



**EVRIŖİMLİ SİNİR AĞLARI İLE ÇEVİRİMDİŖİ İMZA TANIMA VE
DOĞRULAMA**

Bilal TURAN

**YÜKSEK LİSANS TEZİ
ADLİ BİLİŖİM ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
BİLİŖİM ENSTİTÜSÜ**

EKİM 2019

Bilal TURAN tarafından hazırlanan “EVİRİŞİMLİ SİNİR AĞLARI İLE ÇEVİRİMDİŞİ İMZA TANIMA VE DOĞRULAMA” adlı tez çalışması aşağıdaki jüri tarafından OY ÇOKLUĞU ile Gazi Üniversitesi Bilişim Enstitüsü Adli Bilişim Ana Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Danışman: Doç. Dr. Yusuf SÖNMEZ

Elektrik Üretim-İletim-Dağıtım Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

Başkan: Doç. Dr. Harun ARTUNER

Bilgisayar Mühendisliği Ana Bilim Dalı, Hacettepe Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

Üye: Prof. Dr. Şeref SAĞIROĞLU

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylamıyorum.

Tez Savunma Tarihi: 25.10.2019

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Doç. Dr. Aslıhan TÜFEKÇİ

Bilişim Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Bilişim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Bilal TURAN

25/10/2019

EVRIŞİMLİ SİNİR AĞLARI İLE ÇEVİRİMDİŞİ İMZA TANIMA VE DOĞRULAMA
(Yüksek Lisans Tezi)

Bilal TURAN

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

Ekim 2019

ÖZET

Bu çalışmada evrişimli sinir ağı kullanılarak çevrimdışı imza doğrulaması yapılmıştır. Çalışmada İran kökenli UTSig imza veri tabanı kullanılmış ve yeni bir evrişimli sinir ağı mimarisi önerilmiştir. Önerilen ağı Tensorflow kütüphanesi ve Keras API arayüzü kullanılarak Python programlama dilinde oluşturulmuştur. Çalışmada öğrenme doğruluğunun artırılması amaçlanmış ve bunun için orijinal verilere arttırma, seyreltme ve düzeltme teknikleri uygulanmıştır. Sonuçlar önerilen ağı VGG-16 ve LeNet ağı mimarileri ile karşılaştırıldığında daha iyi sonuçlar verdiğini göstermiştir. Ayrıca orijinal veri setinin arttırılması ve uygulanan diğer teknikler önerilen ağı daha iyi sonuçlar vermesini sağlamıştır. Tezin 1. bölümü giriş bölümü olup, 2. bölümünde derin öğrenme, imza tanıma ve doğrulama üzerine yapılmış çalışmalardan bahsedilmiş, 3. ve 4. bölümlerinde yapay zeka, derin öğrenme üzerine yapılan araştırmalara ve bilgilere yer verilmiş, 5. bölümde ise evrişimli sinir ağları ve bu alanda başarılar elde eden AlexNet, VGG-16, GoogleNet gibi mimariler incelenmiştir. Ayrıca 5. bölümde bu tez çalışmasında oluşturulan hibrid ağı mimarisinin detaylarına yer verilmiştir. Tezin 6. bölümünde uygulama ve deneysel sonuçlar açıklanmıştır. Sonuç ve öneriler 7. bölümde sunulmuştur.

Bilim Kodu : 92401
Anahtar Kelimeler: İmza tanıma, imza doğrulama, derin öğrenme, evrişimli sinir ağı
Sayfa Adedi : 83
Danışman : Doç. Dr. Yusuf SÖNMEZ

OFFLINE SIGNATURE RECOGNITION AND VERIFICATION WITH CONVOLUTIONAL NEURAL NETWORKS

(M.Sc. Thesis)

BİLAL TURAN

GAZI UNIVERSITY
INFORMATICS INSTITUTE

November 2019

ABSTRACT

In this study, offline signature verification has been realized by using Convolutional Neural Network (CNN). Persian Offline Signature Dataset (UTSig) has been used in the implementation and a new CNN structure has been proposed to signature verification. The proposed network was created by Keras API and Tensorflow backend with Python programming language and tested with UTSig open access signature dataset. In the study, increasing the learning accuracy is aimed and in order to achieve this, augmentation, dropout and regularization techniques have been applied to original data. Results obtained from original and augmented data are compared. First results show that the proposed network produces better results comparing to VGG-16 and LeNet. Moreover, augmentation and other techniques on the original dataset conducted to more accurate result for proposed network. The first part of the thesis is the introductory part. In the second part, studies on deep learning, signature recognition and verification are mentioned, in the 3rd and 4th chapters artificial intelligence, researches and information on deep learning are given. AlexNet, VGG-16 and GoogleNet have been successful in this field. In addition, the details of the hybrid network architecture created in this thesis are given in chapter 5. In chapter 6 of the thesis, applications and outputs are given. In Chapter 7, conclusions and recommendations are given.

Science Code : 92401

Key Words : Signature recognition, signature verification, deep learning, cnn

Page Number : 83

Supervisor : Assoc. Prof. Dr. Yusuf SÖNMEZ

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren saygıdeğer hocam Doç. Dr. Yusuf SÖNMEZ'e, kıymetli tecrübelerinden yararlandığım hocam Doç. Dr. Hasan Hüseyin SAYAN'a, bilgisiyle ve yardımlarıyla desteğini hiçbir zaman esirgemeyen değerli arkadaşım Ömer Faruk TEKGÖZOĞLU'na ve her zaman her durumda yanımda olan aileme en içten dileklerimle teşekkür ederim.



İÇİNDEKİLER

	Sayfa
ÖZET.....	iv
ABSTRACT.....	iv
TEŞEKKÜR.....	vi
İÇİNDEKİLER.....	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	x
SİMGELER VE KISALTMALAR.....	xiii
1. GİRİŞ.....	1
2. LİTERATÜR ARAŞTIRMALARI.....	3
2.1. İmza Tanıma ve Doğrulama Alanında Yapılan Çalışmalar.....	3
2.2. İmza Veri Tabanları.....	7
3. YAPAY ZEKÂ.....	11
3.1. Yapay Zekânın Tarihsel Süreci.....	12
4. DERİN ÖĞRENMENİN TANIMI VE GELİŞİMİ.....	17
4.1. Tanım.....	17
4.2. Derin Öğrenmenin Gelişimi.....	18
4.2.1 Hebb kuralı.....	18
4.2.2 McCulloch ve pitts modeli.....	19
4.2.3 Perseptron.....	21
4.2.4 İleri yayılım.....	21
4.2.5 Geriye yayılım.....	25
4.2.6 Aktivasyon fonksiyonları.....	27
4.2.7 Optimizasyon algoritmaları.....	27

	Sayfa
5. EVRİŞİMLİ SİNİR AĞLARI (ESA).....	31
5.1. ESA'nın Başlıca Katmanları.....	31
5.1.1. ESA'da evrişim işlemi.....	32
5.1.2. Ortaklama (pooling) işlemi.....	33
5.1.3. Tamamı bağlı katman.....	34
5.1.4. Dolgulama (padding) işlemi.....	34
5.1.5. Paket normalizasyonu.....	35
5.2. Popüler Evrişimli Sinir Ağı Mimarileri.....	35
5.2.1. LeNet-5.....	35
5.2.2. AlexNet.....	36
5.2.3. VGG-16.....	37
5.2.4. GoogleNet (inception).....	39
5.2.5. Microsoft resnet.....	42
5.3. Çalışmada Kullanılan ve Önerilen Ağ Mimarisi.....	43
6. İMZA VERİ SETİNİN ÖNERİLEN AĞ İÇİN UYGULANMASI.....	53
6.1. Önerilen Ağ Mimarisi ile Diğer Mimarilerin Karşılaştırılması.....	56
6.2. Önerilen Ağ Modelinin Optimize Edilmesi ile En iyi Sonuçlarının Eldesi.....	59
7. SONUÇ VE ÖNERİLER.....	75
KAYNAKLAR.....	77
ÖZGEÇMİŞ.....	83

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. MLP-RBF-CSFNN doğruluk oranları.....	4
Çizelge 2.2. Daş M. ve Dülger C. imza doğrulama çalışması sonucu elde edilen sonuçlar.....	5
Çizelge 4.1. Aktivasyon fonksiyonları.....	27
Çizelge 4.2. Optimizasyon algoritmaları.....	28
Çizelge 6.1. UTSig veri seti kullanılarak önerilen model ile karşılaştırılan mimarilerin doğruluk ve maliyet fonksiyonu değerler.....	59
Çizelge 6.2. Önerilen model için en iyileme algoritmalarının seçiminde doğruluk ve maliyet fonksiyonu değerleri.....	60
Çizelge 6.3. Önerilen modelde öğrenme katsayısının en iyilemesinde doğruluk ve maliyet fonksiyonu değerleri.....	61
Çizelge 6.4. Önerilen modelin yığın boyutu en iyilemesinde doğruluk ve maliyet fonksiyonu ve 1 epok eğitim süresi.....	62
Çizelge 6.5. Önerilen modelin çekirdek başlangıç değerlerinin en iyilemesinde doğruluk ve maliyet fonksiyonu değerleri.....	63
Çizelge 6.6. Önerilen modelin tüm en iyileme değerleri.....	63
Çizelge 6.7. Önerilen modelin 300 epokta normal ve seyreltme - düzeltme eklenmiş hali.....	64
Çizelge 6.8. Önerilen modelde 500 epokta orijinal ve artırılmış veriler ile alınan doğruluk ve maliyet fonksiyonu değerleri.....	70

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. UTSig veri tabanı örnekleri.....	9
Şekil 4.1. Biyolojik sinir hücresi.....	17
Şekil 4.2. Matematiksel sinir hücresi modeli.....	19
Şekil 4.3. Perseptron'un yapısı.....	21
Şekil 4.4. Yapay bir sinir ağı.....	22
Şekil 4.5. İleri yayılım için örnek bir sinir hücresi.....	22
Şekil 4.6. Gizli katmana sahip sinir ağında ileri yayılım.....	24
Şekil 4.7. Yerel ve global minimum noktaları.....	26
Şekil 5.1. Maksimum ortaklama işlemi.....	33
Şekil 5.2. Ortalama ortaklama işlemi.....	34
Şekil 5.3. LeNet-5 ağı mimarisi.....	36
Şekil 5.4. AlexNet ağı mimarisi.....	37
Şekil 5.5. VGG-16 ağı mimarisi.....	38
Şekil 5.6. Inception bloğu mimarisi.....	39
Şekil 5.7. Inception blokları.....	41
Şekil 5.8. ResNet ağı artık bağlantı.....	42
Şekil 5.9. ILSVRC yarışması ağ derinlikleri.....	43
Şekil 5.10. Önerilen ağ mimarisi.....	44
Şekil 5.11. Orijinal inception bloklarının yapısı.....	46
Şekil 5.12. Çalışmada kullanılan inception blok yapıları.....	47
Şekil 5.13. Önerilen ağ mimarisinin akış diyagramı ve uygulama süreci.....	49
Şekil 6.1. UTSig veri seti ile modellerin kullanıcı tahmin doğruluk değerleri.....	53
Şekil 6.2. UTSig veri seti ile modellerin gerçek-sahte imza doğruluk değerleri....	57

Şekil	Sayfa
Şekil 6.3. UTSig veri seti ile modellerin kullanıcı tahmin doğruluk değerleri....	58
Şekil 6.4. 300 epok kullanıcı tahmini doğruluk sonuçları.....	64
Şekil 6.5. 300 epok gerçek-sahte imza ayırımına ilişkin doğruluklar.....	66
Şekil 6.6. Maliyet fonksiyonlarına ilişkin değerler.....	68
Şekil 6.7. 500 epok gerçek-sahte imza tahminine ilişkin doğruluk değerleri.....	72
Şekil 6.8. 500 epok kullanıcı tahminine ilişkin doğruluk değerleri.....	73



SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
Adagrad	Adaptif eğim (Adaptive gradient)
Adam	Adaptif moment tahmini (Adaptive moment estimation)
GPU	Grafik işlem birimi
CPU	Merkezi işlem birimi
EPOK	Ağın eğitimindeki her bir döngü
ESA	Evrişimli sinir ağı
ILSVRC	Büyük ölçekli görsel tanıma yarışması
Nadam	Nesterov hızlandırılmış adaptif moment tahmini
Regularization	Düzeltilme
RGB	Red, Green, Blue (Kırmızı, yeşil, mavi)
RMSprop	Kare ortalamalarının karekökü yayılımı
SGD	Olasılıksal eğim düşümü (Stochastic gradient descent)

1. GİRİŞ

Makine öğrenmesi günümüzde sıklıkla kullanılan, veriden anlam çıkarma ve bir problem için çözüme ulaşma yollarından birisidir [1]. Son yıllarda veri miktarında meydana gelen büyük artış neticesinde makine öğrenmesi teknikleri verimli çalışma özelliğini yitirmiş ve yerini derin öğrenme tekniklerine bırakmıştır [2]. Derin öğrenme tekniklerinden Konvolüsyonel Nöral Ağlar olarak da adlandırılan evrişimli sinir ağları, geleneksel nöral ağların geliştirilmesi ile ortaya çıkmıştır. Bu tür ağlar çok fazla katmandan oluşması sebebiyle derin ağların veya derin öğrenmenin bir türüdür. Evrişimli ağlar, özellikle resim tabanlı sınıflandırmada büyük başarılar elde etmişlerdir [3]. Bu sebeple de şu günlerde oldukça popülerdir. GPU teknolojisinin gelişmesi ile CPU'lara göre çok daha hızlı ve büyük boyutlu verilerle işlem yapılabilmeye başlanmıştır. Yapılan işlemler genellikle bir problemi çözmeye yöneliktir. Problemler güvenlik-bilgi teknolojileri-sağlık teknolojisi-reklam gibi yüzü aşkın alanda çeşitlilik gösterebilir [3].

Bu problemlerden bir tanesi de kimlik tespiti ve doğrulaması olarak da değerlendirebileceğimiz imza tanıma ve doğrulamadır. İmza bir kimsenin, bir yazının altına bu yazıyı yazdığını veya onayladığını belirtmek için; her zaman aynı biçimde yazdığı ad ve işaretidir. İmza kimlik doğrulama sistemlerinden bir tanesidir [4]. Parmak izi, yüz, iris (göz), avuç içi tanıma gibi kimlik doğrulama sistemlerinden farklı olarak imza çok daha pratik ve maliyetsiz bir doğrulama yöntemidir [4].

İmza hukuksal anlamda bireyi sorumluluk altına sokan önemli bir işlemdir [5]. Atılan imzalar ile telafisi olmayan ve istenmeyen hukuki sonuçlar doğabilmektedir. Atılan imzanın gerçek kişiye ait olduğunun doğrulanması da bu nedenle bir o kadar önemlidir. Özellikle hukuki sonuçları doğuran olaylarda bilirkişiler, kimlik doğrulama sistemlerinden olan imza/el yazısı gibi kişinin kendine has olan özniteliklerini incelemektedir [5]. Bu incelemeler doğrultusunda kişi para cezası almaktan, nitelikli dolandırıcılığa kadar uzanan ve sonunda mahkûmiyet riski bulunan durumlara sürüklenebilmektedir [5]. Bir kişinin haksız ithama maruz kalmaması adına bu alanda verilecek olan kararlar bilimsel metotlar ile desteklenmelidir. Özellikle söz konusu olan imza gibi taklidi mümkün bir konuda alınan hukuki kararın, mutlak suretle bir bilimsel dayanağı da olmalıdır. Sadece bilirkişiler tarafından alınan kararlarda, bilirkişinin imzanın sahte/gerçek olduğu kararına varırken sahip olduğu motivasyon, ruh hali, stres durumu gibi duygusal etmenler kararına doğrudan etki edebilmektedir. Oysa bilimsel gerçekler kişiden bağımsızdır. Bu nedenle imza doğrulama

sadece insan gözü ile alınabilecek bir karar değildir. Bu yüzden imza tanıma ve doğrulamanın insan kontrolü dışında bilimsel bir metod ile de desteklenmesi gerekmektedir.

