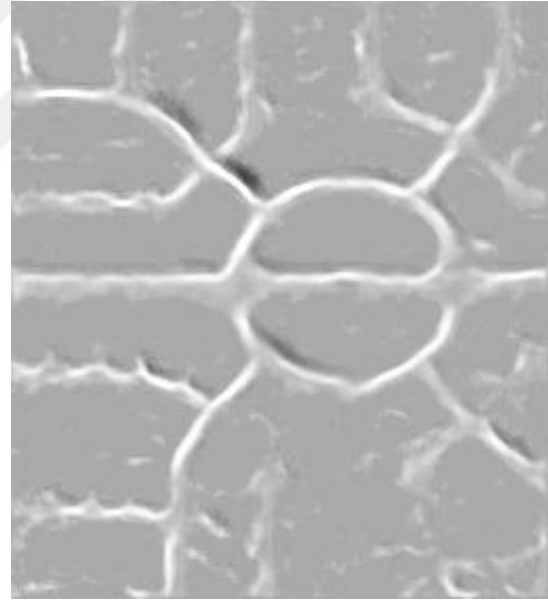
Şekil 5.8 Model Doğruluğu

Şekil 5.9'da görüldüğü üzere 8 bite çevrilmiş giriş görüntüsü ve geojson verisi ile maskesi uygun formatta doğru çıkarılmış hali;



Şekil 5.9 8 Bitlik Resim(Solda) ve Bu Resmin Maskesi(Sağda)

Doğru maskelenmiş veri seti üzerinde yapılan eğitim sonucunda ağ verileri doğru bir şekilde öğrenmiş olup RGB (3 Kanallı) girişli eğitim sonucunda tahminleme Şekil 5.10'daki gibidir.



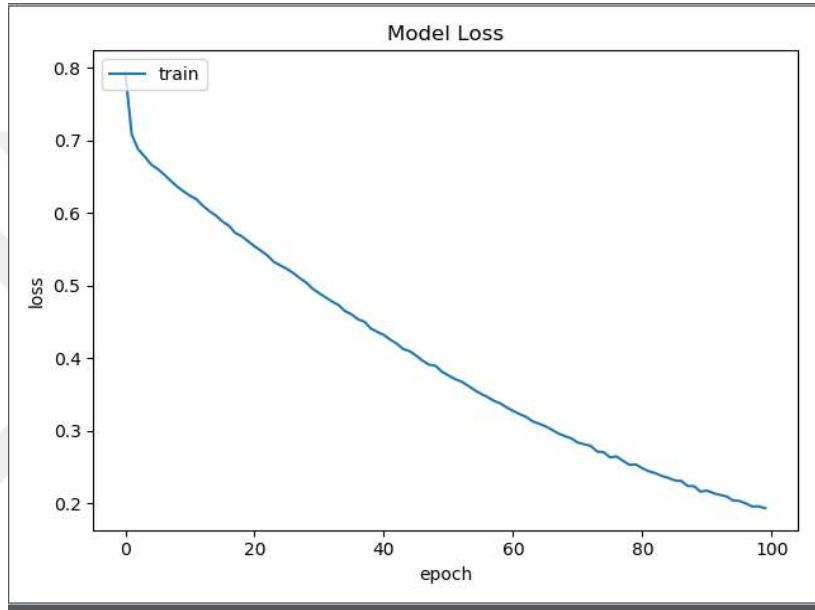
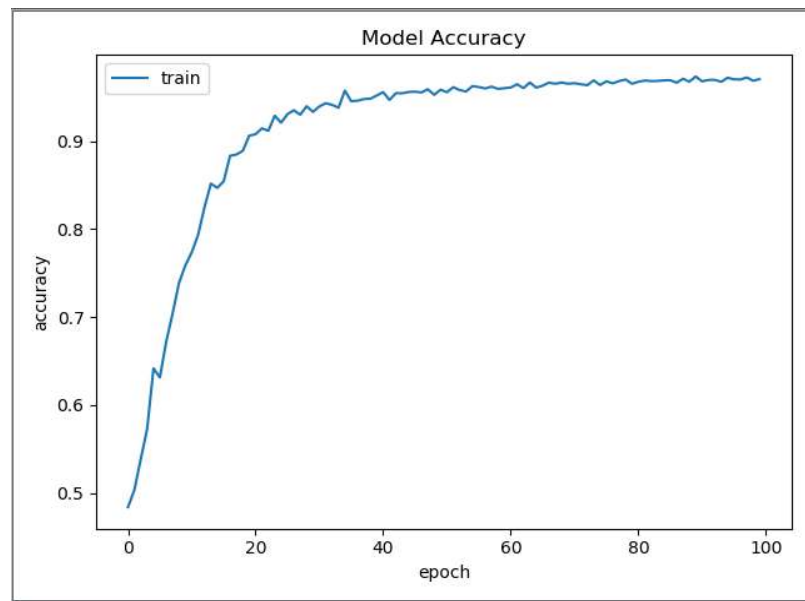
Şekil 5.10 Test Verisi(Solda) ve Tahmini(Sağda)

1400 Adet tif dosyasından oluşan Paris verileri için 100 epoch ve 50 batch size ile eğitim sonucu Tablo 5.4'deki gibidir.

Tablo 5.4 Model Eğitim ve Loss Değeri

Veri seti	Training Accuracy	Val Loss	Epoch	Batch	Veri Kanalı
1400 adet Paris verisi	%97	0.19	100	50	RGB

Paris veri seti kullanılarak Unet modeli eğitilmesi sonucu elde edilen hata fonksiyonları Şekil 5.11 ve doğruluk sonuçları Şekil 5.12'deki gibidir.

**Şekil 5.11** Hata Fonksiyon Değeri**Şekil 5.12** Model Doğruluğu

Epoch değeri arttırılarak yapılan eğitim sonucu daha başarılı olup hata değeri daha aza indirgenmiştir. Öğrenme oranının sürekliliği tahminleme sonucunu olumlu yönde etkilemektedir. RGB (3 Kanallı) girişli eğitim sonucunda tahminleme Şekil 5.13'deki gibidir.



Şekil 5.13 Test Verisi(Solda) ve Tahmini(Sağda)

Hava görüntülerinin segmentasyon sonucunda elde edilen ortalama ve en iyi değerlendirme metrikleri Tablo 5.5'deki gibidir.

Tablo 5.5 Segmentasyon Sonuçları

	Dice Katsayısı	Jaccard Benzerlik Katsayısı
Ortalama	0.68	0.89
En İyi	0.72	0.92

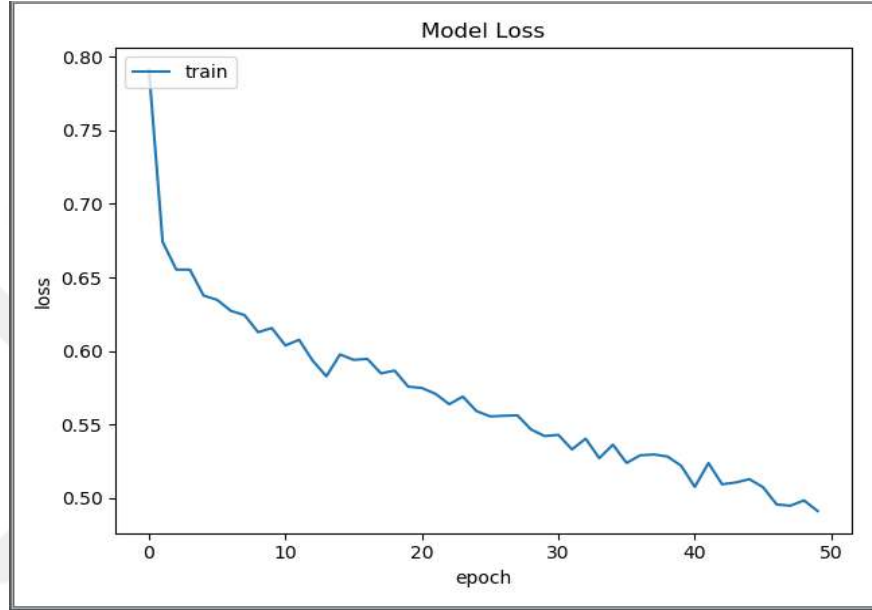
5.1.3. Derin Öğrenme Unet Modeli İle Vegas Veri Seti Eğitimi

16 Bitlik Vegas şehrine ait uydu görüntüleri içeren veriler öncelikle rgb formatında 8 bitlik verilere dönüştürüldü. 8 bitlik verilerin geojson verileri ile yol maskeleri çıkarıldı. Bu veriler üzerinde eğitim yapıldı.

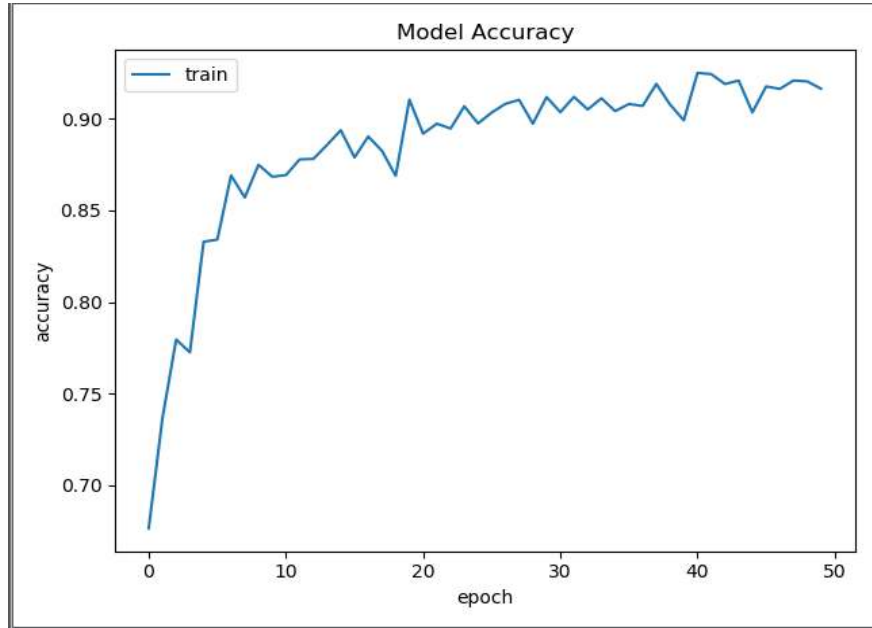
989 Adet tif dosyasından oluşan Vegas verileri için 50 epoch ve 20 batch size ile eğitim sonucu Tablo 5.6'daki gibidir. Vegas veri seti kullanılarak Unet modeli eğitilmesi sonucu elde edilen hata fonksiyonları Şekil 5.14 ve doğruluk sonuçları Şekil 5.15'deki gibidir.