İmza doğrulamada kullanılan yöntemler çevrimiçi ve çevrimdışı olmak üzere ikiye ayrılmaktadır. Çevrimiçi sistemlerde, hız, yön ve basınç bilgilerini ölçüp kaydetmeye yarayan özel ve maliyetli olan dijital ekran, kalem, özel baskı kâğıtlar gerekmektedir. Bu nedenle tercih edilebilecek ucuz bir yöntem değildir [4]. Özellikle kâğıt üzerine atılan eski imzalarda çevrimiçi imza doğrulaması yapılamamaktadır. Çünkü çevrimiçi imza doğrulama yönteminde atılan imzanın daha önceden kayıt edilmiş olan basınç, yön, hız vb. teknik bilgilerine ihtiyaç vardır [4]. Çevrimdışı imza doğrulamada bu kısıtlama olmaksızın eski ve yeni imzalar sahteliği ayırt edilerek doğrulanabilmektedir. Bu nedenle yapılan çalışmaların büyük bir kısmını çevrimdışı imza doğrulama sistemleri oluşturmaktadır.

Tezin Amacı ve Kapsamı

Bu tez çalışmasında çevrimdışı imza doğrulama için evrişimli sinir ağları kullanılmıştır. Öğrenme doğruluğunu artırmak için hibrid bir ağ mimarisi önerilmiştir. İmza doğrulamasını gerçekleştirmek için UTSig veri seti kullanılmıştır. Önerilen ağın performansını test etmek için UTSig veri setinin %20'si kullanılmıştır. Önerilen ağ ile elde edilen sonuçlar VGG-16 ve LeNet mimarilerinin sonuçları ile karşılaştırılmıştır. Bu karşılaştırmalara göre sonuçlar önerilen ağın doğruluğu artırdığını göstermektedir. Geleneksel imza doğrulama sistemlerinin aksine önerilen ağ hangi imzanın hangi kullanıcıya ait olduğunu ve aynı zamanda orijinal veya sahte imza tespiti yapabilmektedir.

Ek olarak; önerilen ağ için test doğruluğu performansı, veri artırma, seyreltme ve düzeltme teknikleri kullanılarak artırılmıştır. Arttırılmış veri setinde, 8280 olan toplam imza görüntü sayısı 12420'ye çıkarılmıştır. Veri artırma işleminde 90 ° 'lik rotasyon ve minimum filtre kullanılmıştır. Görüntülerin minimum filtre ile daha belirgin çizgiler içerdiği görülmüştür.

2. LİTERATÜR ARAŞTIRMALARI

Derin öğrenme alanının gelişmesi eskiden çözülmesi zor olan problemlerin daha kolay şekilde çözülmesini sağlamıştır. Bu problemlerden bir tanesi de kimlik doğrulama alanında kullanılan imzadır. İmza bir kimsenin, bir yazının altına bu yazıyı yazdığını veya onayladığını belirtmek için; her zaman aynı biçimde yazdığı ad ve işarettir [4]. İmza kimlik doğrulama sistemlerinden bir tanesidir. Parmak izi, yüz, iris (göz), avuç içi tanıma gibi kimlik doğrulama sistemlerinden farklı olarak imza çok daha pratik ve maliyetsiz bir doğrulama yöntemidir [4].

İmza hukuksal anlamda bireyi sorumluluk altına sokan önemli bir işlemdir. Atılan imzalar ile telafisi olmayan ve istenmeyen hukuki sonuçlar doğabilmektedir. Atılan imzanın gerçek kişiye ait olduğunun doğrulanması da bu nedenle bir o kadar önemlidir [4].

İmza doğrulamada kullanılan yöntemler çevrimiçi ve çevrimdışı olmak üzere ikiye ayrılmaktadır [4]. Çevrimiçi sistemlerde, hız, yön ve basınç bilgilerini ölçüp kaydetmeye yarayan özel ve maliyetli olan dijital ekran, kalem, özel baskı kâğıtlar gerekmektedir. Bu nedenle tercih edilebilecek ucuz bir yöntem değildir. Özellikle kâğıt üzerine atılan eski imzalarda çevrimiçi imza doğrulaması yapılamamaktadır. Çünkü çevrimiçi imza doğrulama yönteminde atılan imzanın daha önceden kayıt edilmiş olan basınç, yön, hız vb. teknik bilgilerine ihtiyaç vardır. Çevrimdışı imza doğrulamada bu kısıtlama olmaksızın eski ve yeni imzalar sahteliği ayırt edilerek doğrulanabilmektedir. Bu nedenle yapılan çalışmaların büyük bir kısmını çevrimdışı imza doğrulama sistemleri oluşturmaktadır.

Bu çalışmada imza doğrulama yöntemlerinden çevrimdışı imza doğrulama yöntemi üzerinde çalışılmıştır.

2.1. İmza Tanıma ve Doğrulama Alanında Yapılan Çalışmalar

İmza tanıma ve doğrulama alanlarında literatür çalışması yapıldığında bulunan çalışmalar aşağıda verilmiştir.

Şenol ve Yıldırım (2004) yapmış oldukları Standart ve Hibrid Yapılar Kullanarak Yapay Sinir Ağları ile İmza Tanıma tez çalışmasında 8 farklı kişiden 32’şer imza toplamışlardır. Toplam 256 imzadan 200 tanesini imza eğitiminde, 56 tanesini de test amacıyla

kullanmışlardır. Çalışmada Çok Katmanlı Algılayıcı (MLP), Radyal Temelli Fonksiyonlar (RBF) ve Konik Kesit Fonksiyonlu Ağlar (CSFNN) olmak üzere 3 farklı ağ yapısı ile eğitime yapılmış ve sonuçları karşılaştırılmıştır. Buna göre doğruluk sonuçları aşağıdaki gibidir [7].

Çizelge 2.1. MLP-RBF-CSFNN Doğruluk Oranları [8]

Yöntem	Eğitim İmzaları	Test İmzaları
MLP	%87,4	%92,143
RBF	%100,0	%98,21
CSFNN	%96,43	%96,43

Başer ve Şatır (2017) yapmış oldukları Signature Verification Using Artificial Neural Network çalışmasında tıpkı bu çalışmadaki gibi UTSig veri seti kullanmışlardır. UTSig veri setinde 27'si gerçek imza, 42'si sahte imza olmak üzere 115 veri seti bulunmaktadır. Yapılan çalışmada çok katmanlı algılayıcı(MLP) sinir ağı kullanılarak %84,4 başarı elde edilmiştir [8].

Erten (2012) Sınıflandırma Yöntemleri Kullanılarak İmza Biyometriğine Dayalı Kişi Tanıma adlı tez çalışmasında eğitim amaçlı 400 örnek imza, test amacıyla 80 imza olmak üzere toplam 480 imza kullanılmıştır. Erten M.'nin tez çalışmasında Bayes Sınıflandırma Algoritması, En Yakın Komşu Algoritması ve K en Yakın Komşu Algoritması olmak üzere 3 farklı algoritma kullanılmıştır. Yapılan çalışma tanıma sistemi bulunup doğrulama sistemi bulunmamaktadır. Bu çalışmalarda sırasıyla %31,75; %3,75; %1,25 FRR (Yanlış Reddetme Oranı) elde edilmiştir [9].

Baykal, Aktaş, Yıldız (2017) Makine Öğrenmesi Yöntemi ile Otomatik Çevrimdışı İmza Tanıma ve Doğrulama Sistemi adlı çalışmalarında 9 kişiden 20 orijinal, 10 sahte imza toplamışlardır. Toplam 270 imzanın toplandığı çalışmada k En Yakın Komşu (kNN) algoritması ile beraber, YSA, radyal tabanlı fonksiyon (RTF), DVM denenmiştir. Ön işleminden geçirilen her bir imzadan Hu'nun değişmez momentler yöntemiyle elde edilen 7 öznitelik, bölgesel özellikler ile elde edilen 3 öznitelik ve ADD ile elde edilen 13 öznitelik imza doğrulama için kullanılmıştır. Toplamda 23 öznitelik kullanılarak DVM ile %100 doğrulukta imza doğrulama işlemi gerçekleştirmişlerdir [10].

Kaymaz (2010) yapmış olduğu tez çalışmasında GPDS300 Signature imza veri tabanından alınan 50 kişiye ait 1200 gerçek, 1500 sahte imza görüntüsü kullanılan tez çalışmasında imza

görüntülerini temsil eden matrisler arasında Dinamik Zaman Bükmesi mesafeleri hesaplanarak, önceden belirlenen bir eşik değerine göre karşılaştırmalar yapılmıştır [11].

Daş ve Dülger (2007) çalışmalarında imza doğrulaması için temeli yapay sinir ağına dayalı parçacık sürü optimizasyonu ile eğitim aşamasını içeren yeni bir teknik ortaya koymuşlardır. Çalışmada verilen algoritmanın denenmesi için rasgele, yeteneksiz ve yetenekli olmak üzere üç çeşit teknik imza kullanılmış ve elde edilen sonuçlar Çizelge 2.2’de yer aldığı şekilde verilmiştir [12].

Çizelge 2.2. Daş M. ve Dülger C. İmza doğrulama çalışması sonucu elde edilen sonuçlar

	Toplam	Kabul Edilen	Red Edilen
Rastgele seçilen sahte imzalar	18	4 (%22)	14 (%78)
Yetenekli imzalar	20	6 (%60)	4 (%40)
Sahte imzalar	28	21 (%75)	7 (%25)

Bu alanda yurt dışında yapılan çalışmalar aşağıdaki gibi verilmiştir.

Fazli, Pouyan ve Moghaddam (2014) gerçekleştirdiği “High-Performance Signature Recognition Method using Neural Network” adlı çalışmalarında imza tanıma sisteminde, 8 kişiden 20’şer imza örneği alarak toplam 160 imza görüntüsü elde edilmiştir. İmzalardan elde edilen öznitelikler ile DVM algoritması %95,625; k - En Yakın Komşu Algoritması %79,375 ve YSA ile de %91,25 başarı elde edilmiştir [13].

Randhawa ve Sharma (2012) tarafından imza doğrulama üzerine yapılan çalışmada Hu’nun değişmez moment yöntemleri ve zone öznitelik çıkarma kullanılmıştır. 10 kullanıcıdan toplanan imzalar için DVM öğrenme algoritması ile %90 doğruluğa erişilmiştir [14].

Radhika ve Gopika (2015) tarafından çevrimiçi ve çevrimdışı imza doğrulama üzerine bir çalışma gerçekleştirilmiştir. Gradyan ve projeksiyon özellikleri kullanılarak öznitelik çıkarma adımı gerçekleştirilirken, DVM algoritması kullanılarak çevrimdışı imza doğrulama için %74,04 doğruluk oranına ulaşılmıştır [15].

Kumar, Singh, Garg ve Prabhat (2013) YSA kullanarak imza tanıma ve doğrulama sistemi geliştirmişlerdir. Bu çalışmada öznitelik çıkarma işleminden önce imzalar işlenmiş, ardından işlenen görüntüler üzerinde Hu’nun değişmez moment yöntemi uygulanarak öznitelikler elde edilmiştir. Doğrulama aşamasında, test imzalarında önileme ve öznitelik çıkarma işlemleri yapılmıştır. Elde edilen öznitelikler orijinal veya sahte imza olarak sınıflandıran

eğitilmiş bir sinir ağı modeline uygulanmıştır. Sistemin imza doğrulama için doğru sınıflandırma oranı %82,66; imza tanıma için başarı oranı ise %100 olarak bulunmuştur [16].

Al-Omari, Sheikh ve Omar (2011) çevrimdışı imza tanıma ve doğrulama üzerine çalışma yapmışlardır. Çevrimdışı imza doğrulama yöntemlerinde kullanılan özellik çıkarım yöntemleri; istatistiksel özellikler, geometrik özellikler ve doğrulama işleminde kullanılan; şablon eşleştirme yöntemi, yapay sinir ağları yöntemi, Dalgacık dönüşümü, istatistiksel yaklaşımlar, Gizli markov modeli yöntemlerinden bahsedilmiştir [17].

Kisku, Gupta ve Sing (2010) çevrimdışı imza doğrulama için yapmış oldukları çalışmada destek vektör makine algoritması (SVM) kullanılmıştır. İmza görüntüleri küresel ve yerel özellikleri ayıklanmıştır ve imzalar gauss, öklid ve mahalnobis mesafeye dayalı sınıflandırıcılar yardımı ile doğrulanmıştır. SVM algoritması eşleşen puanları birleştirmek için kullanılmıştır [18].

Wai ve Aung (2014) tarafından gerçekleştirilen çevrimdışı imza doğrulama çalışması yapılmıştır. Çalışmada piksel yoğunluğu, hücre açısı, piksel açısı ve piksel varlık öznitelikleri kullanılmıştır. İmza doğrulama için Öklid uzaklık Medyan değerler ve eşikleme teknikleri kullanılmıştır. Çalışmada %98,52 doğruluk oranına erişilmiştir [19].

Kudlacik ve Porwik (2014) tarafından bulanık metot kullanılarak gerçekleştirilen imza tanıma sisteminde 40 kullanıcıdan 20 orijinal ve 20 sahte imza alınmıştır. İmzalar ön işlemden geçirilip, öznitelikleri çıkarıldıktan sonra uygulanan bulanık yöntem ile %99,19 doğruluk oranı elde edilmiştir [20].

Derin öğrenme ile imza doğrulamaya yönelik yapılan çalışmalarda ise UTSig veri tabanı kullanılan derin öğrenme yöntemleri ile yapılan bir çalışmaya rastlanılmamıştır. LeCun, Bottou, Bengio ve Haffner (1998) yaptıkları çalışmada Nist veritabanlarından bazılarını kullanmışlardır. Bu çalışmada LeNet-5 mimarisinde imza doğrulama çalışması yapılmıştır [21].

Khalajzadeh (2012) tarafından yapılan çalışmada, 22 farklı kişiden toplanan imzalar LeNet-5 ağ mimarisinde sınıflandırma yapılmıştır. Çalışma sadece doğru imzaları içeren ve gerçek imzaların birbirinden farklarını sınıflandırmaktadır. Çalışmada sahte imza girdisi bulunmamaktadır [22].

Hafemann (2017) derin evrişimli sinir ağları ile yaptığı çalışmasında gerçek imzaların özniteliklerin öğrenilmesinden sonrasında SVM algoritması ile sınıflandırma yapmaktadır.

Hafemann bu alandaki en iyi sonuçlara ulaşmış bir çalışma yapmıştır. SVM ile ulaşılan %1,72 hata oranı, literatürdeki %6,97 ile karşılaştırıldığında alanının en iyi çalışmasıdır denilebilir [23].

2.2. İmza Veri Tabanları

İmza doğrulamaya yönelik çalışmalarda çalışmayı yapanların en zorlandıkları kısımlardan biri imza örneklerini toplama noktasındadır. İmza kişiye özel ve hukuki sonuçlar doğurabilecek bir kimlik doğrulama sistemi olduğundan kişiler imzalarını verme noktasında mesafeli davranmaktadırlar. Literatürde tüm kısıtlamalara rağmen ulaşılabilen imza veri tabanları yer almaktadır. Bunlardan bazıları şunlardır; MCYT-75 [24], GPDSignature [25], GPDS-960 [26], ICDAR 2009 [27], 4NSigComp2010 [28], SigComp2011 [29], 4NSigComp2012 [30], SigWiComp2013 [31], UTSig [32].

MCYT-75, her biri 15 doğru imza, 15 sahte imza içeren 75 sınıfın olduğu bir veri setidir. Sahte imzalar üç kişi tarafından oluşturulmuştur [24].

GPDSignature, her biri 24 doğru imza, 10 sahte imza içeren 160 sınıfın bulunduğu bir veri setidir. Sahte imzalar 10 farklı kişi tarafından oluşturulmuştur. Doğru imzaların tamamı aynı zaman diliminde alınmıştır. İmzanın farklı zamanlarda alınması önemlidir çünkü her ne kadar kişinin imzasında çok büyük bir değişiklik meydana gelmese de belirli zaman aralıklarında imzada çeşitli farklılıklar olabilmektedir [25].

GPDS-960, her biri 24 doğru, 30 sahte imza ve 960 sınıf içeren bir veri setidir. Doğru imzalar aynı gün içerisinde farklı boyuttaki iki kutucuk içine sığacak şekilde alınmıştır. Toplamda, doğru imzaları oluşturan kişiler dışındaki kişilerden 1920 farklı sahte imza oluşturulmuştur. Sahte imza oluşturan her birey 5 farklı doğru imzadan 1'er adet olmak üzere 5 tane sahte imza atmıştır [26].

ICDAR 2009, aslında bir yarışmanın veri seti olup eğitim ve test veri seti içermektedir. İçerisinde 12 sınıf bulunmaktadır ve her sınıf 5 doğru, 5 sahte imza içermektedir. Sahte imzalar 31 kişi tarafından oluşturulmuştur. Test seti NFI (Netherlands Forensic Institute) tarafından sağlanmıştır [27].

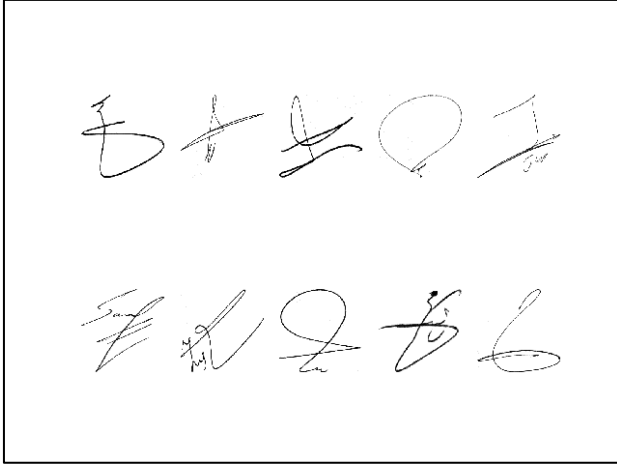
4NSigComp2010, veri setinde 76 doğru, 104 sahte imza olmak üzere 180 adet imzanın yanı sıra, 20 adet doğru-sahte ayrımı yapılmayan imza bulunmaktadır. Toplamda 200 adet imza yer almaktadır [28].

SigComp2011, Çin/Hollanda olmak üzere iki farklı coğrafyanın imza tarzını içeren bir veri setidir. Çin menşei eğitim seti 235 doğru, 340 sahte imza içermektedir. Sahte imzalar 10 farklı kişi tarafından oluşturulmuştur. Değerlendirme seti 116 doğru, 367 sahte, 120 tane doğru/sahte ayrımı yapılmayan imza içermektedir. Hollanda menşei olan eğitim veri seti 240 doğru, 123 sahte imza içermektedir. Sahte imzalar 10 farklı kişi tarafından oluşturulmuştur. Değerlendirme seti 648 doğru, 638 sahte, 648 doğru/sahte ayrımı yapılmayan imza içermektedir [29].

4NSigComp2012, 3 özgün yazar tarafından oluşturulan 15 ile 20 arasında referans imza, 100 ile 250 arasında doğru/sahte ayrımı yapılmayan imza içeren sorgu setinden oluşmaktadır. Sorgu setinde 20-50 arasında doğru imza, 8-47 arasında doğru/sahte imza ayrımı yapılmayan gizlenmiş imza, 42-160 arasında ise sahte imza bulunmaktadır [30].

SigWiComp2013, hem Hollanda kökenli hem de Japon kökenli imzalar içermektedir. Japon kökenli veri seti çevrimiçi imzalardan dönüştürülerek elde edilmiştir. 30 sınıflıdır, her sınıf 42 doğru imza içermektedir. Doğru imzalar 4 günlük süre diliminde alınmıştır. Değerlendirme seti 30 sınıfa ait 36 adet sahte imza 4 farklı kişi tarafından oluşturulmuştur. Hollanda kökenli veri seti 27 kişi tarafından 5 günlük bir zaman diliminde alınmıştır. Her bir kişi 10 adet imza atmıştır. Sahte imzalar 9 kişi tarafından eğitim setindeki sınıfların her birinden ya hiç sahtesini üretmeden ya da tüm örneklerin sahtesini üreterek oluşturulmuştur. Ortalama olarak her sınıf 36 adet sahte imza içermektedir [31].

UTSig veri seti 115 sınıftan oluşan, 8280 adet imza görüntüsü içermektedir. Her sınıf bir kişiye ait 27 doğru imza ve 45 sahte imza içermektedir. Diğer açık kaynaklı veri setlerine kıyasla UTSig sınıflara ait daha fazla sayıda veri içermektedir [32].



Şekil 2.1. UTSig veri tabanı örnekleri [32]

Bu tez çalışmasında İran kökenli UTSig veri tabanı kullanılmıştır. Erişmesi diğer veri setlerine göre daha kolay olan, sınıflara ait daha fazla sayıda veri içeren ve daha fazla sayıda sahte imza içeriği üreten kişiye sahip bir veri tabanı olduğu için tercih edilmiştir.



3. YAPAY ZEKÂ

Sözlük anlamına baktığımızda henüz net bir tanımla karşılaşmadığımız yapay zekâ; bilgisayar ve akıllı sistemlerin tıpkı insanlar gibi algılama, düşünme, mantık yürütme, kavrama, yargılama ve sonuç çıkararak aksiyon alma yeteneklerinin kullanabilir hale gelmesi olarak tanımlanabilir [33]. Yapay zekâ adı dünya literatürüne 1956 yılında Dartmouth'da John McCarthy tarafından önerilerek girmiştir [33]. Yapay zeka ile ilgili düşünme ve muhakeme süreci ile davranışsal olmak üzere iki tür tanımlamanın yapıldığını görebiliriz. Bu tanımları aşağıdaki şekilde verebiliriz.

Düşünme ve muhakeme süreci ile ilgili olarak; “Bilgisayarları düşünmek için yeni heyecan verici çaba... Tam ve gerçek anlamda zihinleri olan makineler yapmak.” (Haugeland, 1985) ve “İnsan düşüncesi ile ilişkilendirdiğimiz faaliyetlerin otomasyonu; karar verme, problem çözme, öğrenme...” (Bellman, 1978) şeklinde tanımlar yapılırken bu tanımlarda başarımların insan performansındaki doğruluk ile kıyaslanmıştır [34].

Düşünme ve muhakeme süreci ile ilgili olarak ayrıca;

“İşlemsel modellerin kullanımı sayesinde zihin becerisi üzerindeki çalışmalardır.” (Charniak ve McDermott, 1985) ve “Algılama, akıl yürütme ve davranmayı mümkün kılan hesaplamalardır.” (Winston, 1992) şeklinde tanımlar da yapılırken bu tanımlarda başarımın ideal performans ile kıyas edilerek rasyonel faydaya yönelik olması öne çıkarılmıştır [34].

Yapay zekâ tanımlarından davranışsallığı ön plana çıkaran tanımlar ise şöyledir;

“İnsanlar tarafından gerçekleştirildiğinde zekâ gerektiren işlevleri yerine getiren makineler üretme sanatıdır.” (Kurzweil, 1990) ve “Bilgisayarların, şu anda insanların daha iyi olduğu şeyleri nasıl yapacağı üzerine bir çalışmadır.” (Rich ve Knight, 1991)

Aşağıdaki tanımlarda da yine davranışsallık ön plana çıkarken başarımların ideal performans ile kıyaslanarak rasyonel fayda gözetilmiştir.

“Yapay zekâ, akıllı ajanların tasarımının incelenmesidir.” (Poole ve ark. 1998) ve “Yapay zekâ... Yapay dokulardaki akıllı davranışlarla ilgilidir.” [34].

Farklı yapay zekâ hedefleri tarihsel süreç içerisinde uygulanmıştır. Beklendiği gibi insan merkezli ve rasyonel merkezli olmak üzere iki kutup oluşmuştur. İnsan merkezli yaklaşımlar deneysel olmak zorundadır ve hipotez ile deneysel doğrulama döngüsü içerisinde gelişmektedir. Rasyonel yaklaşım ise mühendislik ve matematik bilimi birleşimi ile gelişim

göstermektedir. Her iki yaklaşımı kullanan uzmanlar bir diğerine de atıfta bulunmuştur. Bu nedenle her iki yaklaşımda kabul görmüştür [34].

3.1. Yapay Zekânın Tarihsel Süreci

Yapay zekânın ortaya çıkması oldukça eski dönemlerde başlamıştır.

1308 yılında ilk kez Katalan şair ve teolog *Ramon Llull*, kavramların bir araya getirilmesiyle oluşan yeni bir bilgi türünden bahsetmiş ve ilk kez yapay zekânın kavramlaştırılması *Llull* tarafından yapılmıştır. Daha sonraki yüz yıllarda düşünürler yapay zekânın kavramsal çerçevesini geliştirmişlerdir [35].

1854 yılında gelindiğinde *George Boole*, mantıksal akıl yürütmenin denklem çözmeye benzer şekilde sistematik bir şekilde düşünülebileceği düşüncesini ortaya atmıştır. Böylece akademik anlamda akıl yürütmenin bir denklem ile ifade edilip edilemeyeceği yönünde fikirler oluşmaya başlamıştır [35].

1898 yılında *Nikola Tesla*, bir gösteride dünyanın ilk radyo dalgalarıyla kontrol edilen gemisini tanıtmıştır. Bu gemi *Tesla*'nın deyimiyle, “Ödünç alınmış bilinç.” tir [35].

1914 yılında İspanyol mühendis *Leonardo Torres*, ilk satranç oynayabilen makineyi tanıtmıştır, makine hiçbir insan etkisi olmadan oyunu kendi kendine sürdürebilmiştir [35].

1925 yılında *Hundai*, radyo kontrol ile sürücüsüz uzaktan kontrol edilen araçları New York'ta piyasaya çıkarmıştır [35].

1928 yılında Japonya'da Gakutensoku adında ilk robot yapılmıştır. Bu robot 3,20 metre yüksekliğindeydi. Robot yüz ifadesini değiştirebilmek, başını ve ellerini hava basınç mekanizması sayesinde hareket ettirebilmek gibi özelliklere sahipti. Robot Japonca ve Çince yazabilmekteydi.

1943 yılına gelindiğinde ise bir devrim niteliği taşıyan *Warren S. McCulloch* ve *Walter Pitts*, “Sinir Sisteminin İçinde Olan Fikirlerin Mantıksal Hesabı” adlı yayını çıkarmışlardır. İdealleştirilmiş ve sadeleştirilmiş yapay sinir ağlarının ve bunların mantıksal işlevlerinin nasıl gerçekleşebildiklerini anlatan bu etkili makale, bilgisayar temelli sinir ağları ve daha sonrası bu kavramdan gelen derin öğrenmenin çıkış noktası olmuştur [35].

1949 yılında iki önemli gelişme meydana gelmiştir. *Edmund Berkeley*, “Dev Brains” (Düşünen Makineler) adlı eserinde şöyle bir ifadeye yer vermiştir: “Son zamanlarda, devasa

hız ve beceri ile bilgi toplayabilen büyük makineler hakkında çok sayıda haber var. Bu makineler bir beyininkine benzeyen karmaşık yapılardan oluşuyor. Bir makine bilgi işleyebilir, hesaplayabilir, sonuçlandırabilir ve seçebilir; makul işlemleri bilgi ile gerçekleştirebilir. Bu nedenle bir makine düşünebilir.” Makinelerin düşünebileceğine dair söylem kırıntıları ilk defa bu denli büyük şekilde dile getirilmiştir. İkinci önemli olay ise *Donald Hebb* kuralıdır. *Hebb*, “Nöropsikoloji Teorisi” adlı makaleyi yayınlamıştır. Bu teori sinapsların zamanla güçlenmesine veya zayıflamasına ilişkin yeteneklerini içeren bir teoridir. İlerleyen yıllardaki derin öğrenme gelişiminde, *Hebb* kuralı temel alınmıştır ve ağırlık kavramı türetilmiştir [35].

1950 yılında *Alan Turing*, daha sonra bilinen adıyla “Turing Testi” adlı yapay zekâ test sistemini duyurmuştur [35].

1951 yılında *Marvin Minsky* ve *Dean Edmonds* yapay sinir ağı oluşturarak, kırk nörondan oluşan ağı taklit etmek için üç bin vakum tüpü kullanan Stokastik Sinirsel Analog Güçlendirme Hesaplayıcısı (SNARC)’ı ortaya çıkarmıştır [35].

1955 yılında “Yapay Zekâ” terimi, *John McCarty*, *Marvin Minsky*, *Nathaniel Rochester* ve *Claude Shannon* tarafından sunulan “Yapay zekânın iki aylık, on kişilik araştırması” adlı bir sunumda ilk kez kullanılmıştır. Aynı yıl içerisinde *Herbert Simon* ve *Allen Newell* *Whitehead*, Russel *Principia Mathematica*’daki 52 teoremin 38’ini ispatlayacak olan ilk yapay zekâ programı olan Logic Theorist’i geliştirmiştir [35].

1957 yılında *Frank Rosenblatt*, iki katmanlı bir bilgisayar öğrenme ağına dayanan Perceptron’u geliştirmiştir. New York Times, Perceptron’un yürüyebileceğini, konuşabileceğini, görebileceğini, yazabileceğini, üretebileceğini ve varlığının bilincinde olması gereken bir elektronik bilgisayarın embriyosu olabileceğini duyurmuştur [35].

1958 yılında *McCarthy* yapay zekâ araştırmasında çok popüler bir programlama dili olan Lisp’i geliştirmiştir.

1959 yılında *Arthur Samuel*, “Machine Learning – Makine Öğrenmesi” terimini ortaya atmış ve açıklamak için “Program yazan kişinin oynayabileceğinden daha iyi dama oyunu oynamayı öğrenecek.” demiştir.

1961 yılında ilk yapay zekâ kullanan endüstriyel robot olan Unimate, New Jersey’deki General Motors fabrikasında bir montaj hattı üzerinde çalışmaya başlamıştır. Böylece yeni bir sosyal çalışma alanının da doğuşuna şahit olunmuştur. İnsanlar ve yapay zekâ arasında

günümüzde de devam eden yapay zekânın insan iş gücü gerektiren mesleklerde insanın yerini alabileceği tartışmalarının fitilini ateşlemiştir.

1964 ile 1966 yılları arasında MIT Yapay Zekâ Laboratuvarı'nda ELIZA adlı bir ilkel doğal dil işleme bilgisayarı *Joseph Weizenbaum* tarafından geliştirilmiştir. ELIZA insan – makine konuşmalarını şablon eşleme yöntemi ile anlamlandırmaya çalışmaktaydı. Karşısındaki ile sohbet edebiliyor gibi görünmesine karşın, yalnızca karşısındaki insanın cümleleri üzerinde bazı işlemler yapıyordu. İlk makine çevirisi çalışmaları sırasında benzeri yaklaşımlar kullanılıp çok gülünç çevirilerle karşılaşılınca bu çalışmaların desteklenmesi durdurulmuştur [35].

1966'da Shakey adlı robot kendi hareketlerinin sorumluluğunu üstlenen ilk robot olma unvanına erişmiştir. *Marvin Minsky*, Shakey'i "İlk elektronik kişi" olarak nitelendirmiştir. *Minsky* aynı zamanda hızlı gelişen bu süreçlerin cazibesine kapılarak LIFE dergisinin 1970'deki makalesine "Üç aydan sekiz yıla kadar, ortalama bir insanın genel zekâsına sahip makinelere sahip olacağız." demiştir [35].

Ancak sonrasında yaklaşık 30 yıl bu alanda çok fazla çalışma yapılmamış ve bir duraklama dönemi yaşanmıştır.

Gartner'ın beklenti eğrisine göre yapay zekâ er ya da geç potansiyel verimliliğine ulaşacaktır [36]. Nitekim yapay zekâ ilk duraklamasından 1997 yılında IBM'in "Deep Blue" adlı satranç oynayan yapay zekâsı ile çıkmıştır. Deep Blue, dönemin en iyi satranç ustası olan *Garry Kasparov*'u yenerek yapay zekâyı tekrar gündeme getirmiştir.

1998 yılında MIT'den *Cynthia Breazeal*, KISMET adlı robotu duyurmuştur. KISMET insanlar ile etkileşim kuruyor, onların hareketlerini taklit ediyordu. İşitsel, görsel ve ifade sistemleri konusunda çok özenli çalışmaların sonunda resmen bir yapay zekâ sosyal ilişkiler kurarak dünyanın "ilk sosyal robotu" unvanı ile ödüllendirilmiştir [35].

1999 yılında Sony tarafından AIBO piyasaya sürülmüştür. İlk kez kullanıcıya sunulan robot köpek olma özelliği taşımaktadır. Becerileri sonraki yıllarda geliştirilerek artırılmıştır.