Tablo 5.6 Model Eğitim ve Loss Değeri

Veri seti	Training Accuracy	Val Loss	Epoch	Batch	Veri Kanalı
989 adet Vegas verisi	%91	0.491	50	20	RGB



Şekil 5.14 Hata Fonksiyon Değeri



Şekil 5.15 Model Doğruluğu

Doğru maskelenmiş veri seti üzerinde yapılan eğitim sonucunda ağ verileri doğru bir şekilde öğrenmiş olup RGB (3 Kanallı) girişli eğitim sonucunda tahminleme Şekil 5.16'daki gibidir. Artan veri sayısı ile eğitim sonucunda tahminleme sonucu Paris verilerinin tahminleme sonucunda daha başarılı olmuştur.

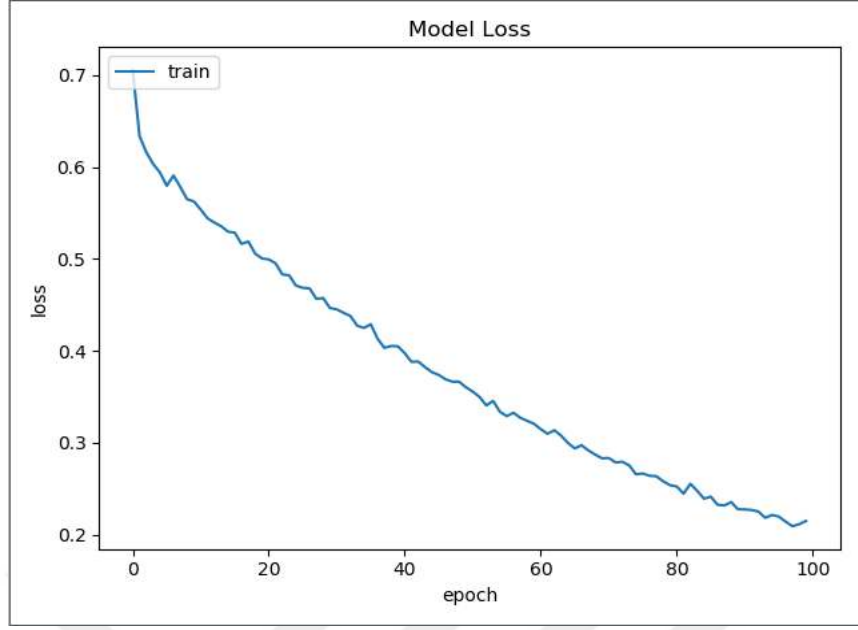


Şekil 5.16 Test Verisi(Solda) ve Tahmini(Sağda)

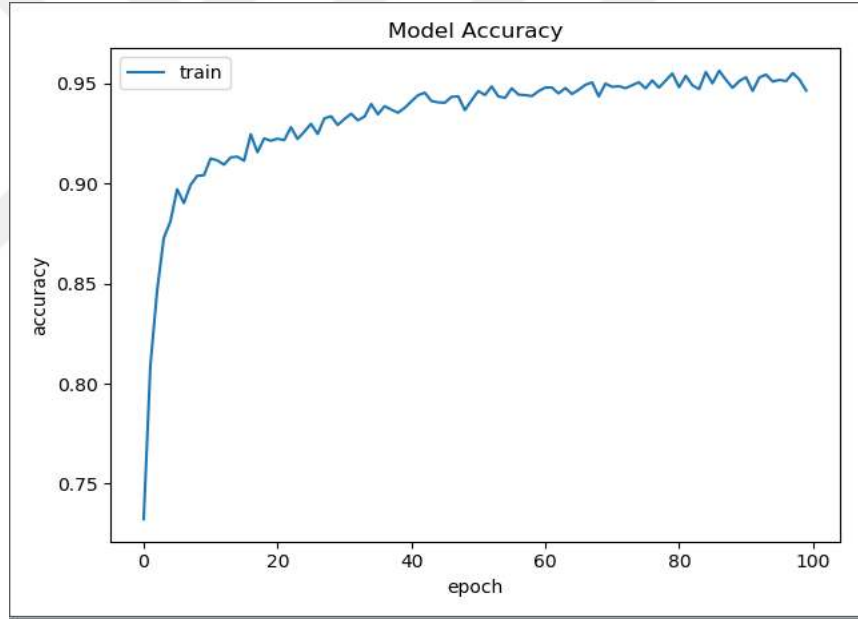
989 Adet tif dosyasından oluşan Vegas verileri için 100 epoch ve 50 batch size ile eğitim sonucu Tablo 5.7'deki gibidir. Vegas veri seti kullanılarak Unet modeli eğitilmesi sonucu elde edilen hata fonksiyonları Şekil 5.17 ve doğruluk sonuçları Şekil 5.18'deki gibidir.

Tablo 5.7 Model Eğitim ve Loss Değeri

Veri seti	Training Accuracy	Val Loss	Epoch	Batch	Veri Kanalı
989 adet Vegas verisi	%94	0.215	100	50	RGB

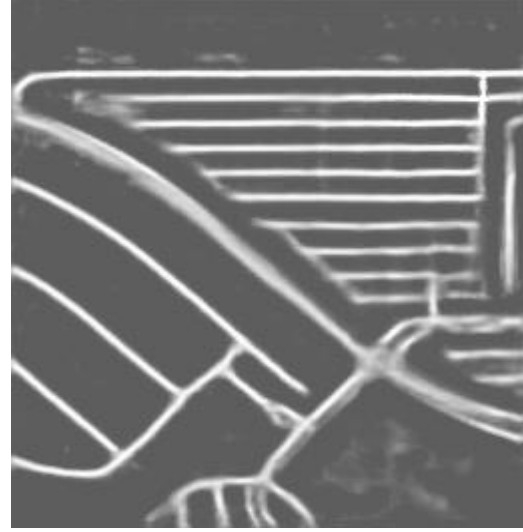
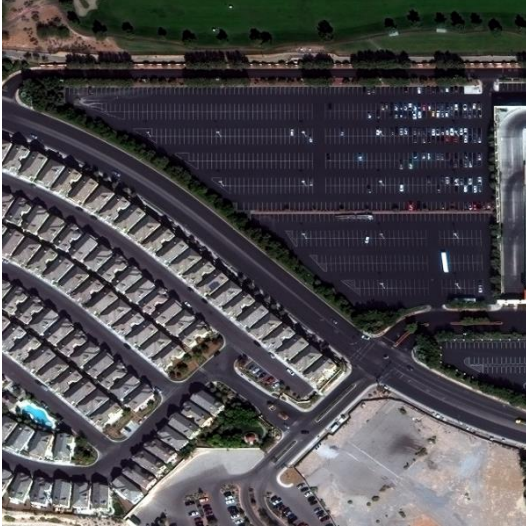


Şekil 5.17 Hata Fonksiyonu Değeri



Şekil 5.18 Model Doğruluğu

Epoch değeri artırılarak yapılan eğitim sonucu daha başarılı olup hata değeri daha aza indirgenmiştir. Öğrenme oranının sürekliliği tahminleme sonucunu olumlu yönde etkilemektedir. RGB (3 Kanallı) girişli eğitim sonucunda tahminleme Şekil 5.19'daki gibidir.



Şekil 5.19 Test Verisi(Solda) ve Tahmini(Sağda)

Hava görüntülerinin segmentasyon sonucunda elde edilen ortalama ve en iyi değerlendirme metrikleri Tablo 5.8’deki gibidir.

Tablo 5.8 Segmentasyon Sonuçları

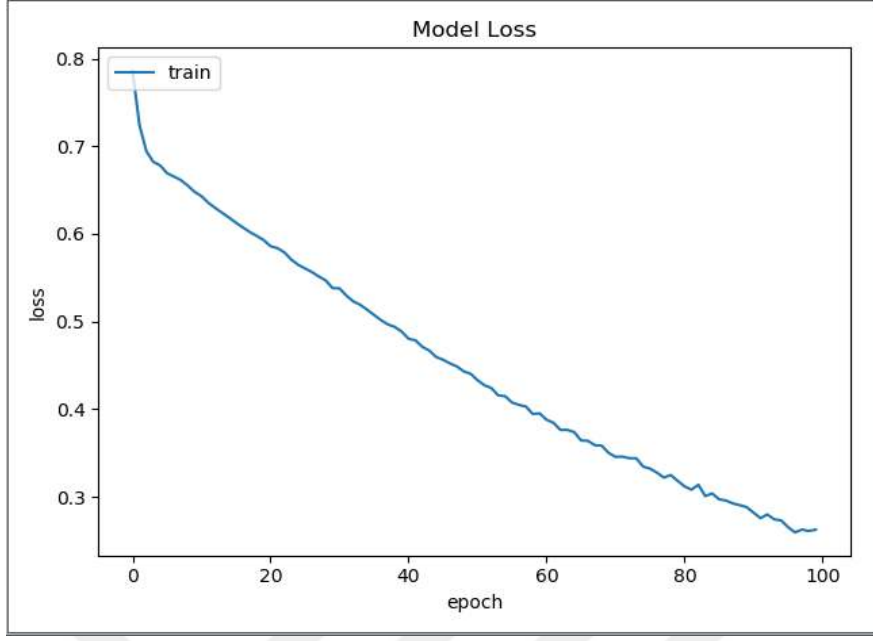
	Dice Katsayısı	Jaccard Benzerlik Katsayısı
Ortalama	0.63	0.88
En İyi	0.70	0.94

5.1.4. Derin Öğrenme Unet Modeli İle Shanghai Veri Seti Eğitimi

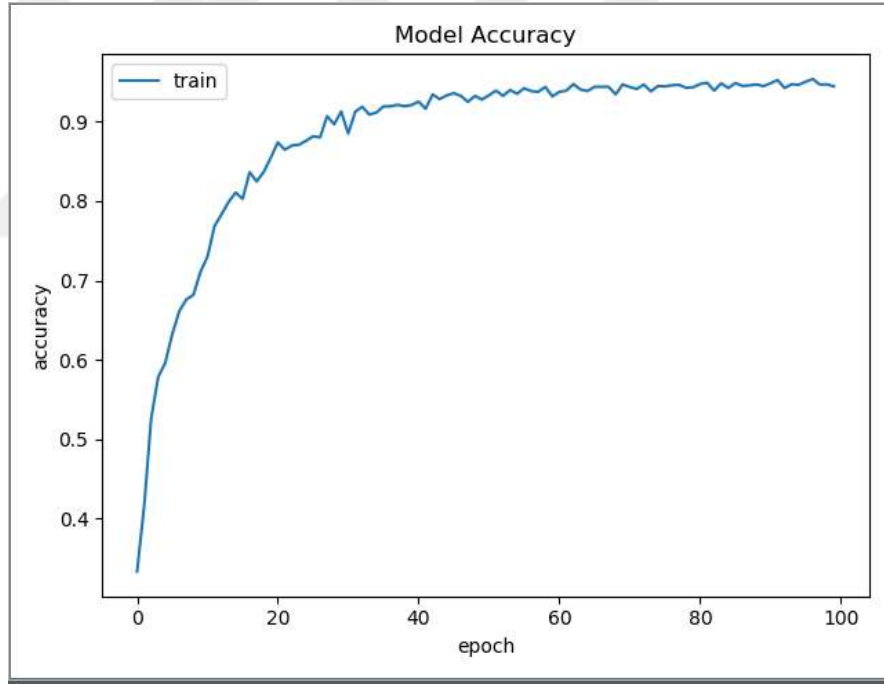
1198 Adet tif dosyasından oluşan Shanghai verileri için 100 Epoch ve 50 Batch size ile eğitim sonucu Tablo 5.9’daki gibidir. Shnagia veri seti kullanılarak Unet modeli eğitilmesi sonucu elde edilen hata fonksiyonları Şekil 5.20 ve doğruluk sonuçları Şekil 5.21’deki gibidir.