2002 yılında Roomba, ilk büyük çaplı üretime sahip olan otonom robotik temizleyici olarak çıkarılmıştır. Robot kendiliğinden ev içerisinde dolaşarak temizlik yapmaktadır. Devam eden yıllarda robotun özellikleri geliştirilmiştir [35].

2011 yılında Apple firması Siri adlı yapay zekâ telefon asistanını duyurmuştur. Siri tüm dünyada ses ile kontrol edilebilen asistan çağının da ilklerinden biri olmuştur.

Yine 2011 yılı içerisinde IBM, Watson isimli soruları cevaplayan bir bot geliştirmiştir. Watson “Jeopardy” isimli TV programında \$ 1 Milyon dolarlık soru yarışmasında birinci olmuştur [35].

2014 yılında, 1950’de Turing Testi olarak bilinen test ilk kez bir yapay zekâ tarafından geçilmiştir. Testi geçen yapay zekânın adı *Eugene Goostman*’dır. Deneye katılan gerçek insanlar, *Eugene*’nin gerçek bir insan olduğu görüşüne varmışlardır. Aynı yıl içerisinde çok bilinen çevrimiçi alışveriş mağazası Amazon, Amazon-Alexa adlı bir çevrimiçi sanal asistan piyasaya çıkarmıştır [35].

Microsoft yapay zekâ alanındaki sessizliğini ise 2016 yılında bozmuş ve Tay isimli bir sohbet botu geliştirerek “Twitter” adlı platformda bir kullanıcı hesabıyla çıkarmıştır. Tay kullanıcıların verilerinden kendi kendine çıkarımlarda bulunarak kendini geliştirmiştir. Çıkarımları gittikçe ırkçı mesajlara dönüştüğü için insanların tepkisini çekmiştir ve Microsoft sohbet botunu kaldırmak zorunda kalmıştır [35].

2017 yılına gelindiğinde Google tarafından geliştirilen AlphaGo isimli yapay zekâ Dünya Go şampiyonu olan *Ke Jie*’yi yenmiştir. Go oyunu bilgisayarlar için oldukça karmaşık seviyede ihtimal içeren bir oyundur. Oyunda yaklaşık olarak 2^{170} olası hamle bulunmaktadır.

AlphaGo ile karşılaşan *Ke Jie* için DeepMind şirketi CEO’su *Demis Hassabis*, *Ke Jie*’ye bu denli güzel bir oyun oynadığı ve AlphaGo’yu limitlerine kadar zorladığı için büyük saygı duyulmasının gerekliliğine değinmiştir. Sonuç olarak AlphaGo *Ke Jie*’yi yenerek kendisini kanıtlamıştır [35].



4. DERİN ÖĞRENMENİN TANIMI VE GELİŞİMİ

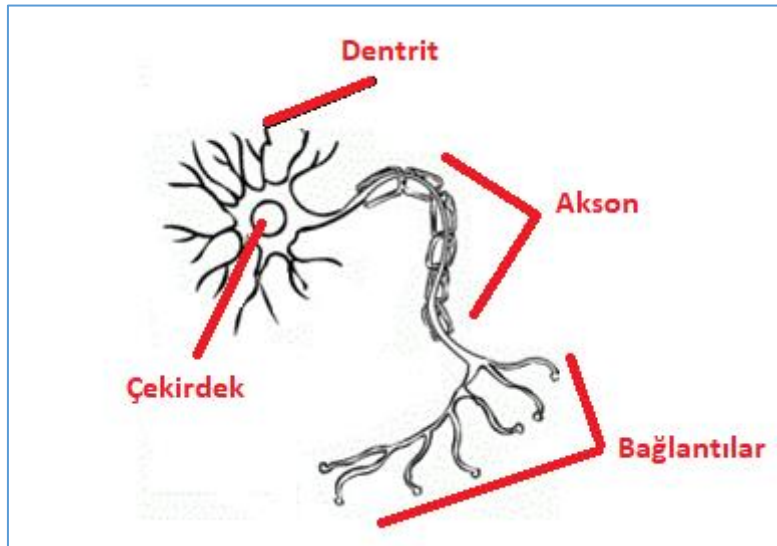
Derin öğrenme zaman içinde gelişim göstermiş, günümüze kadar birçok model ve kural ortaya konulmuştur. Derin öğrenmenin tanımı ve gelişimine dair bilgiler aşağıdaki gibi verilmiştir.

4.1. Tanım

Derin öğrenme konusunda birçok tanım yapılmıştır. Bu tanımlar ortak bir noktada birleştirilecek olursa; insan beyninin karmaşık problemler için gözlem yapma, öğrenme, analiz yapma ve karar verip aksiyon alma gibi yeteneklerini taklit eden, denetimli veya denetimsiz olarak özellik çıkarma, dönüştürme ve sınıflandırma gibi işlemleri çok büyük miktarda veri içerisinde çıkararak bir makine öğrenmesi tekniğidir denilebilir.

Derin öğrenmede çok sayıda katmandan oluşan ve çeşitli dönüşümlerle verilerin anlamlandırıldığı yöntemler topluluğu vardır.

Biyolojik bir hücrede sinir ağı; bir gövde, bir akson, çok sayıda sinir ucu (dentrit) ve akson ile diğer sinir hücresinin sinir ucu arasında kalan ince uzantılar olmak üzere dört ana bölümden oluşur [37].



Şekil 4.1. Biyolojik sinir hücresi [37]

Yapay sinir, biyolojik sinirin matematiksel karşılığının modellenmesi yada kısaca taklit edilmesi ile oluşturulmuştur.

Biyolojik sinir hücreleri ile donatılmış beynimiz oldukça karmaşık ve güçlü bir yapıdadır. Beyni, basit küçük bloklar yardımıyla daha kolay anlaşılır hale getirmek için sinir hücreleri modellenmektedir. İnsan beyninde ortalama 10^{11} adet sinir hücresi olduğu düşünülmektedir. Her hücre kendine özgün görevlerde aktivasyon üretmekte ve diğer sinir hücrelerine yaklaşık 10^{14} adet sinaps ile bağlantı sağlamaktadır. İnsan beyninin karmaşık yapısı paralel olarak çalışan 10^{11} tane bilgisayar işlemcisi gibi düşünülebilir. Sinir hücreleri arasında yer alan bağlantılar ise zamanla güçlenmekte veya zayıflamaktadır. Bazı durumlarda yeni bağlantılar eklenirken bazı durumlarda ise mevcut bağlantılar unutulmaktadır [37].

4.2. Derin Öğrenmenin Gelişimi

Derin öğrenme yöntemlerinin gelişmesi yakın geçmiş içinde olmuştur. Tarihsel gelişim içinde derin öğrenmenin aşamaları aşağıdaki gibi ilerlemiştir.

Sinir hücreleri arasında yer alan bağlantıların zamanla güçlenmekte veya zayıflamakta olduğunu ayrıca bazı durumlarda yeni bağlantılar eklenirken bazı durumlarda ise mevcut bağlantılar unutulmakta olduğunu söylemiştik.

Bağlantıların zamanla değişmesine dair ortaya atılan ilk çalışma *Donald Hebb* tarafından 1949 yılında yapılmıştır.

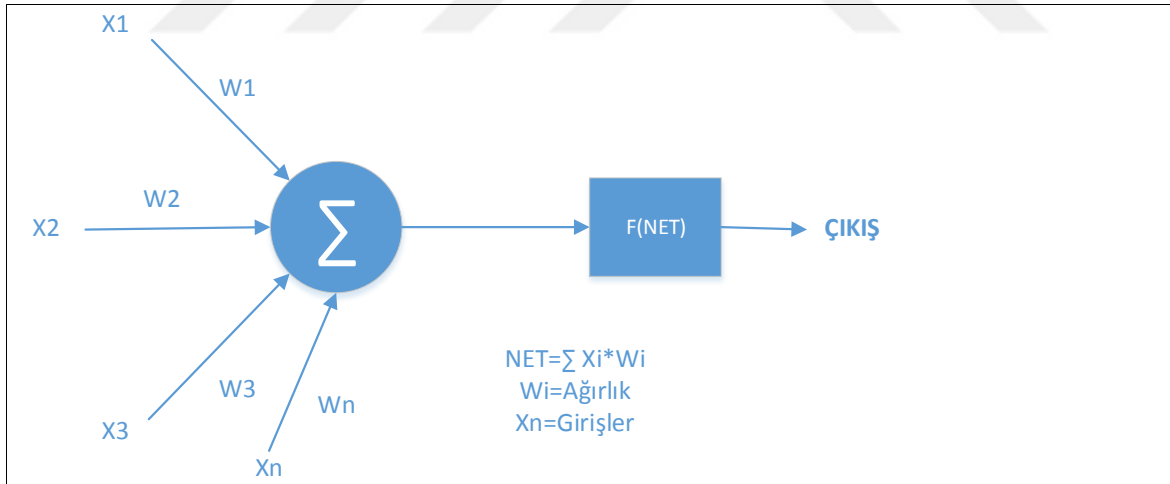
4.2.1 Hebb kuralı

Hebb kuralı iki sinir hücresi arasındaki bağlantının kullanımına bağlı olarak bağlantı kuvvetinin yani ağırlığının artması veya azalması olarak açıklanmaktadır. Yani iki sinir hücresi arasındaki bağlantı sıklıkla kullanılıyor ise bağlantı güçlenir ve ağırlık kat sayısı artar. Ancak sinir ağları arasındaki bağlantı hiç kullanılmıyor veya çok az kullanılıyorsa bağlantı gücü azalır yani ağırlık kat sayıları azalır. Bu durum *Hebb* kuralına dayanarak açıklanmaktadır [37].

Hebb kuralı ile yapay sinir hücresi ağırlıklarının değişebileceği ortaya çıkmıştır. Ancak *Hebb* kuralı ile ortaya çıkan sinir hücresi modeli ilk değildir. İlk sinir hücresi modeli, *Warren McCulloch* ve *Walter Pitts* tarafından 1943 yılında ortaya çıkarılmıştır [37].

4.2.2. McCulloch ve Pitts modeli

İlk matematiksel yapay sinir modeli *Warren McCulloch* ve *Walter Pitts* modeli olarak tarafından ortaya çıkarılmış ve kabul görmüştür. Bu model 1943 yılında “Sinir Sisteminin İçinde Olan Fikirlerin Mantıksal Hesabı” adlı makaleyle duyurulmuştur. Model aynı zamanda doğrusal eşik kapısı olarak da bilinmektedir. Bu şekilde anılmasının asıl nedeni adım fonksiyonu kullanmasıdır. Adım fonksiyonunda ara değerler için belirli bir eşik değeri koyulmaktadır. Bu eşik üzerinde kalan ara değerler tepe değere yuvarlanmaktadır. *McCulloch* ve *Pitts* modelinde yer alan adım aktivasyon fonksiyonu (0,1) veya (-1,1) değerlerini alabilen bir çıktı üretmesi, girdi verilerinin iki sınıfa ayrılmasını sağlar ve böylece sonuç ikili sayı sisteminde bir çıktı olmaktadır. Yapay Sinir hücresi belirlenen girdilere karşı bir çıktı değeri üretmektedir. Her bir girdi değeri dentritler tarafından çekirdeğe iletilir, çekirdek dentrit vasıtasıyla gelen sinyalleri bir araya toplar ve aksona iletir. Toplanan sinyaller akson tarafından işlenerek sinapslara iletilir. Sinapslar yeni sinyali bir sonraki hücreye gönderir[38].



Şekil 4.2. Matematiksel sinir hücresi modeli [38]

Şekil 4.2’de yer alan sinir hücresi modelinde $X_1, X_2, \dots, X_n$ giriş sinyallerini ifade ederken, $w_1, w_2, \dots, w_n$ ise ağırlıkları ifade etmektedir. X değerleri, ağırlıklar ile çarpılıp toplamları alınmaktadır ve aktivasyon fonksiyonuna $F(NET)$ tabi tutularak çıkışı meydana getirmektedir.

Biyolojik sinir hücresinde Akson’un karşılığı Çıktı,

Dentrit'in karşılığı Toplama Fonksiyonu,

Çekirdek'in karşılığı Aktivasyon Fonksiyonu,

Sinaps'ın karşılığı Ağırlıklar'dır.

McCulloch ve *Pits* modelinin aslında gerçek sinir hücresine çok benzemez çünkü gerçek dünyada sinirler çok daha karmaşık yapıdadırlar. Örneğin; sinir hücresi aktivasyon değerleri doğrusal olmak zorunda değildir. Ancak *McCulloch* ve *Pits* modeli doğrusal aktivasyon sinyallerini esas almıştır. Ayrıca gerçek sinir hücreleri tek bir çıkış yerine birden fazla çıkış vermektedir. Bağlantı değerlerinin güncellenmesi ise bilgisayarlarda sıralı olarak, oluşturulan tüm modeli kapsarken gerçek sinir hücreleri ağırlıkları rastgele olarak ve asenkron biçimde güncellemektedir [38].

Bir sinir hücresini oluşturan kavramlardan girdi, çıktı, ağırlık, hedef, aktivasyon fonksiyonu ve hata kavramlarını açıklayacak olursak:

Girdi: Girdi sinir ağına uygulanan verilerin vektör haline getirilmiş halidir. Girdi vektörleri birden fazla olabildiği için x_i şeklinde ifade edilirse “i” indisi vektör elemanını işaret etmektedir [38].

Çıktı: Girdi vektörü gibi çıktı değeri de bir vektördür. Genel olarak “y” ile ifade edilmektedir [38].

Ağırlık: w_{ij} , i. sinir hücresi ile j. sinir arasındaki ağırlıklı bağlantıyı ifade etmektedir. Ağırlık insan beynindeki sinapsların matematiksel bir sinir ağındaki karşılığıdır[38].

Hedef: Hedef vektörü “t” ile ifade edilir. Hedef ile çıktı aynı ise doğru, yanlış ise hata bulunur ve geriye dönüş olur [38].

Hata: E ile ifade edilir. Hatanın tespiti için farklı hesaplama yöntemleri bulunmaktadır. Örneğin çıktı değeri ile hedef değeri arasındaki farkın *Eucliden* uzaklığını alan sistemler olduğu gibi aradaki değerlerin *Manhattan* uzaklığını alan sistemler de bulunmaktadır. Hata için kayıp veya maliyet fonksiyonu da denilebilmektedir [38].

Manhattan ve Öklid mesafeleri Eş. 4.1 ve Eş. 4.2’de gösterildiği gibidir.

(x_1, y_1) ve (x_2, y_2) koordinatları bilinen iki boyutlu uzayda birer nokta olsun;

Manhattan mesafesi;

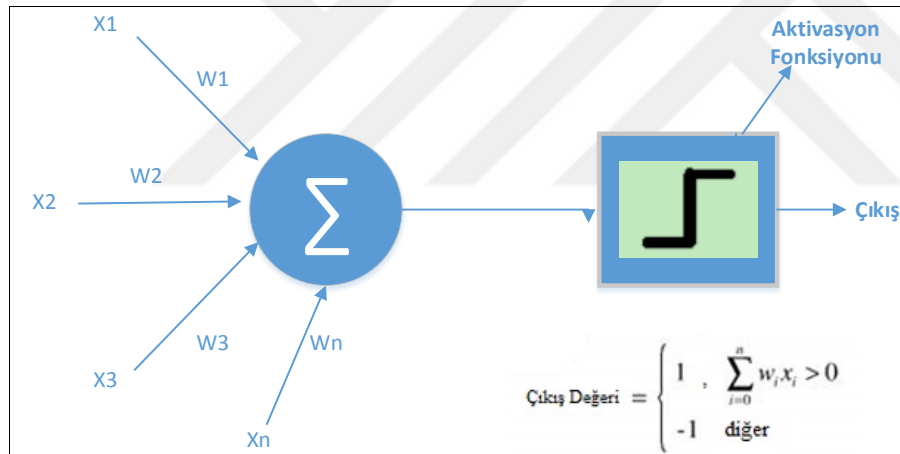
$$|x_1 - x_2| + |y_1 - y_2| \quad (4.1)$$

Öklid mesafesi;

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (4.2)$$

4.2.3. Perseptron

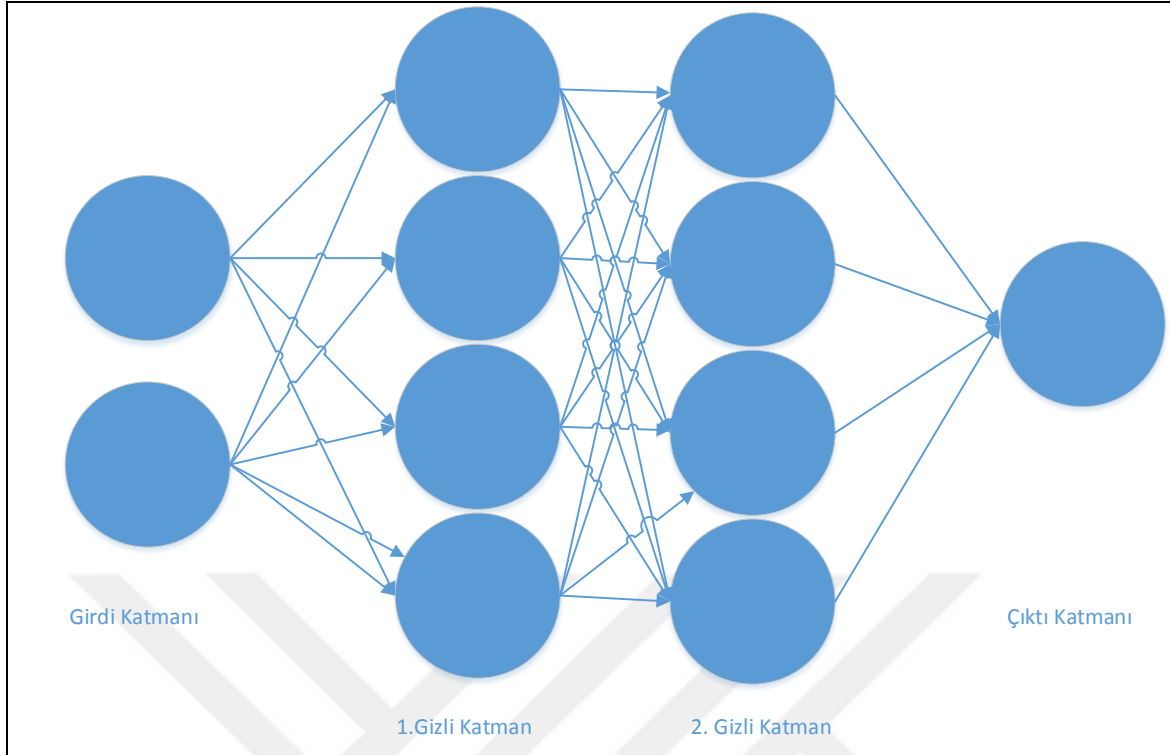
Perseptron, 1958 yılında F.Rosenblatt tarafından ortaya konmuştur. *Perseptron*, *McCulloch* ve *Pits* sinir hücrelerinin bir araya getirilmesi ile oluşturulan bir yapıdır. *Perseptron*' un sahip olduğu sinir hücresi bir diğer sinir hücresine bağlı değildir. Yani diğer sinir hücrelerinin ne tür bir değer ürettiği veya değer üretmediği önemli değildir. Ancak *Perseptron* sinir hücreleri girdi değerlerini paylaşırlar. Girdi değerleri ağırlıklı bir bağlantı ile her bir sinir hücresine ulaşmaktadır. *Perseptron*' un yapısı aşağıda Şekil 4.3' de gösterildiği gibidir [39].



Şekil 4.3. Perseptron'un yapısı [39]

4.2.4. İleri yayılım

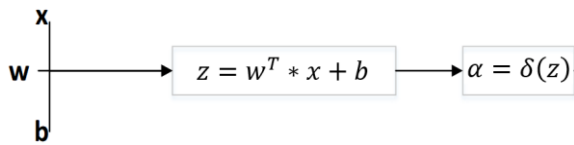
Reel hayatta tek bir sinir hücresi ile sonuca gitme ihtimali yoktur. Sonuca gitmek için birden fazla sinir hücresine ve bu sinir hücreleri arasında da daha karmaşık bağlantılara ihtiyaç vardır. Sinir ağları üç ana katmadan oluşmaktadır. Bunlar; girdi katmanı, gizli katman ve çıktı katmanıdır. Toplam katman sayısına girdi katmanı dâhil değildir. Çünkü girdi katmanının tek görevi girdi verilerini gizli(ara) katmanlara iletmektir. Şekil 4.4' te yapay bir sinir ağı ve katmanları verilmiştir [40].



Şekil 4.4. Yapay bir sinir ağı [40]

Ara katman sayısı her hangi bir koşul ile sınırlı değildir. En az bir ara katmana sahip sinir ağları, kabul edilebilir bir doğruluk skoru ile bir fonksiyonu tahmin edebilecek şekilde kabul görmektedir. Problemlerin çözümünde karşılaşılan düşük veya yetersiz doğruluk skorları gizli katman sayısı ve girdi ile çıktı arasında bağlantı ile ilişkilendirilmektedir [40].

Girdi katmanı, verileri vektör veya bir matris olarak alıp ara katmanlara iletmektedir. Girdi verileri bir resim veya bir sayı dizisi olabilmektedir. Girdi katmanından çıkan veriler önceden atanmış bir ağırlık değeri ile ilişkilendirilerek gizli katmana iletilmektedir. Gizli katmanda kendinden sonraki, eğer varsa bir sonraki gizli katmana veya çıktı katmana iletmektedir. Bu şekilde sıralı olarak bağlı ağlara ileri yayımlı ağlar adı verilmiştir. Veri sürekli mevcut katmandan bir sonraki katmana aktarılmaktadır [40]. İleri yayılım için örnek bir sinir hücresi Şekil 4.5’de gösterilmiştir.



Şekil 4.5. İleri yayılım için örnek bir sinir hücresi [40]

z Eş. 4.3’de görüldüğü üzere

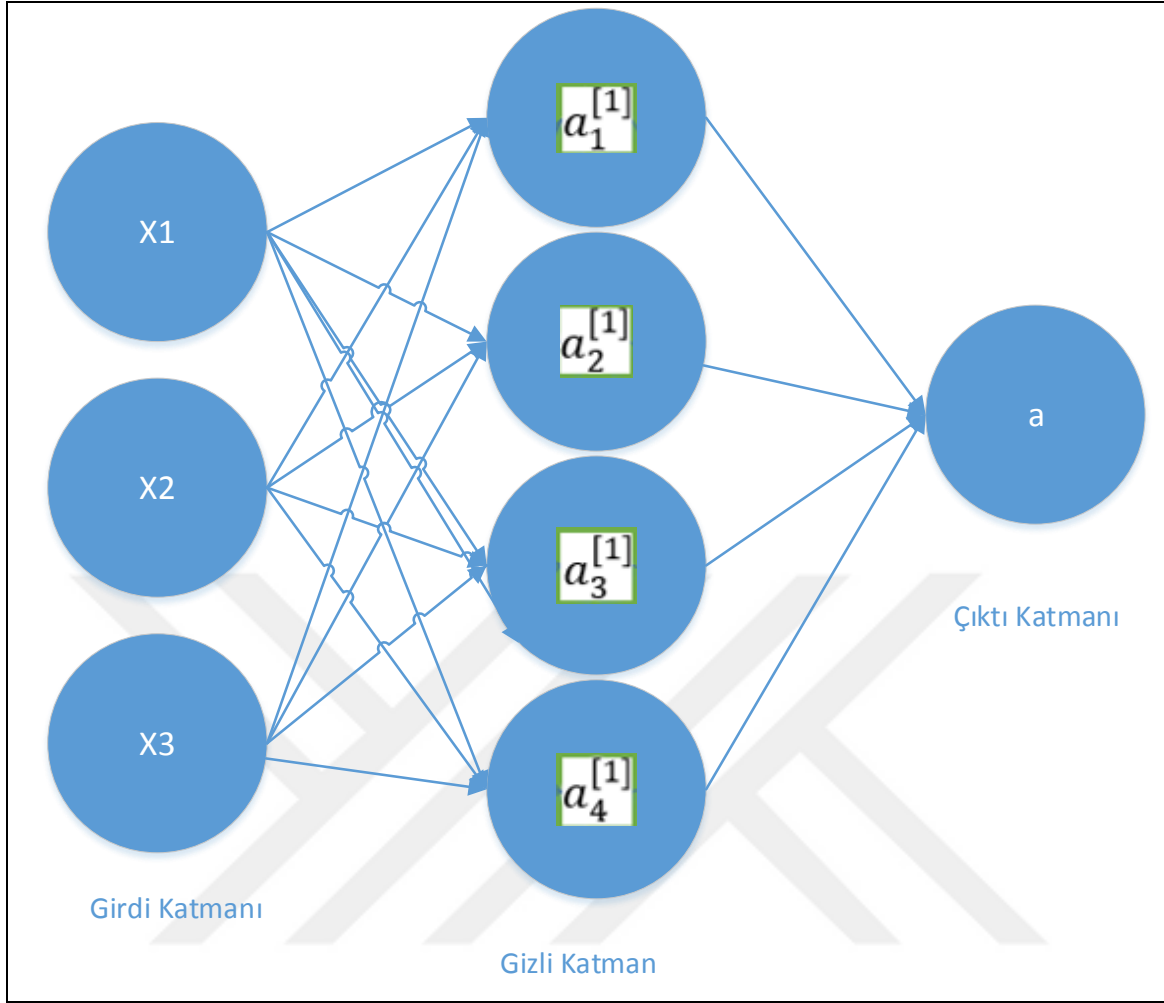
$$z = w^T * x + b \quad (4.3)$$

α Eş 4.4 görüldüğü üzere

$$\alpha = \delta(z) = \frac{1}{1 + e^{-z}} \quad (4.4)$$

şeklindedir.

Örneğin; Şekil 4.5'te yer alan sinir hücresinde x, w ve b sırasıyla girdi verisi, ağırlık ve kutuplanma değerini ifade etmektedir. z değeri sinir hücresinde, ağırlık değeri matrisinin transpozu ile girdi değerinin çarpılıp kutuplanma değeri ile toplanmasıyla elde edilmiştir. x değeri vektörünün, w ağırlık değeri ile çarpılması için w değerinin transpozu alınmaktadır. İki matris ifadesinin çarpılabilmesi için ilk matrisin sütun sayısının ikinci matrisin satır sayısına eşit olması gerekmektedir. Bu eşitliği sağlamak amacıyla transpoz alma işlemi gerekebilmektedir. Daha sonra elde edilen z değeri bir sigmoid aktivasyon fonksiyonuna δ tabi tutulursa α değeri elde edilmektedir. Çıktı olarak ise istenilen, ulaşılmak istenen y değeri ile elde edilen α değeri bir maliyet fonksiyonuna tabi tutulur. Bu fonksiyon iki değer farkı, farkların karelerinin karekökü veya mutlak değerce farkları olabilmektedir [40]. Problemin çeşidine göre maliyet fonksiyonları arasından uygun olan seçilebilir.



Şekil 4.6. Gizli katmana sahip sinir ağında ileri yayılım [40]

Gizli katmana sahip sinir ağında ileri yayılım Şekil 4.6'daki gibi gösterilmiştir. Şekil 4.6'da yer alan x_1, x_2, x_3 girdi verileridir. Gösterimi yapılan şekilde bir gizli katman bulunmaktadır ve katmanlar üst indis ile gösterilmektedir. Girdi katman $a^{[0]}$, gizli katman $a^{[1]}$ şeklinde ifade edilmelidir. n adet katmanı olan bir ağda katmanlar $a^{[1]}, a^{[2]}, a^{[3]}, \dots, a^{[n]}$ şeklinde gösterilmektedir. Ancak Şekil 3.6'da $a^{[2]}$ katmanı çıktı katmanını ifade etmektedir. Bu katmandan sonra elde edilen $\hat{y}$ tahmin edilen değerdir.

$a^{[1]}$, gizli katmanında altı adet sinir hücresi bulunmaktadır. Katmanda yer alan sinir hücreleri sırasıyla $a_1^{[1]}, a_2^{[1]}, a_3^{[1]}, a_4^{[1]}, a_5^{[1]}, a_6^{[1]}$ şeklinde ifade edilmelidir. Gizli katman $a^{[1]}$ 'de sinir hücrelerine ait kutuplanma ve ağırlık değerlerini içeren değerlerde bulunmaktadır. $a_1^{[1]}$ sinir hücresine ait ağırlık değeri $w_{(6,3)}^{[1]}$, kutuplanma değeri ise $b_{(6,1)}^{[1]}$ şeklinde gösterilmektedir. Şekil 3.5'te yer alan formül eşitliğinde ağırlık değerinin neden transpozunun alındığı burada daha net olarak ortaya çıkmaktadır. Çünkü $w_{(6,3)}^{[1]}$ de yer alan

(6,3) alt indisi, altı satırlı üç sütunlu bir matrisi ifade etmektedir. Bu matrisin oluşumu ise bulunduğu katmanda altı adet sinir hücresi, kendine girdi sağlayan üç adet ağırlık olduğu içindir. $w_{(6,3)}^{[1]}$ matrisi ile $b_{(6,1)}^{[1]}$ matrisinin çarpılabilmesinin tek yolu, $w_{(6,3)}^{[1]}$ matrisinin transpozunu almaktır. Aynı rotasyon, çıktı katmanı olan $a^{[2]}$ için düşünülürse ağırlık matrisi $w_{(1,6)}^{[2]}$ ve kutuplanma matrisi ise $b_{(1,1)}^{[2]}$ şeklinde olacaktır. Sonuç olarak $a^{[1]}$ değeri Eş. 4.5'deki gibi olmaktadır.

$$a^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \\ a_5^{[1]} \\ a_6^{[1]} \end{bmatrix} = \begin{bmatrix} z_1^{[1]} = w_1^{[1]T} * x + b_1^{[1]} \\ z_2^{[1]} = w_2^{[1]T} * x + b_2^{[1]} \\ z_3^{[1]} = w_3^{[1]T} * x + b_3^{[1]} \\ z_4^{[1]} = w_4^{[1]T} * x + b_4^{[1]} \\ z_5^{[1]} = w_5^{[1]T} * x + b_5^{[1]} \\ z_6^{[1]} = w_6^{[1]T} * x + b_6^{[1]} \end{bmatrix} = \begin{bmatrix} \delta(z_1^{[1]}) \\ \delta(z_2^{[1]}) \\ \delta(z_3^{[1]}) \\ \delta(z_4^{[1]}) \\ \delta(z_5^{[1]}) \\ \delta(z_6^{[1]}) \end{bmatrix} \quad (4.5)$$

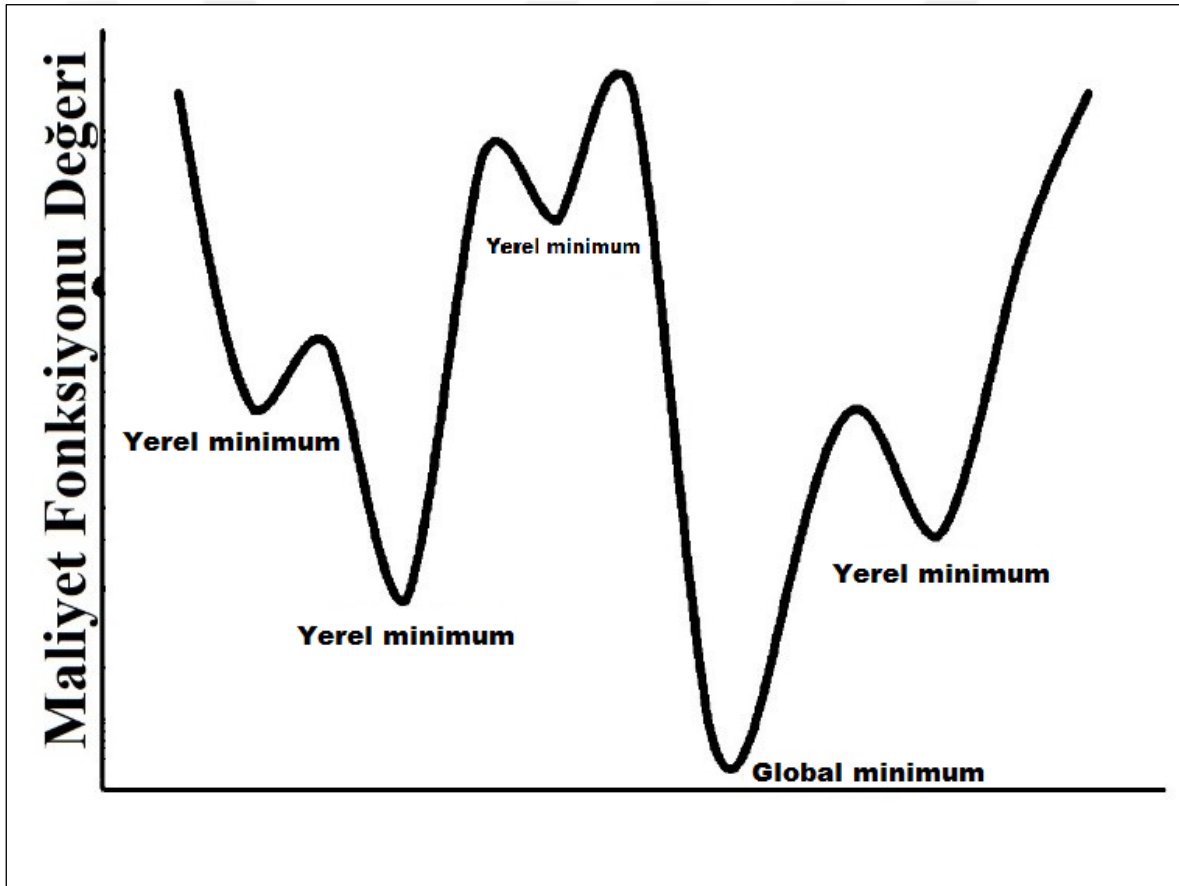
Gizli katman çıktısından sonra $a^{[2]}$ katmanına ulaşan değerler yine belirli bir ağırlık değeri, varsa kutuplanma değeri ile çarpıldıktan sonra aktivasyona tabi tutulacak ve çıktı tahmin değeri elde edilecektir. Böylelikle sinir ağı ileri yayılımını devam edecektir [40].