Tablo 5.9 Model Eğitim ve Loss Değeri

Veri seti	Training Accuracy	Val Loss	Epoch	Batch	Veri Kanalı
1198 adet Shanghai verisi	%94	0.259	100	50	RGB

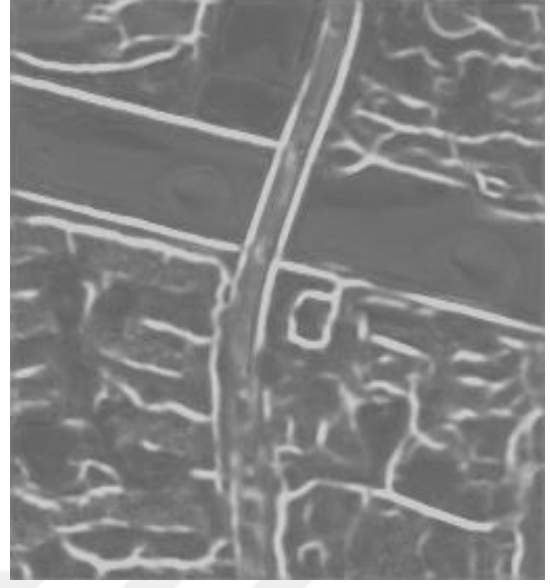
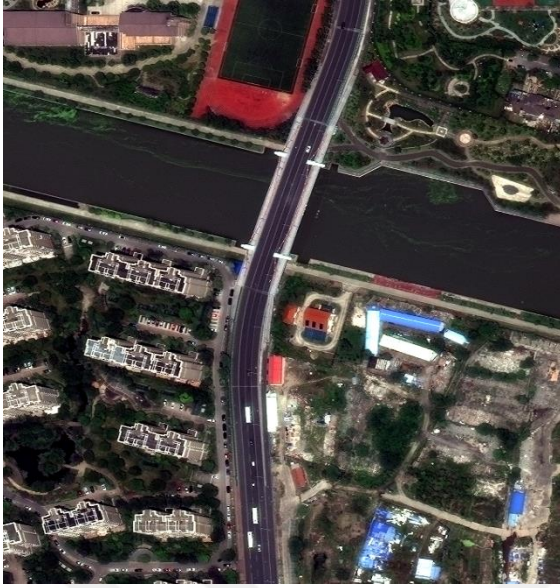


Şekil 5.20 Hata Fonksiyonu Değeri



Şekil 5.21 Model Doğruluğu

RGB (3 Kanallı) girişli eğitim sonucunda tahminleme Şekil 5.22'deki gibidir.



Şekil 5.22 Test Verisi(Solda) ve Tahmini(Sağda)

Hava görüntülerinin segmentasyon sonucunda elde edilen ortalama ve en iyi değerlendirme metrikleri Tablo 5.10'daki gibidir.

Tablo 5.10 Segmentasyon Sonuçları

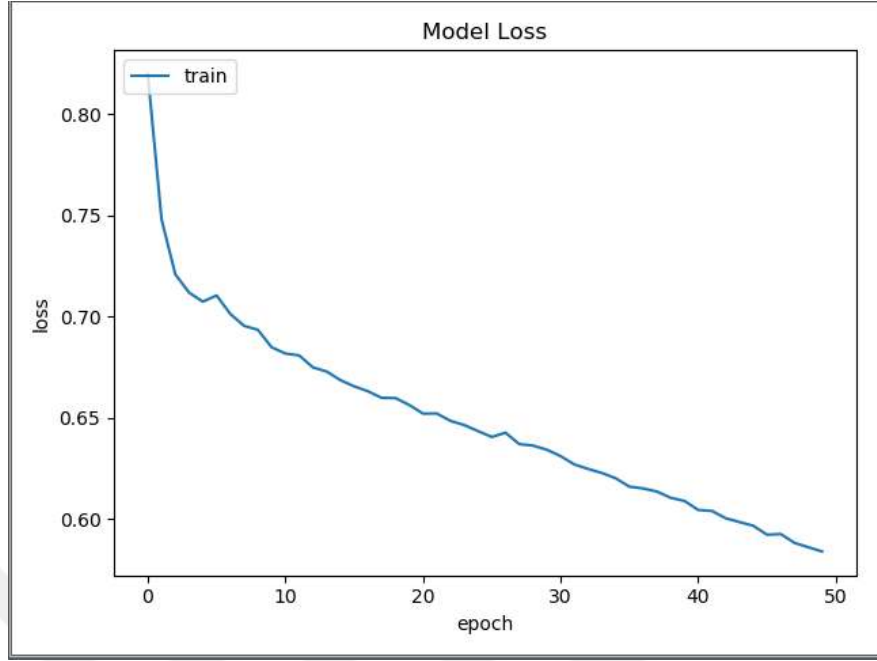
	Dice Katsayısı	Jaccard Benzerlik Katsayısı
Ortalama	0.05	0.87
En İyi	0.08	0.91

5.1.5. Derin Öğrenme Unet Modeli İle Khartoum Veri Seti Eğitimi

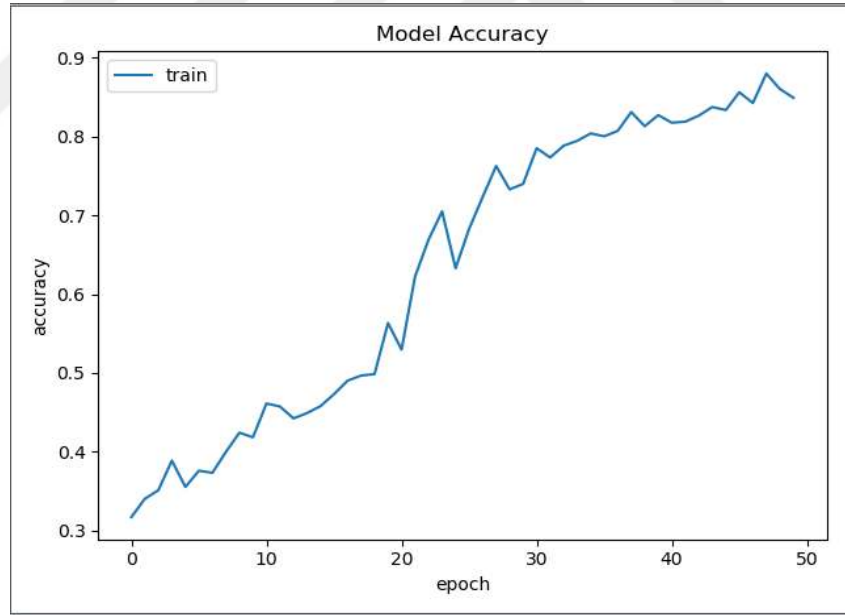
1400 Adet tif dosyasından oluşan Khartoum verileri için 50 epoch ve 20 batch size ile eğitim sonucu Tablo 5.11'deki gibidir. Khartoum veri seti kullanılarak Unet modeli eğitilmesi sonucu elde edilen hata fonksiyonları Şekil 5.23 ve doğruluk sonuçları Şekil 5.24'deki gibidir.

Tablo 5.11 Model Eğitim ve Loss Değeri

Veri seti	Training Accuracy	Val Loss	Epoch	Batch	Veri Kanalı
1400 adet Khartoum verisi	%84	0.584	50	20	RGB



Şekil 5.23 Hata Fonksiyonu Değeri



Şekil 5.24 Model Doğruluğu

RGB (3 Kanallı) girişli eğitim sonucunda tahminleme Şekil 5.25'deki gibidir.

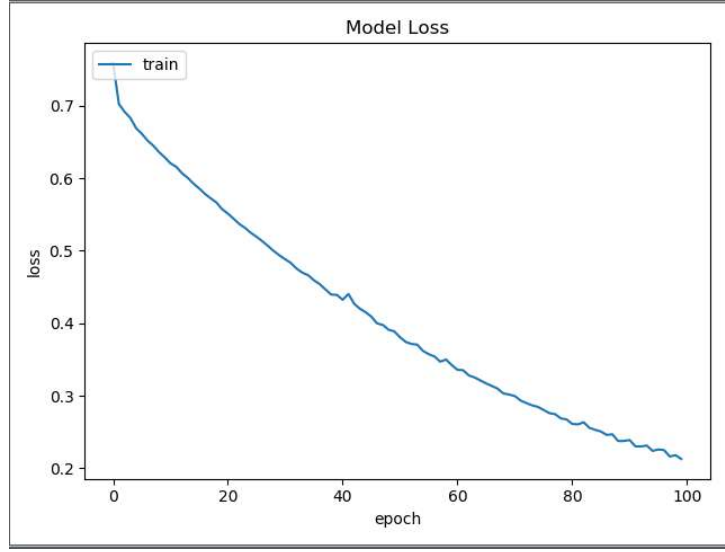


Şekil 5.25 Test Verisi(Solda) ve Tahmini(Sağda)

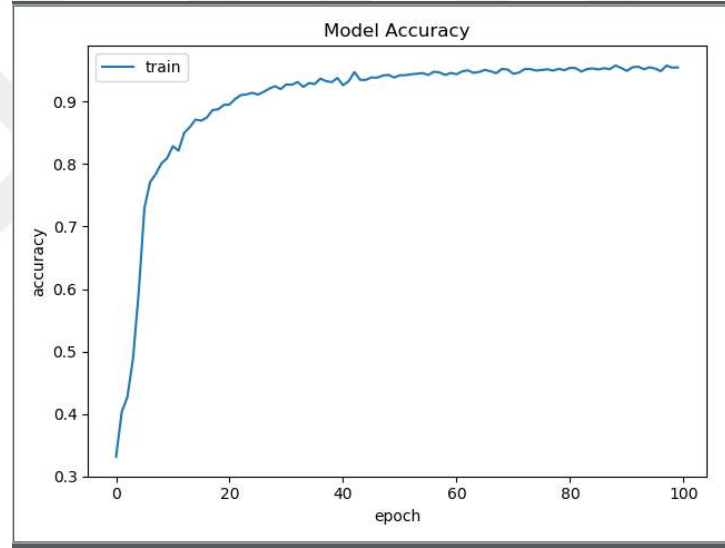
1400 Adet tif dosyasından oluşan Khartoum verileri için 100 epoch ve 50 batch size ile eğitim sonucu Tablo 5.12'deki gibidir. Khartoum veri seti kullanılarak Unet modeli eğitilmesi sonucu elde edilen hata fonksiyonları Şekil 5.26 ve doğruluk sonuçları Şekil 5.27'deki gibidir.