4.2.5. Geriye yayılım

Rumelhart, Hinton ve Williams (1986) tarafından ilk kez geriye yayılım algoritmasının yapay sinir ağlarında kullanılması ortaya atılmıştır. Geriye yayılım, gerçek değer vektörü ile tahmin vektörü arasındaki farkı en aza indirmek için ağırlıkları hızlı ve etkin olarak ayarlamaktır [41]. Geriye yayılım iki değer arasındaki fark en az olana kadar devam eder. Sinir ağlarında sadece ileri yayılım sonrası elde edilecek tahmin ile gerçek değer arasındaki fark olan maliyet fonksiyonunun(hata) değeri istenen değerden büyük yada istenmeyen bir değer olabilmektedir. Bu noktada geriye yayılım devreye girmektedir. Geriye yayılım yapılarak ağırlık değerleri güncellenmektedir ve bu yeni ağırlık değerleri ile tekrar ileri yayılım yapılmaktadır. Bu şekilde ağırlıkların geriye doğru güncellenmesi ve tekrar yayılması yöntemi ile elde edilen doğruluğun arttırılmaktadır.

Geriye yayılım algoritmasının gerçekleşmesinde en önemli nokta ileri yayılımda türevlenebilir aktivasyon fonksiyonlarının kullanılmasıdır. Geriye yayılımda aktivasyon fonksiyonlarının türevi alınmaktadır. Böylece bulunan eğimin negatif veya pozitif olmasına göre mevcut ağırlıklar güncellenmektedir. Hedef global minimum adı verilen en düşük maliyet fonksiyonu değerini veren noktaya ulaşmaktır [41].

Optimum seviyeye ulaşılan dek elde edilen tahmin ve gerçek değerler arasındaki fark alınarak sinir ağı boyunca ağırlıklar güncellenmektedir. Nihai olarak geriye yayılım eğrisi en iyi noktasına ulaşınca ya da elde edilen hata değeri sıfıra çok çok yakın bir noktaya gelinceye kadar devam eder.



Şekil 4.7. Yerel ve global minimum noktaları [42]

Derin öğrenme problemlerinde birden fazla minimum olabilmektedir. Ancak çözülmesi gereken problem bir adet global minimuma sahiptir. Diğer noktalar yerel minimum olarak adlandırılır. Yerel minimum noktalarında eğim sonucu sıfır olduğu için geriye yayılım bu noktaları global minimum olarak değerlendirebilmektedir. Bu durumun aşılması için yada daha fazla döngü(epok) yapılmalı yada öğrenme katsayısı yerel minimumlara takılmayacak

şekilde ayarlanmalıdır. Şekil 4.7’de olası yerel ve global minimum noktaları gösterilmiştir [42].

Yerel minimum haricinde yine çok sık olarak karşılaşılan bir diğer sorun ise plato olarak adlandırılan ve özellikle yüksek epok değerlerinde ve aynı zamanda adaptif öğrenme katsayısının düşük olduğu durumlarda ortaya çıkmaktadır. Plato veya düzlük denilen bu bölgelerde maliyet değişim göstermez. Bu bölgelerde elde edilen türev değeri sıfır değerine oldukça yakın olacağından ağırlıkların güncellenmesi ile de bu sorun aşılamamaktadır. Yapılan her iyileştirme sonrası gelen yeni epokta adaptif azalan öğrenme katsayısı daha küçük adımlar atarak global minimuma yaklaşmakta sorun yaşamaktadır [42].

4.2.6. Aktivasyon fonksiyonları

Aktivasyon fonksiyonları bir sinir hücresinin çıkış değerinin üretilmesini sağlarlar. Aktivasyon fonksiyonu olarak geçmişte daha çok sigmoid fonksiyonları kullanılmaktaydı. Günümüzde ise birçok aktivasyon fonksiyonu bulunmaktadır. Çizelge 4.1’de çeşitli aktivasyon fonksiyonları verilmiştir.

Çizelge 4.1. Aktivasyon fonksiyonları [43]

Aktivasyon Adı	Aktivasyon Fonksiyonu
Sigmoid [43]	$\delta(x) = \frac{1}{1 + e^{-x}}$
ReLU [43]	$f(x) = \max(0, x)$
LeakyReLU [43]	$f(x) = \begin{cases} ax, & x < 0 \\ x, & x \geq 0 \end{cases}$
ELU [44]	$f(x) = \begin{cases} x, & \text{eğer } x > 0 \\ a(e^x - 1), & \text{eğer } x \leq 0 \end{cases}$
SELU [45]	$f(x) = \lambda \begin{cases} x, & x > 0 \\ ae^x - a, & x \leq 0 \end{cases}$
PReLU [46]	$f(x) = \begin{cases} x, & x > 0 \\ ax, & x \leq 0 \end{cases}$
Tanh [43]	$f(x) = 2\delta(2x) - 1$
Softmax [47]	$f(x) = \frac{e^{z_j}}{\sum_k e^{z_k}}$

Aktivasyon fonksiyonları formüllerinin parametreleri [48]

x: değişken

e: üstel fonksiyon, eksponansiyel

α : sabit katsayısı

δ : delta

k : vektör boyutu

4.2.7. Optimizasyon algoritmaları

Geri yayılımda maliyet fonksiyonu değeri yani hatanın büyüklüğüne göre ağırlıklar güncellenmektedir. Türev alınarak hesaplanan eğimin global minimum noktasına ulaşması için çeşitli optimizasyon algoritmaları üretilmiştir. Bu algoritmalar ile ağırlıklar güncellenerek ileriye tekrar yayılım sağlanmakta ve problemin çözümüne yönelik yol kat edilmektedir.

Bu optimizasyon algoritmaları Çizelge 4.2’de verilmiştir.

Çizelge 4.2. Optimizasyon algoritmaları [48]

Algoritma Adı	Fonksiyon
Momentum[48]	$v_{t+1} = \gamma v_t + \eta \nabla_{\theta} J(\theta)$ $\theta_{t+1} = \theta_t - v_{t+1}$
Nesterov Hızlandırılmış Eğimi [48]	$v_{t+1} = \gamma v_t + \eta \nabla_{\theta} J(\theta - \gamma v_{t-1})$ $\theta_{t+1} = \theta_t - v_t$
RMSprop [48]	$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$
Olasılıksal eğim düşümü (SGD) [48]	$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} J(\theta_t; x^{(i)}; y^{(i)})$
Adagrad [48]	$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} g_{t,i}$
Adadelata [48]	$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$
Adam [48]	$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\frac{v_t}{1 - \beta_2^t} + \epsilon}} \frac{m_t}{1 - \beta_1^t}$
Adamax [48]	$\theta_{t+1} = \theta_t - \frac{\eta}{(1 - \beta_1^t) u_t} m_t$
Nadam [48]	$\theta_{t+1} = \theta_t - \frac{a_t}{\sqrt{\hat{n}_t + \epsilon}} m_t$

Algoritma formüllerindeki parametreleri [48]

η : Öğrenme katsayısı;

$\nabla_{\theta} J(\theta_t; x^{(i)}; y^{(i)})_{\theta}$: Parametrelere bağlı olarak hedef fonksiyonunun eğimi;

$G_{t,ii}$: Her bir köşegen elemanı, $\theta\theta_{ii}$ parametresine göre, t. iterasyona kadar hesaplanmış eğim değerlerinin kareleri toplamı;

ϵ : Öğrenme katsayısının 0'a bölünmesini engellemek için atanan sabit değer;

v_t : Geçmiş eğimlerin karelerinin üssel olarak ağırlıklandırılmış ortalamaları;

m_t : Momentum değişikliği;

β_1 : Varsayılan değer (0,9)

β_2 : Varsayılan değer (0,999)





5. EVRİŞİMLİ SİNİR AĞLARI (ESA)

Evrişim (Convolution) işlemi sinyal ve görüntü işlemede kullanılan önemli bir işlemdir. Evrişim; Verilen $f(t)$ ve $g(t)$ fonksiyonlarının örtüşme miktarını gösteren $f(s)g(t-s)$ ifadesinin tüm zaman eksenini boyunca tümlevi olarak hesaplanan ve doğrusal sistemlerde, çıkış sinyalinin belirlenmesi için, girdi sinyali ile sistem tepki fonksiyonunu arasında uygulanan işlemidir. Evrişim işlemi son yıllarda derin ağlarda da sıkça kullanılmaya başlamıştır. Evrişimin tercih edilmesinin bir sebebi de derin ağlarda büyük veriler ile çalışabilme özelliğidir. Matematiksel olarak evrişim işlemi Eş. 5.1’de görülmektedir [49].

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \quad (5.1)$$

Evrişimli sinir ağı (ESA) kullanımının artışıındaki önemli faktörlerden biri de GPU teknolojisindeki gelişmelerdir. Günümüzde paralel çalışan GPU donanımları ile eğitilmesi imkânsız olan ağlar artık eğitilebilmektedir. Evrişim işlemini temel alan LeNet, AlexNet, ResNet, VGG, Inception gibi popüler ağ mimarileri bulunmaktadır. Bu mimariler evrişim işlemlerinin farklı dizilimleri ile birleşim yapılarak oluşturulmuştur [49].

Evrişimli sinir ağlarının başlıca katmanları aşağıdaki gibi verilmiştir.

5.1. ESA’nın Başlıca Katmanları

Evrişimli sinir ağı mimarilerinde temel olarak aşağıdaki katmanlar kullanılır.

- Evrişim katmanı,
- Havuzlama – ortaklama katmanı,
- Tamamı bağlı katman,
- Dolgulama işlemi,
- Paket normalizasyonu,

şeklinde olsa da bahsedilmeyen ancak sıklıkla kullanılan farklı katmanlar da bulunmaktadır.

5.1.1. ESA’da evriřim iřlemi

Evriřim iřlemi derin aęlar iin filtreler yardımıyla uygulanmaktadır. rneęin iki boyutlu bir grnt iin evriřim iřleminde uygulanacak filtrenin x ve y eksenine gre simetrięi alınır. Elde edilen filtre deęerleri ile grnt iindeki filtreye karřılık gelen deęerler arpılır ve ıktı olarak ilgili matrisin elemanı olarak yazılır. Bu iřlemleri matematiksel olarak gsterecek olursak Eř. 5.2’de ki gibi olacaktır [50].

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 & 2 \\ 2 & 0 & 3 & 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 \\ 0 & 2 & 0 \\ 3 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 3 \\ 0 & 2 & 0 \\ 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 7 & 8 & 2 \\ 11 & 3 & 0 & 6 \\ 9 & 2 & 8 & 4 \\ 8 & 3 & 9 & 8 \end{bmatrix} \quad (5.2)$$

Eř. 5.2’de yer alan evriřim iřlemi uygulanacak matris 6x6 boyutunda olup, uygulanacak filtre ise 3x3 boyutundadır. Evriřim iřleminde ıkıř matrisinin boyutunun hesaplanması iin kullanılan forml Eř. 5.3’te verilmiřtir.

$$\frac{n + 2p - f}{s} + 1 \quad (5.3)$$

Eř. 5.3’te yer alan n ; girdi verisi olan matris boyutlarını,

p ; eęer varsa dolgulama boyutunu,

f ; evriřim filtresi boyutunu,

s ; eęer varsa adım kaydırma deęerini ifade etmektedir. Eęer dolgulama ve adım kaydırma zellikle belirtilmemiřse bu deęerler sırasıyla 0 ve 1 olarak alınmaktadır.

Eř. 5.3 ile Eř. 5.2’de ki matrisin evriřim iřlemi sonrası boyutu hesaplanırsa ařaęıdaki deęer elde edilecektir.

$$\frac{n + 2p - f}{s} + 1 = \frac{6 + 2 * 0 - 3}{1} + 1 = 4 \quad (5.4)$$

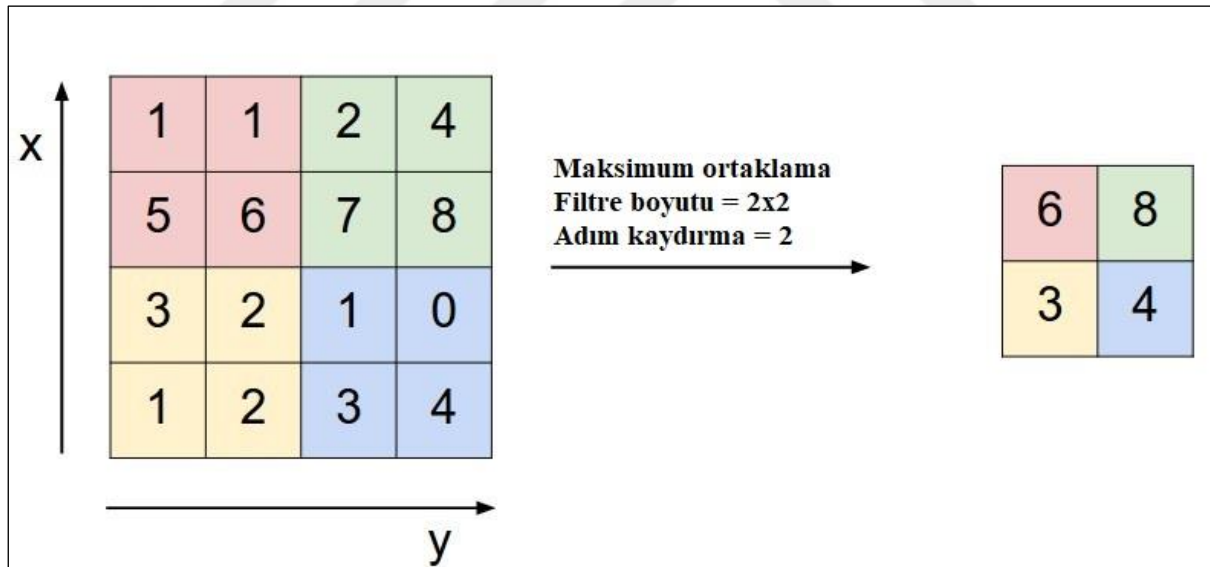
Eş. 5.2’de yer alan çıkış matris boyutu 4x4 elde edilmiştir.

İki boyutlu bir matris için evrişim işlemi yukarıdaki gibi olmaktadır ancak matris üç boyuta çıkarıldığından evrişim işlemi yapılacak filtre sayısı üçe çıkarılmalıdır. Örnek verilecek olursa RGB formatında bir görüntü için Eş. 5.2’deki işlem yapılacağı zaman 3x3 filtreden üç tane ihtiyaç olacaktır. Çıkış matrisi de yine üç boyutlu olacaktır [50].

5.1.2. Ortaklama (Pooling) işlemi

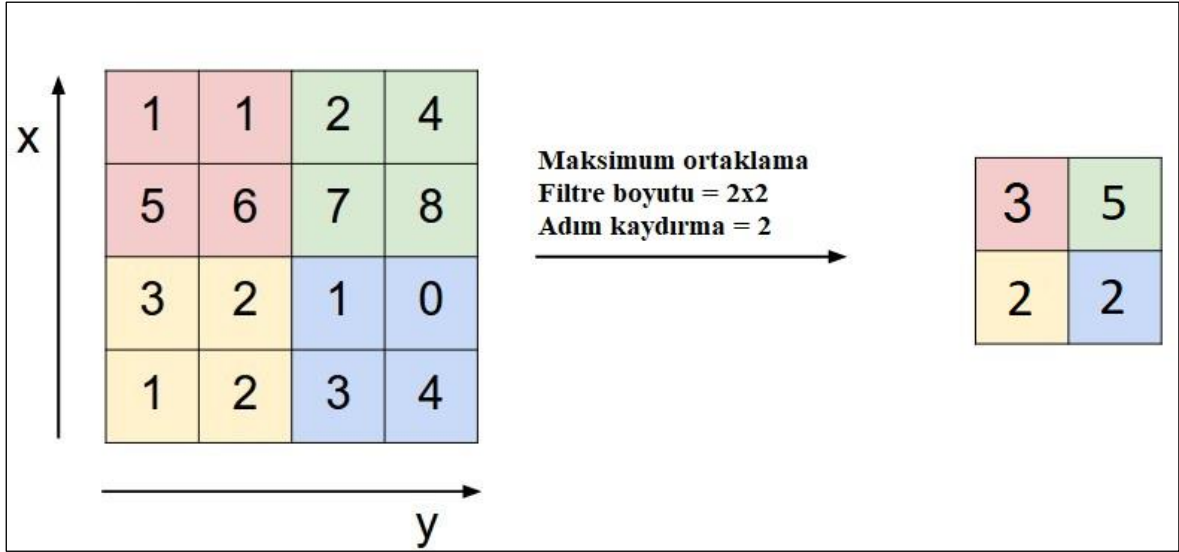
Evrişim işlemi yapılan bir görüntü, boyut olarak büyük ise ortaklama (pooling) işlemi yapılmaktadır. Bu işleme havuzlama işlemi de denilebilmektedir [43].

Ortaklama işlemi için en çok maksimum ortaklama kullanılmaktadır ancak minimum ve ortalama ortaklama yöntemleri de vardır. Ortalama ortaklama filtre boyutunca matris elemanlarının ortalamasını almaktadır, maksimum ortaklama ise filtre boyutunca matris elemanlarından en büyük olan değeri almaktadır. Şekil 5.1’de maksimum ortaklama işlemi gösterilmiştir [43].



Şekil 5.1. Maksimum ortaklama işlemi [43]

Eğer Şekil 5.1’de ortaklama işlemi ortalama ortaklama olarak yapılsaydı çıktı matrisi Şekil 5.2’de ki gibi olacaktır.



Şekil 5.2. Ortalama ortaklama işlemi [43]

5.1.3. Tamamı bağlı katman

Bu katmanlar ileri-geri yayılım prensibine bağlı olarak çalışabilen çok sayıda sinir hücresinden oluşan ve aktivasyon fonksiyonuna sahip, gizli katman olarak adlandırabileceğimiz yapılardır. Tamamı bağlı katman bir evrişim işlemi sonrasına veya bir ortaklama işlemi sonrasına bağlanabilmektedir. Kendinden önceki tüm girdileri lineer bir düzlemde her bir girdi bir sinir hücresine veya birden fazla girdi tek sinir hücresine bağlanacak şekilde konumlandırılabilirler [51].

5.1.4. Dolgulama(Padding) işlemi

Evrişim işleminden sonra giriş matrisi boyutlarında azalma olur. Giriş matris boyutunun azalmasının istenmediği durumlarda dolgulama kullanılır. Dolgulama işlemi başlı başına bir ESA katmanı değildir. Kimi durumlarda kullanılmasının avantaj sağlandığı görülmüştür. Keras gibi geliştirme araçlarında dolgulama bir katman olarak kullanılabilir.

Dolgulama olarak bilinen işlem giriş matrisinin çevresine piksel dolgulaması yapmaktır. Bu işlem sonucunda giriş matrisinin boyutu artırılmış olmaktadır. Dolgulama işlemi için genel olarak kullanılan yöntem 0 değerli piksellerden matrisin etrafını çevretilmesidir. Bu yöntem dışında, 1 değerli piksel veya orijinal giriş matrisinin en dış çerçevesindeki değerlerin tekrarı şeklinde de olabilmektedir. Dolgulama, giriş matrisinin yatay ve dikeydeki tüm çevresine

eklendiği için dolgulama yapılan boyutun iki katı kadar giriş matris boyutu artmaktadır. Eş. 5.5'te 0 değerli bir dolgulama işlemi verilmiştir.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & \dots & 0 \\ \vdots & 1 & 1 & 1 & \vdots \\ 0 & \dots & 0 \end{bmatrix} \quad (5.5)$$

Giriş matrisi 3x3 boyutlu ve tamamı 1 değerli bir matristir. Yapılan dolgulama işlemi sonucunda giriş matrisinin dört bir yanı 0 değerli pikseller ile çevrelenmiştir.

Giriş matrisinin boyutu $a \times a$, ağırlık matrisinin boyutu $b \times b$ olduğunda, çıkış matrisi $(a-b+1) \times (a-b+1)$ şeklinde hesaplanır [51].

5.1.5. Paket normalizasyonu

Paket normalizasyonu kendinden önceki katman sonucu elde edilen doğrusal olmayan değeri doğrusal hale getirmekte ve sonucu normalizasyon işlemine tabi tutmaktadır. Özellikle gizli katmanlarda etkisi önemli boyutlardadır. Paket normalizasyonu kovaryans kaymalarını düzeltmektedir [48].

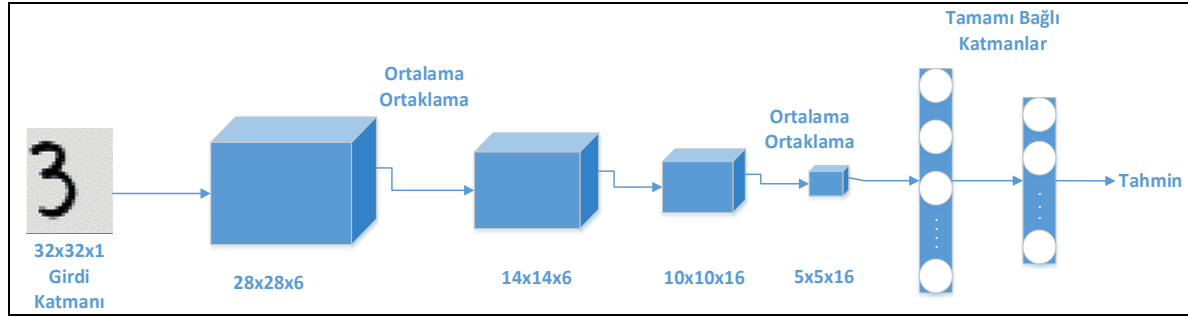
5.2. Popüler Evrişimli Sinir Ağı Mimarileri

Popüler evrişimli sinir ağı mimarilerinden olan LeNet-5, AlexNet, VGG-16, GoogleNet (Inception) ve ResNet alt başlıklardaki gibi sunulmuştur.

5.2.1. LeNet-5

LeNet evrişimli sinir ağı mimarisi LeCun ve diğerleri (1998) tarafından geliştirilmiştir. Bulunduğu dönem için ilk kez derin bir ağ denilebilecek bir ağ mimarisine sahiptir. LeNet evrişimli sinir ağlarının ilki olarak kabul edilebilir. Günümüzde en çok bilinen veri setlerinden olan MNIST veri seti üzerinde eğitilmiştir. Mimarisinde ortalama ortaklama, aktivasyon fonksiyonu olarak sigmoid ve tanh fonksiyonları kullanılmıştır. Ağın son

kısımında tamamı bağlı katmanlardan yararlanılmıştır. Çıktı katmanı ise on sınıf tahminli bir softmax aktivasyonu kullanılan bir katmandır [21]. LeNet-5 ağı mimarisi Şekil 5.3'deki gibi gösterilmiştir.



Şekil 5.3. LeNet-5 ağı mimarisi [21]

LeNet-5 ağında girdi katmanı $32 \times 32 \times 1$ boyutunda gri renk uzayındaki el yazısı rakamlardır. Bu katmandan sonra 5×5 evrişim işlemine tabi tutulmaktadır. İlk evrişim işleminde bir adım kaydırma yapılmaktadır ve altı adet evrişim filtresi kullanılmaktadır. Evrişim işlemi sonucu görüntü boyutu $28 \times 28 \times 6$ olmaktadır. Ortalama ortaklama katmanı filtre boyutu 2×2 ve adım kaydırma boyutu da 2'dir. Bu nedenle ortaklama sonrası boyut $14 \times 14 \times 6$ olmaktadır. İkinci evrişim işlemi ilki ile aynı filtre boyutundadır. İkinci evrişim sonrası boyut $10 \times 10 \times 16$ olmaktadır. İkinci evrişim de filtre sayısı on altıya çıkmıştır. Daha sonra ortaklama katmanı ve tamamı bağlı katmana geçilmektedir. Tamamı bağlı katmanlar iki adettir, ilkinde 120 adet sinir hücresi, ikincisinde ise 84 adet sinir hücresi bulunmaktadır. En son tahmin katmanında ise on sınıflı bir softmax yer almaktadır.

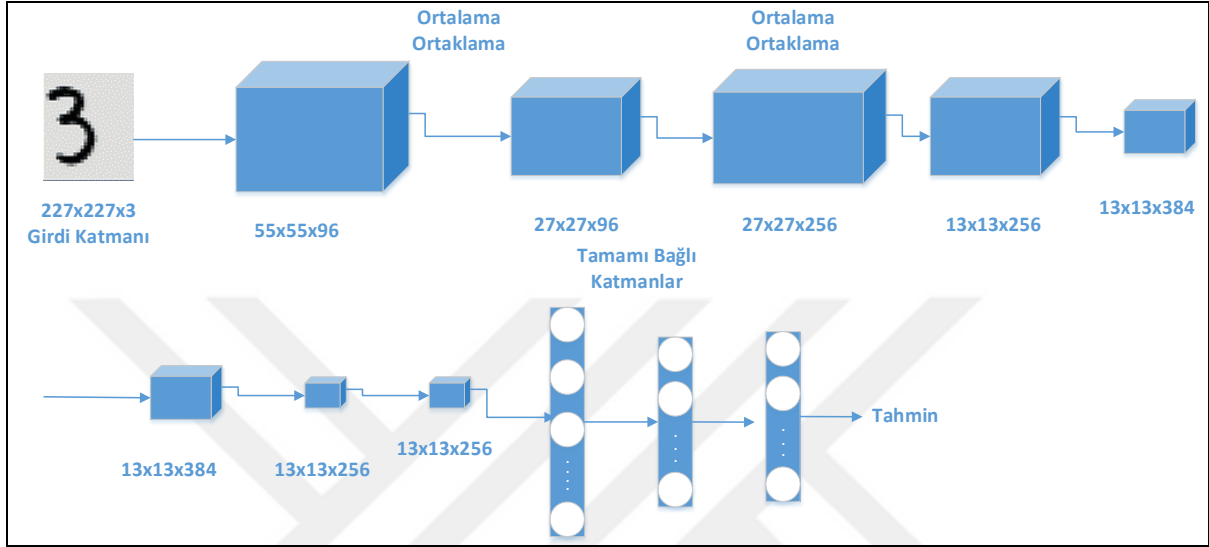
5.2.2. AlexNet

LeNet sonrasında derin ağ mimarilerindeki ilerleyiş on dört yıl kadar yavaşlamıştır. 2012 yılında derin ağ mimarileri için düzenlenen ILSVRC 2012 yarışmasını AlexNet kazanarak gücünü ispatlamıştır [52].

ILSVRC yarışması 14 milyon etiketlenmiş görüntü ve 20 bin sınıfın var olduğu veri seti üzerinden alınan 1,2 milyon kadar eğitim verisinin ve 1000 sınıfın olduğu bir yarışmadır.

AlexNet, tıpkı LeNet gibi evrişim katmanlarına ve ortaklama katmanlarına sahiptir. Fakat AlexNet, maksimum ortaklama ve ReLU aktivasyon fonksiyonu kullanmaktadır. LeNet mimarisinden daha fazla sayıda katmanda ve daha fazla veri üzerinde çalışmaktadır.

AlexNet sekiz katman, 60 milyon parametreye sahiptir. AlexNet eğitimi bir hafta boyunca iki GPU üzerinde yapılmıştır. Aynı zamanda seyreltme (dropout) ve düzeltme (regularization) tekniklerinden yararlanılmıştır [52]. AlexNet ağı mimarisi Şekil 5.4'deki gibi gösterilmiştir.



Şekil 5.4. AlexNet ağı mimarisi [52]

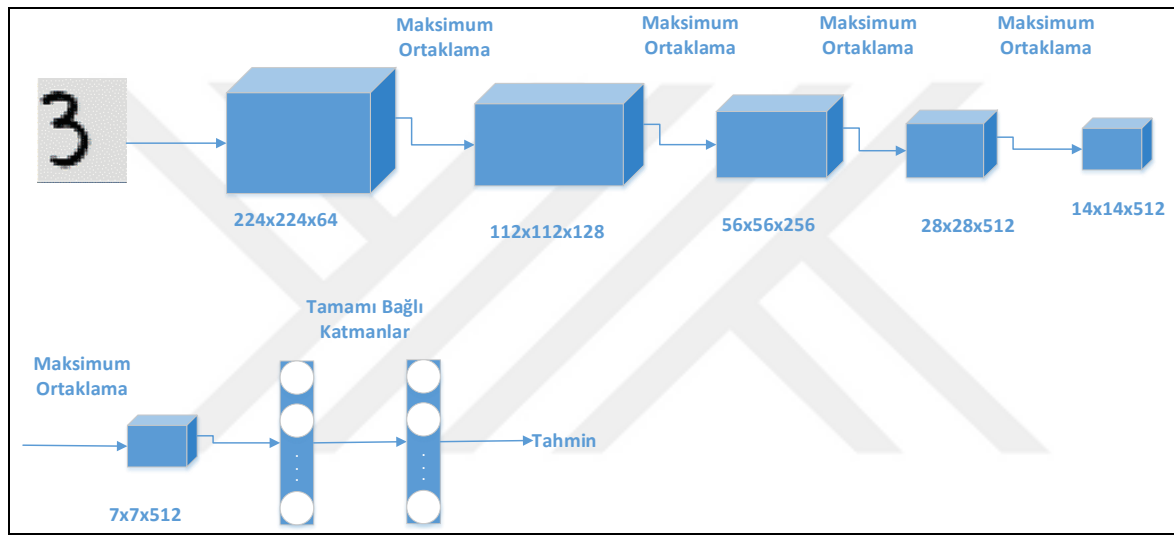
AlexNet öncesi derin öğrenme sınıflandırma doğruluk oranı %74,3 civarlarındaydı. AlexNet ILSVRC yarışmasında sınıflandırma doğruluk oranını %83,6'ya çıkararak LeNet sonrasında yaşanan durgun dönemi bitirmiş ve derin ağlar üzerine olan merakı yeniden canlandırmıştır.