Tablo 5.12 Model Eğitim ve Loss Değeri

Veri seti	Training Accuracy	Val Loss	Epoch	Batch	Veri Kanalı
1400 adet Khartoum verisi	%95	0.212	100	50	RGB

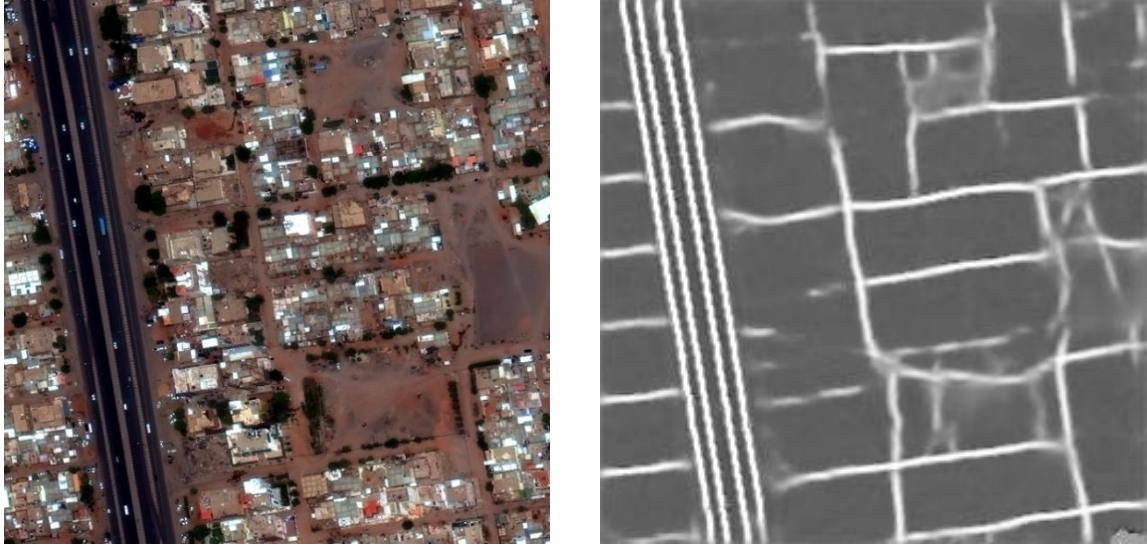


Şekil 5.26 Hata Fonksiyonu Değeri



Şekil 5.27 Model Doğruluğu

Epoch değeri artırılarak yapılan training sonucu daha başarılı olup hata değeri daha aza indirgenmiştir. Öğrenme oranın sürekliliği tahminleme sonucunu olumlu yönde etkilemektedir. RGB (3 Kanallı) girişli eğitim sonucunda tahminleme Şekil 5.28'deki gibidir.



Şekil 5.28 Test Verisi(Solda) ve Tahmini(Sağda)

Hava görüntülerinin segmentasyon sonucunda elde edilen ortalama ve en iyi değerlendirme metrikleri Tablo 5.13’deki gibidir.

Tablo 5.13 Segmentasyon Sonuçları

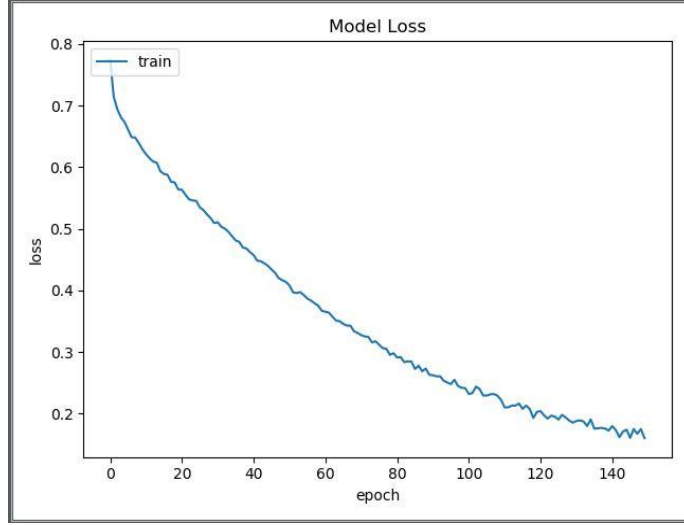
	Dice Katsayısı	Jaccard Benzerlik Katsayısı
Ortalama	0.55	0.85
En İyi	0.68	0.93

5.1.6. Derin Öğrenme Unet Modeli İle Tüm Veri Setleri Eğitimi

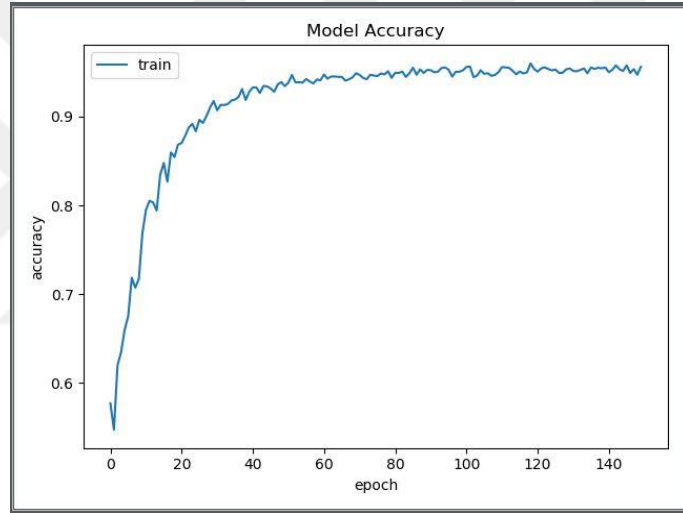
6095 Adet tif dosyasından oluşan Khartoum, Paris, Vegas, Shanghai ve Toronto verileri için 150 epoch ve 50 batch size ile eğitim sonucu Tablo 5.14’deki gibidir. Veri çeşitliliğinin artırılması tahminleme sonucunu farklı veriler için daha başarılı kılacaktır. Tüm veri setleri kullanılarak Unet modeli eğitilmesi sonucu elde edilen hata fonksiyonları Şekil 5.29 ve doğruluk sonuçları Şekil 5.30’daki gibidir.

Tablo 5.14 Model Eğitim ve Loss Değeri

Veri seti	Training Accuracy	Val Loss	Epoch	Batch	Veri Kanalı
6095 adet görüntü verisi	%95	0.160	150	50	RGB



Şekil 5.29 Hata Fonksiyonu Değeri



Şekil 5.30 Model Doğruluğu

RGB (3 Kanallı) girişli eğitim sonucunda tahminleme Şekil 5.31'deki gibidir.



Şekil 5.31 Test Verisi(Solda) ve Tahmini(Sağda)

5.2. Türkiye Görüntü Veri Setleri

Farklı ülkelere ait uydu görüntüleri bilimsel alanda çalışmak üzere hazırlanmışken Türkiye için eğitilebilir formata uydu görüntüleri açık kaynak olarak bulunmamaktadır. Çalışmamızın ülkemiz için katma değerini arttırmak adına Türkiye'nin Elazığ şehri için uydu görüntüleri ve bu uydu görüntülerinin etiketi (maske) oluşturuldu.

Google maps görüntü kalitesi olarak kötü olduğu için Harita Genel Müdürlüğünün resmi uygulaması olan HGM Küre üzerinden kuş bakışı uydu görüntüleri Windows uygulaması (ekran alıntı aracı) ile alındı.

Alınan bu görüntüler Photoshop(CSx) uygulaması ile açıldı. Açılan görüntüye gerçek görüntünün arkasında kalacak yeni bir katman eklendi. Gerçek görüntü üzerinden photoshopun silgi özelliği ile yollar üzerinden geçilerek alt tarafta kalan katmanda silinmiş yolların oluşması sağlandı. Yol bilgisi içeren katmanın arka planı beyaza boyandı. Bu katmanın altında kalacak yeni bir katman eklenerek arka planı siyaha boyandı. İki katman birleştirildikten sonra arka planı siyah ve yol bilgisinin beyaz görüldüğü maske elde edildi. Elde edilen maske tif dosyası olarak kaydedildi. Üzerinde çalışma yapılan orijinal uydu görüntüsü de tif dosyası olarak kaydedildi. Kaydedilen örnek veriler Şekil 5.32'deki gibidir.

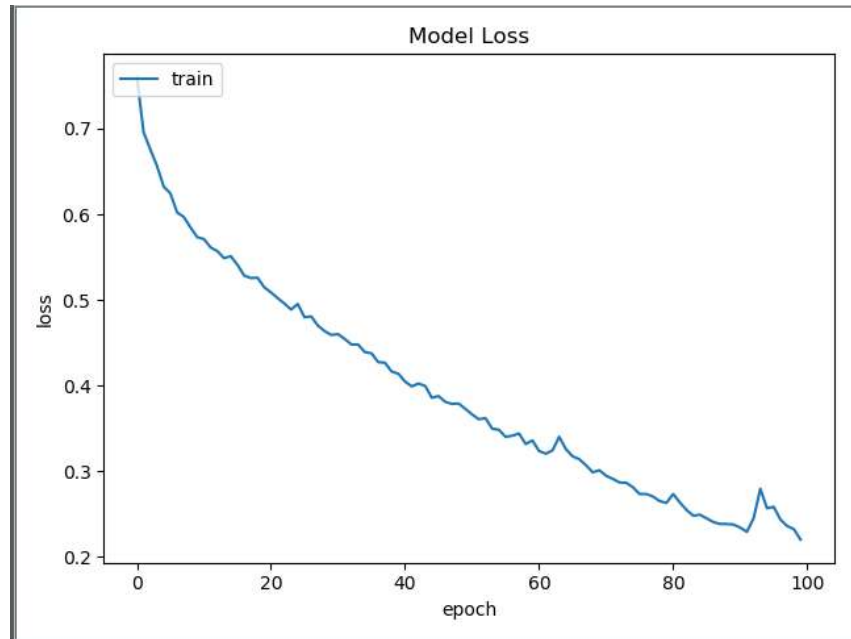


Şekil 5.32 Elazığ Uydu Görüntüsü(Solda) ve Photoshop ile Oluşturulan Maske(Sağda)

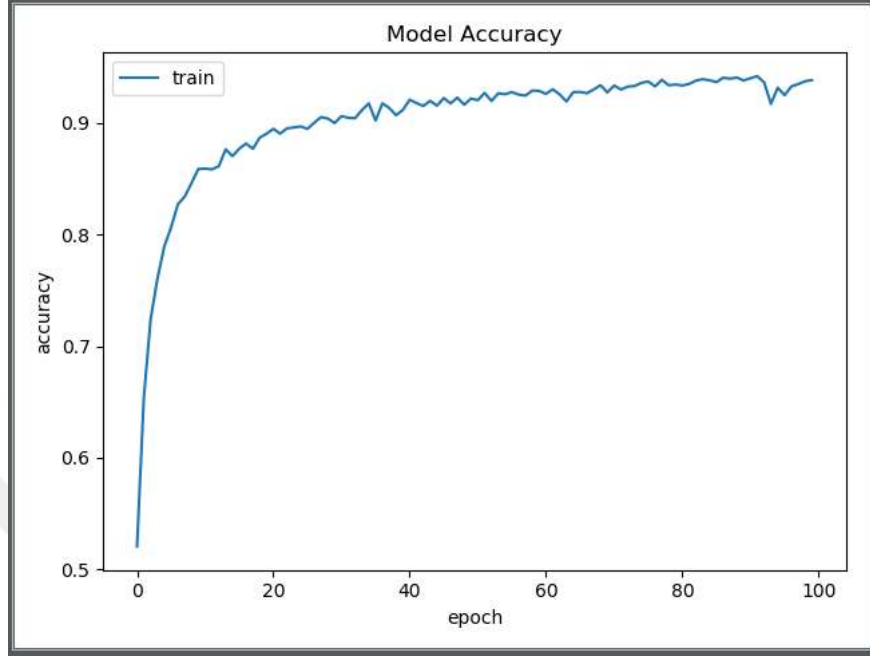
1000 Adet tif dosyasından oluşan Elazığ verileri için 100 epoch ve 50 batch size ile eğitim sonucu Tablo 5.15'deki gibidir. Elazığ veri seti kullanılarak Unet modeli eğitilmesi sonucu elde edilen hata fonksiyonları Şekil 5.33 ve doğruluk sonuçları Şekil 5.34'deki gibidir.

Tablo 5.15 Model Eğitim ve Loss Değeri

Veri seti	Training Accuracy	Val Loss	Epoch	Batch	Veri Kanalı
1000 adet Elazığ verisi	%93	0.220	100	50	RGB

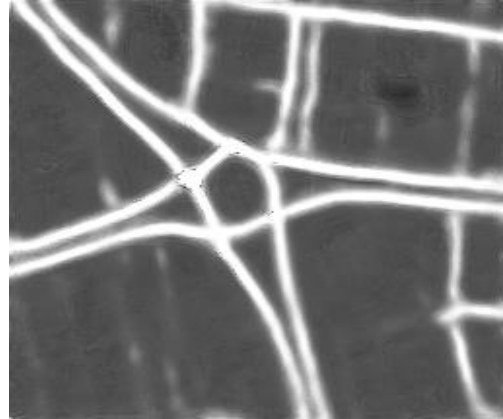


Şekil 5.33 Hata Fonksiyonu Değeri



Şekil 5.34 Model Doğruluğu

RGB (3 Kanallı) girişli eğitim sonucunda tahminleme Şekil 5.35’deki gibidir.



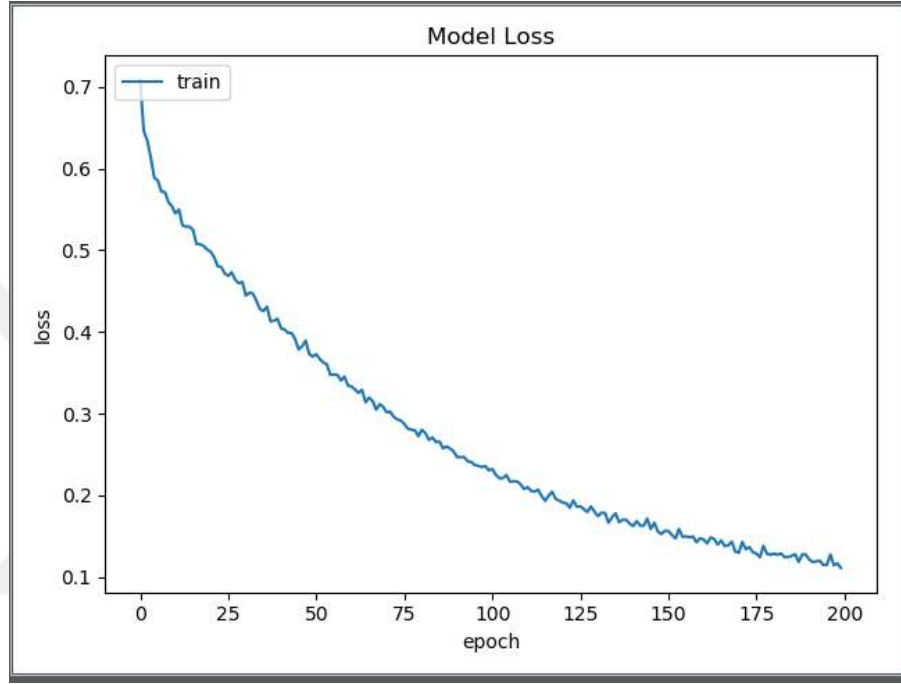
Şekil 5.35 Test Verisi(Solda) ve Tahmini(Sağda)

5.2.1. Derin Öğrenme Unet Modeli İle Elazığ Veri Seti Eğitimi

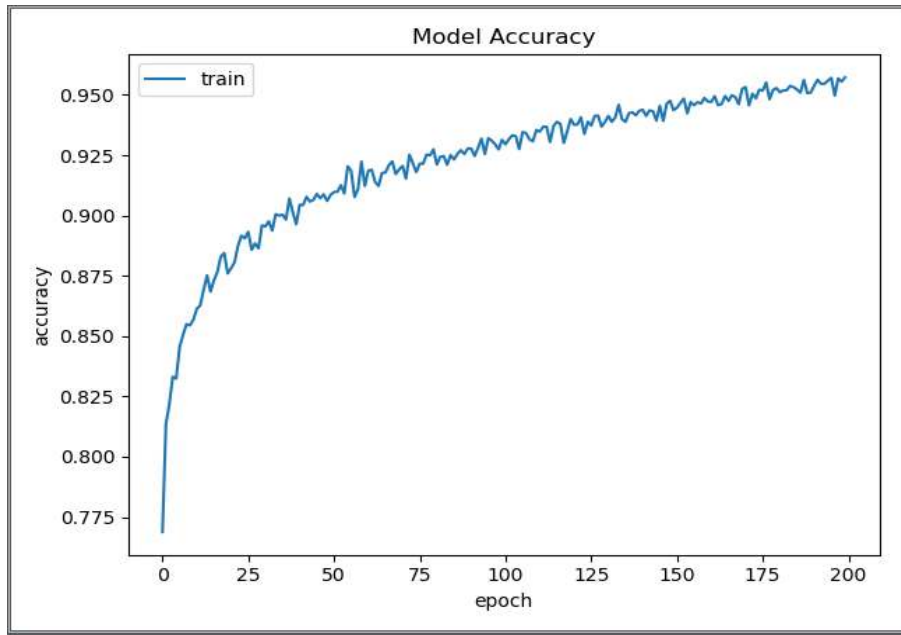
1000 Adet tif dosyasından oluşan Elazığ verileri için 200 epoch ve 50 batch size ile eğitim sonucu Tablo 5.16’daki gibidir. Elazığ veri seti kullanılarak Unet modeli eğitilmesi sonucu elde edilen hata fonksiyonları Şekil 5.36 ve doğruluk sonuçları Şekil 5.37’deki gibidir.

Tablo 5.16 Model Eğitim ve Loss Değeri

Veri seti	Training Accuracy	Val Loss	Epoch	Batch	Veri Kanalı
1000 adet Elazığ verisi	%95	0.111	200	50	RGB

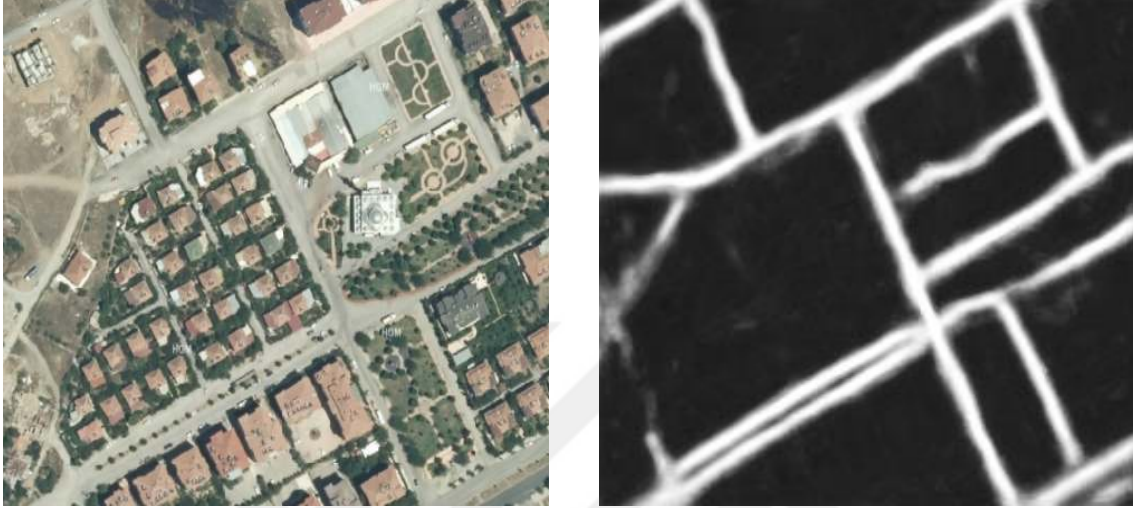


Şekil 5.36 Hata Fonksiyonu Değeri



Şekil 5.37 Model Doğruluğu

Epoch değeri arttırılarak yapılan training sonucu daha başarılı olup hata değeri daha aza indirgenmiştir. Öğrenme oranın sürekliliği tahminleme sonucunu olumlu yönde etkilemektedir. RGB (3 Kanallı) girişli eğitim sonucunda tahminleme Şekil 5.38'deki gibidir.



Şekil 5.38 Test Verisi(Solda) ve Tahmini(Sağda)

Hava görüntülerinin segmentasyon sonucunda elde edilen ortalama ve en iyi değerlendirme metrikleri Tablo 5.17'deki gibidir.

Tablo 5.17 Segmentasyon Sonuçları

	Dice Katsayısı	Jaccard Benzerlik Katsayısı
Ortalama	0.81	0.79
En İyi	0.87	0.85

5.2.2. Malatya Uydu Görüntülerine Ait Veri Setinin Hazırlanması ve Unet Modeli ile Eğitimi

Çalışmamızın ülkemiz için katma değerini arttırmak adına Türkiye'nin Malatya şehri için uydu görüntüleri ve bu uydu görüntülerinin etiketi (maske) oluşturuldu.

Malatya şehrindeki yollara ait coğrafik veriler temin edildi. Bu veriler Şekil 5.39'da gösterilen GEOSERVER denilen geospatial verilerin yayınlanabildiği, güncellenebildiği ve işlenebildiği açık kaynak bir sunucuda yayınlanarak WMS (Web Map Service) katmanı oluşturuldu. Yolların kalınlıkları ve renkleri değiştirildi. WMS katmanı Şekil 5.40'daki gibidir.



Şekil 5.39 Geoserver



Şekil 5.40 Yol Verilerinin WMS Katmanı Üzerindeki Görüntüsü

OpenLayers katmanı olarak Google maps ve yol verilerini içeren WMS katmanları eklendi ve katmanlar opsiyonel olarak seçilebilir yapıldı. OpenLayers katmanları Şekil 5.41'deki gibidir.



Şekil 5.41 WMS ve Google Maps Katmanlarının Birleştirilmesi

Eklenen Boş bir katman sayesinde kesilerek alınacak olan uydu görüntüsünün maskesini elde etmemizi sağlayacaktır. Tüm yolların maskesi Şekil 5.42'deki gibidir. Üzerinde eğitim yapılacak olan uydu görüntüsü ve maskesi jpg dosyası olarak kaydedildi ve Şekil 5.43'deki gibidir.



Şekil 5.42 Yol Verilerinin Maskesi

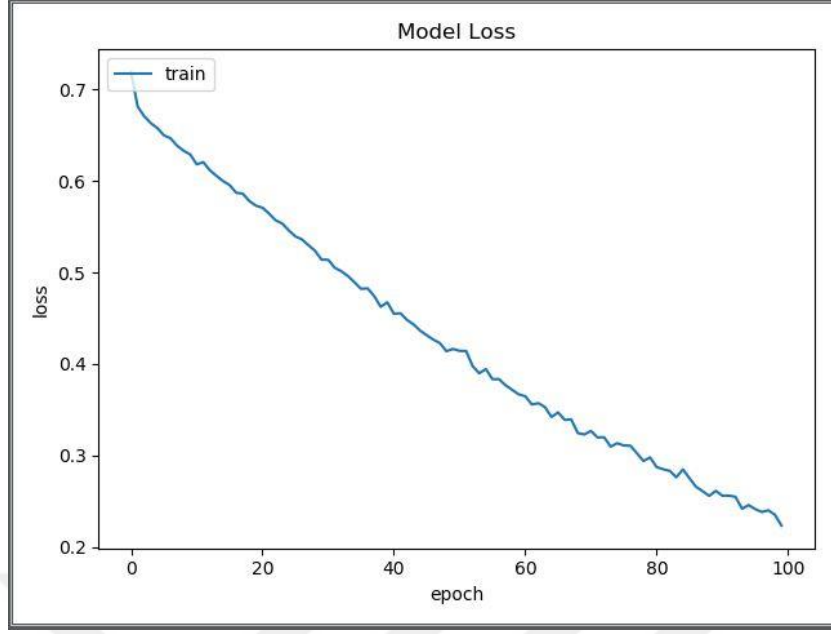


Şekil 5.43 Görüntü Kesiti(Solda) ve Görüntü Kesitinin Maskesi(Sağda)

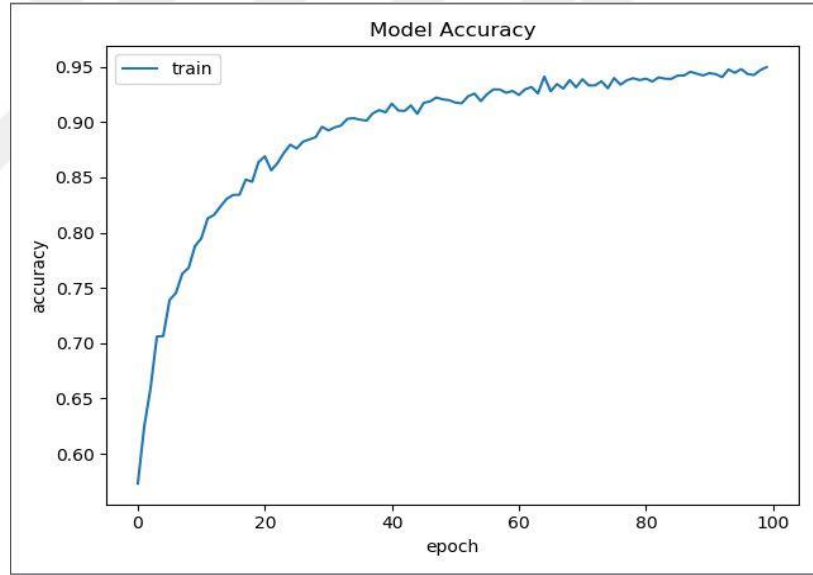
1000 Adet jpg dosyasından oluşan Malatya verileri için 100 epoch ve 50 batch size ile eğitim sonucu Tablo 5.18'deki gibidir. Malatya veri seti kullanılarak Unet modeli eğitilmesi sonucu elde edilen hata fonksiyonları Şekil 5.44 ve doğruluk sonuçları Şekil 5.45'deki gibidir.

Tablo 5.18 Model Eğitim ve Loss Değeri

Veri seti	Training Accuracy	Val Loss	Epoch	Batch	Veri Kanalı
1000 adet Malatya verisi	%94	0.223	100	50	RGB

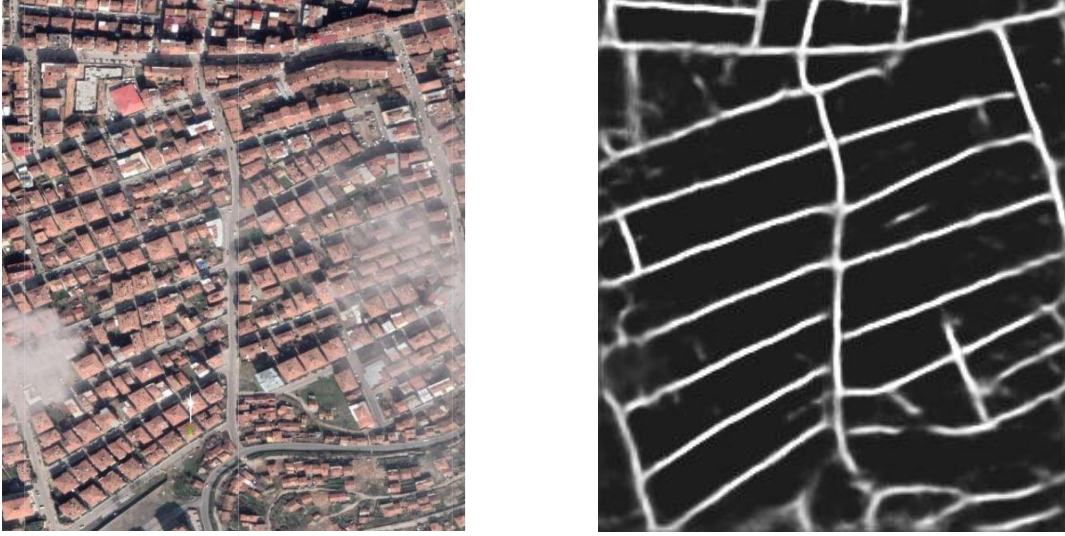


Şekil 5.44 Hata Fonksiyonu Değeri



Şekil 5.45 Model Doğruluğu

RGB (3 Kanallı) girişli eğitim sonucunda tahminleme Şekil 5.46'daki gibidir.

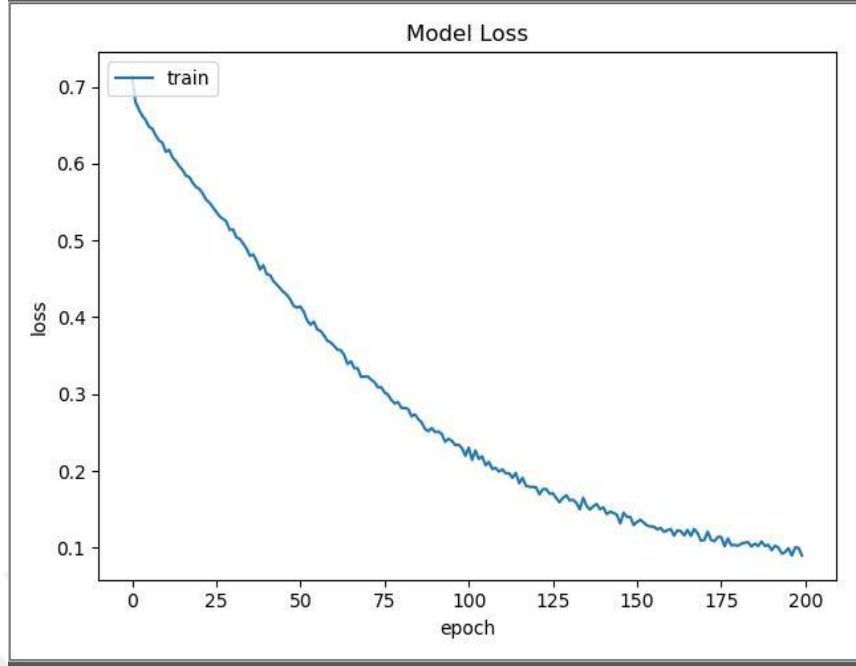


Şekil 5.46 Test Verisi(Solda) ve Tahmini(Sağda)

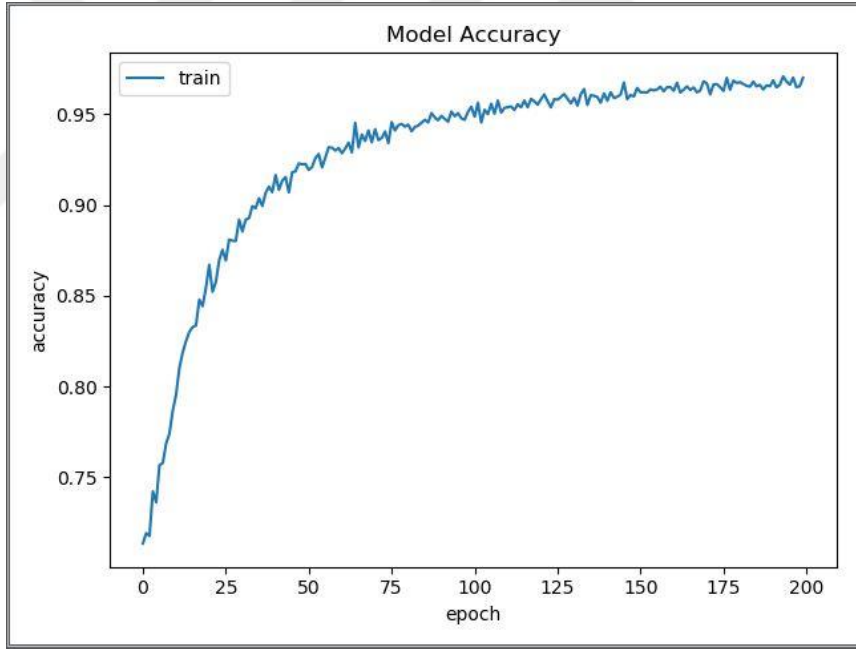
1000 Adet jpg dosyasından oluşan Malatya verileri için 200 epoch ve 50 batch size ile eğitim sonucu Tablo 5.19'daki gibidir. Malatya veri seti kullanılarak Unet modeli eğitilmesi sonucu elde edilen hata fonksiyonları Şekil 5.47 ve doğruluk sonuçları Şekil 5.48'deki gibidir.

Tablo 5.19 Model Eğitim ve Loss Değeri

Veri seti	Training Accuracy	Loss Val	Epoch	Batch	Veri Kanalı
1000 adet Malatya verisi	%97	0.089	200	50	RGB



Şekil 5.47 Hata Fonksiyonu Değeri



Şekil 5.48 Model Doğruluğu

Epoch değeri arttırılarak yapılan training sonucu daha başarılı olup hata değeri daha aza indirgenmiştir. Öğrenme oranının sürekliliği tahminleme sonucunu olumlu yönde etkilemektedir. RGB (3 Kanallı) girişli eğitim sonucunda tahminleme Şekil 5.49'daki gibidir.



Şekil 5.49 Test Verisi(Solda) ve Tahmini(Sağda)

Hava görüntülerinin segmentasyon sonucunda elde edilen ortalama ve en iyi değerlendirme metrikleri Tablo 5.20'deki gibidir.

Tablo 5.20 Segmentasyon Sonuçları

	Dice Katsayısı	Jaccard Benzerlik Katsayısı
Ortalama	0.21	0.83
En İyi	0.39	0.91

5.3. Tahminleme Sonucunda Oluşan Görüntüdeki Yolların Geographic Verilerinin Çıkarılması

Tahminlemesi yapılan bir hava görüntüsündeki yol ağının haritadaki enlem ve boylam karşılıklarını bulmak için çeşitli coğrafi bilgi sistemleri yöntemlerinden yararlanıldı.

Yol ağının çıkarılma adımları;

- Google Maps'ten 4 köşe noktasına denk gelen enlem ve boylam değerleri bilinen bir görüntünün ekran alıntısı aracı ile kesilmesi ve deep learning ile tahminlemesinin yapılması.



Şekil 5.50 Dört noktasının enlem ve boylam değeri bilinen Test Verisi(Solda) ve Tahmini(Sağda)

- Tahmini görüntünün piksellerinin çıkarılması ve piksel renginden siyah zemin ve beyaz yolun ayrıştırılması(Renk aralığı [0-255])
- Yol içerdiği düşünülen beyaz piksellerin renk aralığı için eşik değerinin belirlenmesi(Yazılan uygulamada eşik değeri 100 seçilmiştir.)
- Enlem, Boylam değeri bilinen iki GPS koordinat noktası arasındaki mesafenin **HAVERSINE** formülü ile km türünden bulunması. Haversine formülü denklem 5.3'deki gibidir.
- Haversine formülü iki GPS koordinatı arasındaki en kısa mesafeyi ölçmek için kullanılır.
- Başlangıç noktasından bir sonraki noktaya olan en kısa mesafenin ölçülmesi için **BEARING** formülünün kullanılması. Bearing formülü denklem 5.4'deki gibidir.
- Bearing formülünde de amaç iki GPS koordinatı arasındaki en iyi çemberi yani en kısa mesafeyi bulmaktır.
- Çember içine alınan bir noktadan sonsuz tane çizgi çizilebilir. Bu çizgilerden hangisinin yol bilgisi içeren noktaya denk geldiği ve bulunan hypotenuse uzunluğunu içerdiği bilgisini bulmamız gerekir.
- Görüntünün bütünüyle bir koordinat düzlemi gibi kullanılması ve değeri bilinen X,Y piksel ikilisinin orijin noktası ile yaptığı açısının bulunması.
- $\sin A = Y/\text{hypotenuse}$

- A açısının radyana çevrilmesi
- Açısı ve başlangıç noktası ve başlangıç noktasına olan mesafenin bilindiği parametrelerden bearing formülü ile bir sonraki noktanın bulunması.

Haversine Denklemi:

$$a = \sin^2\left(\varphi B - \frac{\varphi A}{2}\right) + \cos\varphi A * \varphi B * \sin^2\left(\lambda B - \frac{\lambda A}{2}\right) \quad (5.3)$$

$$c = 2 * \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R * c$$

Bearing Denklemi:

$$\Delta\psi = \ln\left(\frac{\tan\left(\frac{\pi}{4} + \frac{\varphi_2}{2}\right)}{\tan\left(\frac{\pi}{4} + \frac{\varphi_1}{2}\right)}\right) \quad (5.4)$$

$$\theta = \text{atan2}(\Delta\gamma, \Delta\psi)$$

İzlenilen adımlar sonucunda yazılan uygulamadan çıktı olarak multi point türünde geodata alınır. Alınan Geodata yazılan başka bir uygulamda Google maps üzerine bir katman şeklinde gösterilir.



Şekil 5.51 Dört noktasının enlem ve boylam değeri bilinen Tahmini Veri (Solda) ve Geodata çıkarılan Görüntü (Sağda)

6. SONUÇ

Bu tez çalışmasına yüksek çözünürlüklü hava görüntü veri setleri ve çeşitli derin öğrenme mimarileri araştırılmış ve bu anlamda çalışmalar yapılmıştır. Derin öğrenme ile ağ eğitilirken önemli bir parametre veri miktarıdır. Test ve eğitim veri miktarının yeteri kadar olması başarı oranını da arttırmıştır. Veri setinin yetersiz olmasına bağlı olarak hiper parametrelerin uygun seçilmesi yetersiz veri seti üzerinde de başarılı sonuçlar sergilemiştir.

Vegas, Toronto, Paris, Khartoum ve Shanghai şehirlerine ait geotif dosyalarından oluşan veri setleri (görüntü ve bu görüntüye karşılık gelen maske) kullanılarak farklı giriş parametreleri ile görüntü segmentasyonu yapılmıştır. Türkiye için veri setleri mevcutta olmayıp Elazığ ve Malatya şehirleri için üzerinde çalışabilecek sayıda uydu görüntüsü ve maskesi oluşturulmuştur. Elazığ için oluşturulan veri seti photoshop uygulaması ile maskesi çıkarılırken görüntüler tif dosyası şeklinde oluşturulmuştur. Malatya verileri geoserverdan kesit alındığı için jpg olarak oluşturulmuştur. Verilerin %90'ı eğitim, %10'u test verisi olarak kullanılmış olup validation için veri ayırımına gerek duyulmamıştır.

Vegas, Toronto, Paris, Khartoum, Shanghai ve Elazığ verilerinin tamamı bir araya getirilerek eğitim gerçekleştirilmiştir. Bu eğitim sonucunda model farklı özellikteki görüntüleri öğrenmiş olup tahminleme aralığı daha da genişlemiştir.

Segmentasyonda U-Net ağ modeli kullanılmıştır. Epoch ve batch size giriş parametreleri değiştirilerek aynı veri setleri üzerinde yapılan farklı eğitim sonuçları gözlemlenmiş ve eğittiği verileri daha iyi tahminleye bildikleri sonucu çıkarılmıştır. Eğitilen uydu görüntüsünün yakınlık ve uzaklık seviyesi farklı veriler üzerinde yaptığı tahminleme sonucunu etkilemiştir. Yakın mesafede alınan ve eğitilen görüntülerin hem uzak hem de yakın mesafede çekilmiş görüntülerde daha başarılı tahminleme yaptığı gözlemlenmiştir. Hava görüntülerinin segmentasyon sonucunda elde edilen ortalama ve en iyi değerlendirme metrikleri olarak Dice ve Jaccard metrikleri kullanılmış olup tahminleme oranları çıkarılmıştır.

Eğitilen görüntülerin tif dosya uzantılı olması ve jpg dosya uzantılı olması eğitim değerlerinde (doğruluk, hata oranı) ciddi bir farklılık göstermemesine rağmen tahminleme sonucunda farklılıklar göstermiştir. Tif dosyalarında görüntü kalitesi jpg dosyalarına göre daha iyi olduğu gözlemlenmiştir. Tif uzantılı uydu görüntüleriyle eğitilen modellerde tahminleme sonucu daha başarılı olduğu gözlemlenmiştir.

Türkiye için derin öğrenme çalışmalarında kullanılabilecek veri seti eksikliği giderilmiş olup oluşturulan veri setleri ve veri setlerinin oluşturulma yöntemleri açık kaynak şeklinde paylaşılacaktır. Var olan coğrafik verilerden yeni coğrafik verilerin elde edilmesi sağlanmıştır. Eğitilen sinir ağının yaptığı tahminleme sonucunda görüntüdeki yol ağının HAVERSINE ve BEARING formülleri kullanılarak geliştirilen algoritma ile coğrafik verileri çıkarılmıştır. Coğrafi Bilgi Sistemleri alanında derin öğrenme teknolojisi kullanılarak yeni ve katma değeri yüksek çalışmaların yapılması amaçlanmaktadır.

Bu tez çalışmasının daha başarılı sonuçlar vermesi için çalışmalar devam edecek ve gerçek hayatta kullanılabilecek uygulamalar haline getirilecektir. Verilerin en doğru şekilde elde edilmesi ve kullanılabilmesi için gerekli kurum ve kuruluşlarla ortak işler yapılacaktır.

KAYNAKLAR

- [1] **Aras, İ., Yıldız, F.**, 2011, “İnternet Tabanlı CBS’nin Sivil ve Askerî Amaçlı Acil Durum Uygulamalarında Kullanılmasında Yeni Bir Yaklaşım”, *Harita Dergisi*, 145: 38-51
- [2] **Bengio, Y., and LeCun, Y.**(2007) “In Large Scale Kernel Machines,” MIT Press.
- [3] **D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber**, “Multi-column deep neural network for traffic sign classification,” *Neural Networks*, vol. 32, pp. 333–338, 2012.
- [4] **F. Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. E. Barbano**, “Toward automatic phenotyping of developing embryos from videos,” *IEEE Transactions on Image Processing*, vol. 14, no. 9, pp. 1360–1371, 2005.
- [5] **J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler**, “Efficient object localization using convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 648–656.
- [6] **P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun**, “Pedestrian detection with unsupervised multi-stage feature learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3626–3633.
- [7] **Li Deng, Dong Yu**, 2014. “Deep Learning Methods and Applications”, *Foundations and Trends Signal Processing*, vol.7, 198-250.
- [8] **Xia, Wei, et al.** "A weakly supervised road extraction approach via deep convolutional nets based image segmentation." *2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP)*. IEEE, 2017.
- [9] **Ale, Laha, Ning Zhang, and Longzhuang Li**. "Road Damage Detection Using RetinaNet." *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018.
- [10] **Zhang, Lei, et al.** "Road crack detection using deep convolutional neural network." *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016.
- [11] **Wang, Yanbo J., et al.** "Deep Proposal and Detection Networks for Road Damage

Detection and Classification." *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018.

[12] **Kulkarni, Ruturaj, Shruti Dhavalikar, and Sonal Bangar.** "Traffic Light Detection and Recognition for Self Driving Cars Using Deep Learning." *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE, 2018.

[13] **Zhang, Zhaojin, Cunlu Xu, and Wei Feng.** "Road vehicle detection and classification based on deep neural network." *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2016.

[14] **Buslaev, Alexander, et al.** "Fully convolutional network for automatic road extraction from satellite imagery." *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2018.

[15] **Li, Jun, et al.** "Deep neural network for structural prediction and lane detection in traffic scene." *IEEE transactions on neural networks and learning systems* 28.3 (2017): 690-703.

[16] **Horita, Luiz Ricardo T., and Valdir Grassi.** "Employing a fully convolutional neural network for road marking detection." *Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), 2017 Latin American*. IEEE, 2017.

[17] **Cisek, Daniel, et al.** "Parking lot delineation and object detection using a localized Convolutional Neural Network." *Scientific Data Summit (NYSDS), 2016 New York*. IEEE, 2016.

[18] **Jan, Zohaib, et al.** "A Convolutional Neural Network Based Deep Learning Technique for Identifying Road Attributes." *2018 International Conference on Image and Vision Computing New Zealand (IVCNZ)*. IEEE, 2018.

[19] **Yadav, Shashank, et al.** "Deep CNN with color lines model for unmarked road segmentation." *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017.

- [20] **Singh, Priya, and Ratnakar Dash.** "A Two-Step Deep Convolution Neural Network for Road Extraction from Aerial Images." *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 2019.
- [21] **Zhang, Zhengxin, Qingjie Liu, and Yunhong Wang.** "Road extraction by deep residual u-net." *IEEE Geoscience and Remote Sensing Letters* 15.5 (2018): 749-753.
- [22] **Proulx-Bourque, Jean-Samuel, and Mathieu Turgeon-Pelchat.** "Toward the Use of Deep Learning for Topographic Feature Extraction from High Resolution Optical Satellite Imagery." *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018.
- [23] **Huang, Xin, and Liangpei Zhang.** "Road centreline extraction from high-resolution imagery based on multiscale structural features and support vector machines." *International Journal of Remote Sensing* 30.8 (2009): 1977-1987.
- [24] **Mnih, Volodymyr, and Geoffrey E. Hinton.** "Learning to detect roads in high-resolution aerial images." *European Conference on Computer Vision*. Springer, Berlin, Heidelberg, 2010.
- [25] **Cheng, Guangliang, et al.** "Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network." *IEEE Transactions on Geoscience and Remote Sensing* 55.6 (2017): 3322-3337.
- [26] **Saito, Shunta, Takayoshi Yamashita, and Yoshimitsu Aoki.** "Multiple object extraction from aerial imagery with convolutional neural networks." *Electronic Imaging* 2016.10 (2016): 1-9.
- [27] **Alshehhi, Rasha, and Prashanth Reddy Marpu.** "Hierarchical graph-based segmentation for extracting road networks from high-resolution satellite images." *ISPRS journal of photogrammetry and remote sensing* 126 (2017): 245-260.
- [28] **Liu, Bo, et al.** "Main road extraction from zy-3 grayscale imagery based on directional mathematical morphology and vgi prior knowledge in urban areas." *PloS one* 10.9 (2015): e0138071.

- [29] **Sujatha, Chinnathevar, and Dharmar Selvathi.** "Connected component-based technique for automatic extraction of road centerline in high resolution satellite images." *EURASIP Journal on Image and Video Processing* 2015.1 (2015): 8.
- [30] **Cheng, Guangliang, et al.** "Road centerline extraction via semisupervised segmentation and multidirection nonmaximum suppression." *IEEE Geoscience and Remote Sensing Letters* 13.4 (2016): 545-549.
- [31] **Song, Mingjun, and Daniel Civco.** "Road extraction using SVM and image segmentation." *Photogrammetric Engineering & Remote Sensing* 70.12 (2004): 1365-1371.
- [32] **LEE, T. S., and MUMFORD, D.,** 2003. "Hierarchical Bayesian inference in the visual cortex." *JOSA A*, 20(7): 1434-1448.
- [33] **BENGIO, Y.,** 2009. "Learning deep architectures for AI. Foundations and trends, in Machine Learning", *Dept. IRO, Universite de Montreal*, 2(1): 1-127.
- [34] **Tesauro, G.,** 1992. "Practical issues in temporal difference learning. Machine Learning", *IBM Thomas J. Watson Research Center*, pp. 8, 257-277.
- [35] **Uzun E.,** 2007. "İnternet Tabanlı Bilgi Erişimi Destekli Bir Otomatik Öğrenme Sistemi", *Doktora Tezi., T.C. Trakya Üniversitesi Fen Bilimleri Enstitüsü.*
- [36] **Öztemel E.,** 2003. "Yapay Sinir Ağları". İstanbul, Türkiye, Papatya Yayıncılık
- [37] **Elmas Ç.,** 2003. "Yapay Sinir Ağları Kuram, Mimari, Eğitim, Uygulama", 1. Baskı, Ankara, Türkiye, Seçkin Yayıncılık.
- [38] **Aslan O., Cheng H., Schuurmans D. And Zhang X.,** 2013. "Convex two-layer modeling", *In Proceedings of Neural Information Processing Systems(NIPS).*
- [39] **Cho Y. and Saul L.,** 2009. "Kernel Methods for Deep Learning", *In Proceedings of Neural Information Processing Systems(NIPS)*, pages 342-350.

- [40] **Deng L., Tur G. and Hakkani-Tur D.,** December 2012 . “Use of Kernel Deep Convex Networks and End-To-End Learning for Spoken Language Understanding”, *In Proceedings of IEEE Workshop on Spoken Language Technologies*.
- [41] **Viyals O., Jia Y., Deng L. and Darrell T.,** 2012 .“Learning with Recursive Perceptual Representations”, *In Proceedings of Neural Information Processing Systems(NIPs)*.
- [42] **Li Deng, Dong Yu,** 2014.“Deep Learning Methods and Applications”, *Foundations and Trends Signal Processing* ,vol.7 ,198-250.
- [43] **George D.,** 2008. “How to Brain Might Work: A Hierarchical and Temporal Model for Learning and Recognition”, *Ph.D. Thesis*, Stanford University.
- [44] **Salakhutdinav R. And Hinton G.,** 2009. “Semantic Hashing”, *Int. J. Approx. Rason.*, vol. 50, pp. 969-978.
- [45] **C.Y. Liou, W.C. Cheng, J.W. Liou, and D.R. Liou,** 2014. “Autoencoder for Words”, *Neurocomputing*, vol. 139, pp. 84-96.
- [46] **A. Krizhevsky and G.E. Hinton,** 2011. “Using Very Deep Autoencoders for Content Based Image Retrivel”, *in European Symposium on Artificial Neural Networks*, pp 489-494.
- [47] **Sebastian B., Stefan J. Kiebel,** 2012. “Recognizing Recurrent Neural Network (rRNN): Bayesian inference for recurrent neural networks”, Cornell University.
- [48] **K. Fukushima,** 1980. “Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”, *Biol. Cybern.* Vol. 36 no. 4, pp. 193-202.
- [49] **D. Cireşan, U. Meier, J. Schmidhuber,** 2012. “Multi-column Deep Neural Networks for Image Classification”, *Computer Vision and Pattern Recognition*, pp 3642-3649.
- [50] **Srivastava N., G. E. Hinton, A.Krizhevsky, I. Sutskever and R. Salakhutdinov,** 2014. “Dropout: a simple way to prevent neural networks fro overfitting”, *Journal of Machine Learning Research* 15(1),1929-1958.

- [51] **Srivastava N., G. E. Hinton, A.Krizhevsky, I. Sutskever and R. Salakhutdinov,** 2012. “Improving Neural Networks by Preventing Co-adaptation of Feature
- [52] **Long J., Shelhamer E. and Darrell T.,** 2014.“Fully Convolutional Networks for Semantic Segmentation” ,*IEEE*.
- [53] **Tsung-Han C., Kui J., Shenghua G., Jiwen L., Zinan Z., Yi M.,** December 2015.“PCANet: A Simple Deep Learning Baseline for Image Classification?”, *IEEE*, 5017-5032.
- [54] **Alex K., Ilya S., and Geoffrey E.Hinton,** 2012. “ImagiNet Classification with Deep Convolutional Neural Network”, Universty of Toronto.
- [55] **Karen S. And Andrew Z.,** 2015. “VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION”, *Conference Paper at ICLR*.

ÖZGEÇMİŞ

Figen ÖNÜN	
KİŞİSEL BİLGİLER	
ADRES :	Fırat Üniversitesi, 23119 – Elazığ
TELEFON :	+90 (5349614981)
EPOSTA :	figenonun@gmail.com
DOĞUM YERİ/TARİHİ:	23.11.1993 / MARDİN
MEDENİ HALİ:	BEKAR
EĞİTİM BİLGİLERİ	
LİSANSÜSTÜ	2015-Devam Ediyor, Fırat Üniversitesi , Fen Bilimleri Enstitüsü, Ekobilişim Anabilim dalı, Elazığ
LİSANS :	2010-2014, Fırat Üniversitesi , Mühendislik Fakültesi , Bilgisayar Mühendisliği, Elazığ
LİSE :	2006-2010, Bismil Anadolu Lisesi, Diyarbakır
İŞ DENEYİMİ	
2017- ...	Eksim Yatırım Holding A.Ş Yazılım Geliştirme Uzmanı
2015- 2017	Yonca CBS Harita ve Mühendislik Şirketi Yazılım Geliştirici
2015-2016	Calleva Yazılım AR-GE Şirketi Yazılım Geliştirici
2014-2015	Fizyokin Bilişim AR-GE Danışmanlık Yazılım Geliştirici ve Firma sahibi
BİLDİĞİ YABANCI DİLLER	
	İngilizce (B2)
ARAŞTIRMA DENEYİMİ	
Programlama Dilleri: C#,ASP.NET, .NET MVC,WCF,Bootstrap,Java, AngularJS, Ionic Framework1, Ionic Framework 2,Angular 2 (Type Script) , Android, Kinect, Windows Phone, Bitbucket, JIRA Veritabanları: Oracle,Sql,MongoDB, Postgresql, MySql	
İLGİLENDİĞİ DİĞER BİLİMLER	