5.2.3. VGG-16

VGG (Visual Geometry Group) mimarisi, Oxford Üniversitesi'nde ILSVRC sınıflandırma yarışması için geliştirilmiştir. AlexNet ağ yapısı kullanılarak bu ağ mimarisi geliştirilmiştir. AlexNet ağından farklı olarak AlexNet'te olmayan art arda evrişim işlemlerini içermektedir. Evrişim filtreleri daha önceki filtre boyutlarından farklı olarak 3x3 gibi çok düşük bir boyuttadır. VGG ağının çeşitli derinliklerde varyasyonları bulunmaktadır. 16 katmanlı olan VGG mimarisi VGG-16 olarak adlandırılırken aynı ağın 19 ve 21 derinlikte olan çeşitleri de bulunmaktadır. VGG-16 ağının çıktı katmanında her biri 4096 nöron içeren iki adet tamamı bağlı katman bulunmaktadır. Bu katmanlardan sonra ise 1000 sınıf için hazırlanmış bir softmax tahmin katmanı bulunmaktadır. VGG-16 ağındaki çok sayıdaki nöron toplam parametre sayısını çok büyük miktarda artırmıştır. VGG-16 ağının toplam parametre sayısı

138 milyon civarındadır. Parametre sayısının fazla olması ile ağıın derlenme ve eğitim süresi de artış göstermiştir [53].

Tıpkı birçok evrişimli sinir ağıında olduğu gibi girişten çıkışa görüntü boyutları azalırken toplam filtre sayısı artmaktadır. ILSVRC yarışmasında VGG-16 ağı konumlandırma ve sınıflandırma alanlarında sırasıyla birincilik ve ikincilik ödülü kazanmıştır. Sınıflandırma hata oranı VGG-16 için %7,4; VGG-19 için %7,3'tür. Doğruluk oranı VGG-16 için %92,6 ve VGG-19 için %92,7'dir [54]. VGG-16 ağı mimarisi Şekil 5.5'de verilmiştir.

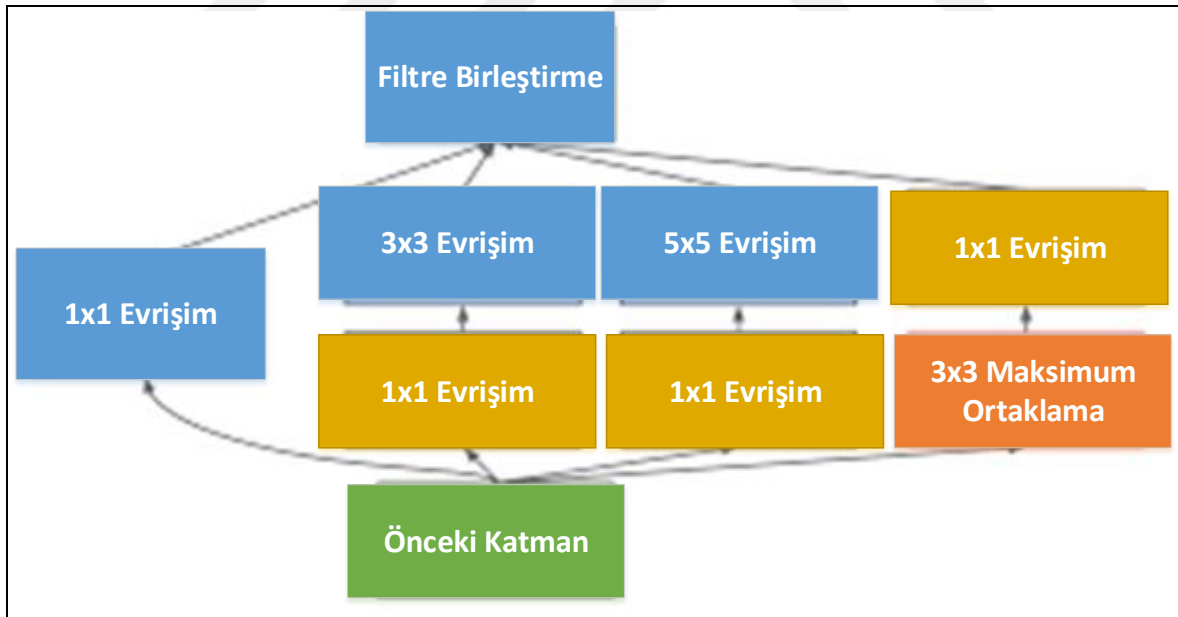


Şekil 5.5. VGG-16 ağı mimarisi [54]

VGG-16 ağı girdi katmanı verisi 224x224 ve RGB yani renkli resim formatındadır. Tüm evrişim işlemlerinde boyutun azalmasını önlemek için dolgulama işlemi yapılmaktadır. Maksimum ortalama katmanları ile ağı derinleştikçe görüntü boyutunun azalması sağlanmaktadır. Evrişim işlemleri art arda aralarında ortalama katmanı olmaksızın yapılmaktadır. Her blok geçişinde bir adet ortalama katmanı bulunmaktadır. İlk iki blok ikişer adet evrişim işleminin art arda uygulaması ile oluşmaktadır. Sonraki bloklarda ise üçer adet evrişim işlemi art arda uygulanmıştır. Tamamı bağlı katman öncesinde görüntü boyutları 7x7 ve toplam filtre sayısı 512 olmuştur. Buradan 25.088 adet doğrusallaştırılmış çıktı oluşmaktadır. Doğrusal çıktılar her biri 4096 adet nöron içeren tamamı bağlı katmana uygulanmaktadır. Son olarak tahmin katmanı 1000 sınıf için ayarlanmış bir softmax katmanıdır [55].

5.2.4. GoogleNet (Inception)

GoogleNet aynı zamanda Inception olarak da adlandırılmaktadır. GoogleNet ağı 2014 yılında düzenlenen ILSVRC yarışmasında birincilik ödülü almıştır. Ağın en büyük avantajı hesaplama maliyeti olan toplam parametre sayısını artırmadan ya da diğer ağlara göre daha az artırarak daha derin ağların oluşmasını sağlamasıdır. Temelde bu görevi 3×3 'lük bir evrişim işlemi yerine aynı sonucu 3×1 ve 1×3 olmak üzere iki adet evrişim işleminden geçirmekle yapmaktadır. Bu işlem toplam parametre sayısının daha az artmasını sağlamaktadır. GoogleNet modern ağlar içerisinde en karmaşık yapıda olan evrişimli sinir ağlarından birisidir. Karmaşık yapıya rağmen hızlı çalışan, parametre sayısı az olan bir ağıdır. GoogleNet ağı bloklardan meydana gelmektedir. Her bloğa Inception bloğu denilmektedir. Yani Inception hem blokların hem de ağın adıdır. Inception bloklarında farklı evrişim katmanları bulunmaktadır. Bu evrişimlerin bazıları 3×1 ve 1×3 bazıları ise 1×7 ve 7×1 şeklindedir. Evrişimlerin sahip olduğu filtre boyutlarında farklılıklar bulunmaktadır [56]. Şekil 5.6'da Inception bloğu mimarisi gösterilmiştir.

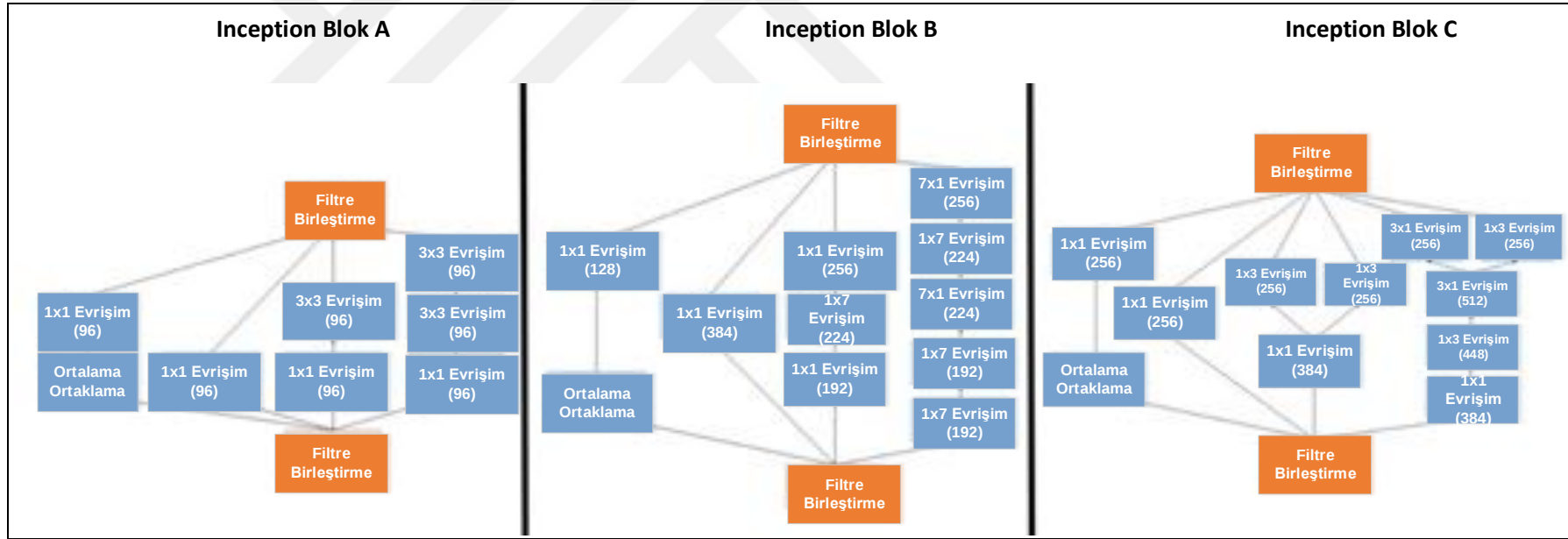


Şekil 5.6. Inception bloğu mimarisi [56]

Şekil 5.6'daki Inception bloğunda 1×1 evrişim işlemleri görülmektedir. Bu evrişim işlemine darboğaz denilmektedir. Şekil 5.6'daki tasarım Inception ağının dayandığı tasarım şeklindedir. Ancak gerçek bir Inception ağında yalnızca bir blok yoktur. Yukardaki bloğun farklı birleşimleri ile elde edilen bloklar bulunmaktadır.

Şekil 5.7’de GoogleNet ağına ait sol baştan sırasıyla Inception A, B ve C blokları görülmektedir. B ve C blokları içerisinde GoogleNet’in öne sürdüğü 3x1, 1x3, 7x1, 1x7 tarzında evrişim işlemleri görülmektedir. Özellikle ağı daha derin katmanlarında B ve C blokları kullanılmaktadır.





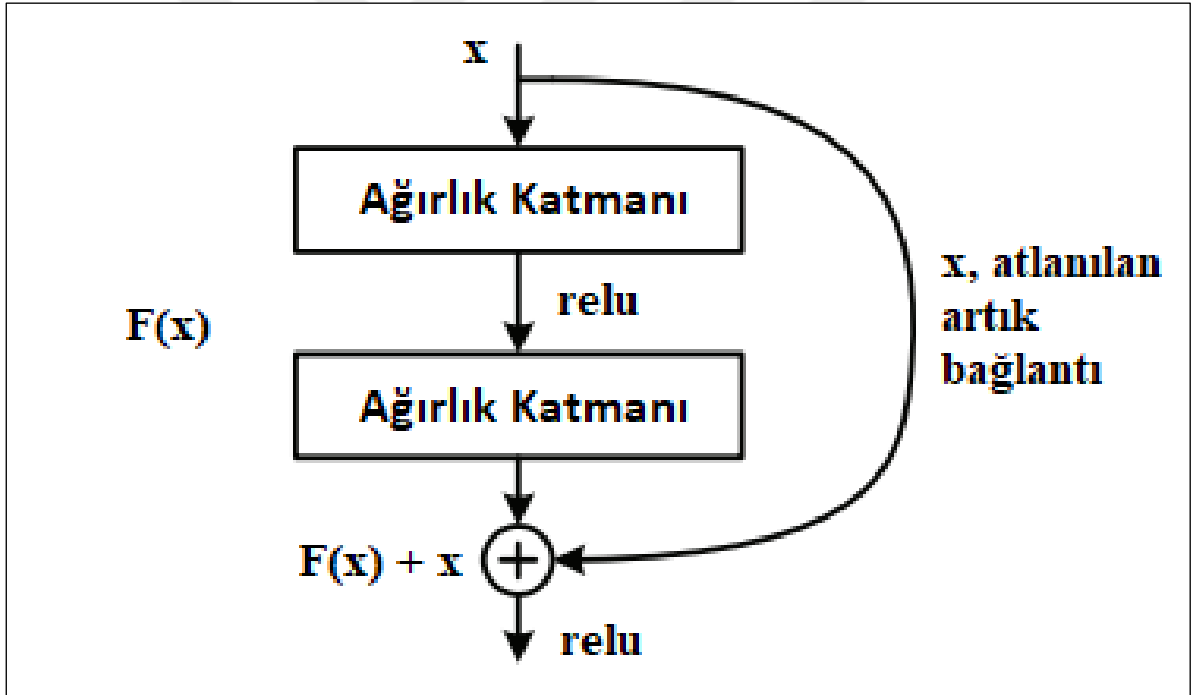
Şekil 5.7. Inception blokları [57]

5.2.5. Microsoft ResNet

Microsoft'a ait ResNet ağında atlamalı blok yapısı vardır. ResNet ağı diğer ağ mimarilerinden farklı olarak bazı evrişim bloklarının atlanarak bir sonraki evrişim katmanına girilmesi prensibi ile oluşturulmuştur.

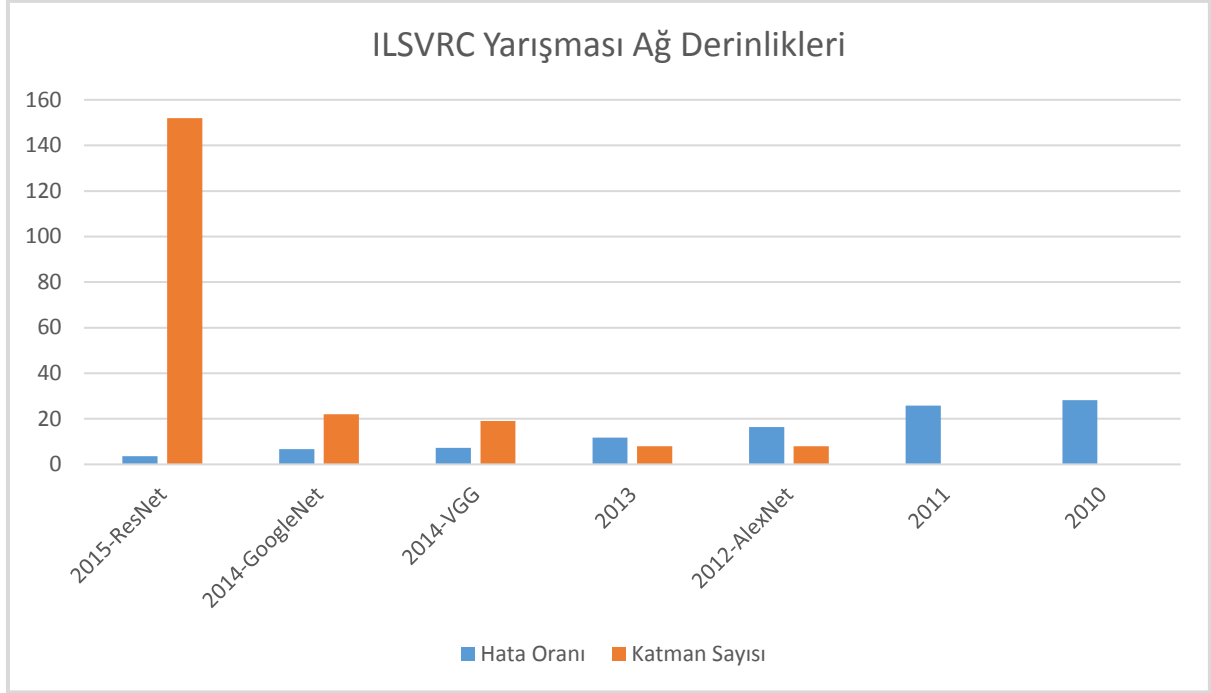
Günümüzde farklı ağ mimarileri birleştirilerek elde edilen kazanımlar ve başarı oranları artırılmaya çalışılmıştır. GoogleNet ağının bazı varyasyonlarında ağ içerisindeki Inception bloklarında atlamalı bağlantılardan yararlanılarak ResNet ana fikri aslında uygulanmıştır.

ResNet Microsoft araştırma ekibi tarafından 2015 yılında duyurulmuştur. ResNet 2015 ILSVRC yarışmasında %3.57 hata oranı ile birinci olmuştur [57]. Şekil 5.8'de ResNet ağı artık bağlantı blok diyagramı verilmiştir.



Şekil 5.8. ResNet ağı artık bağlantı [57]

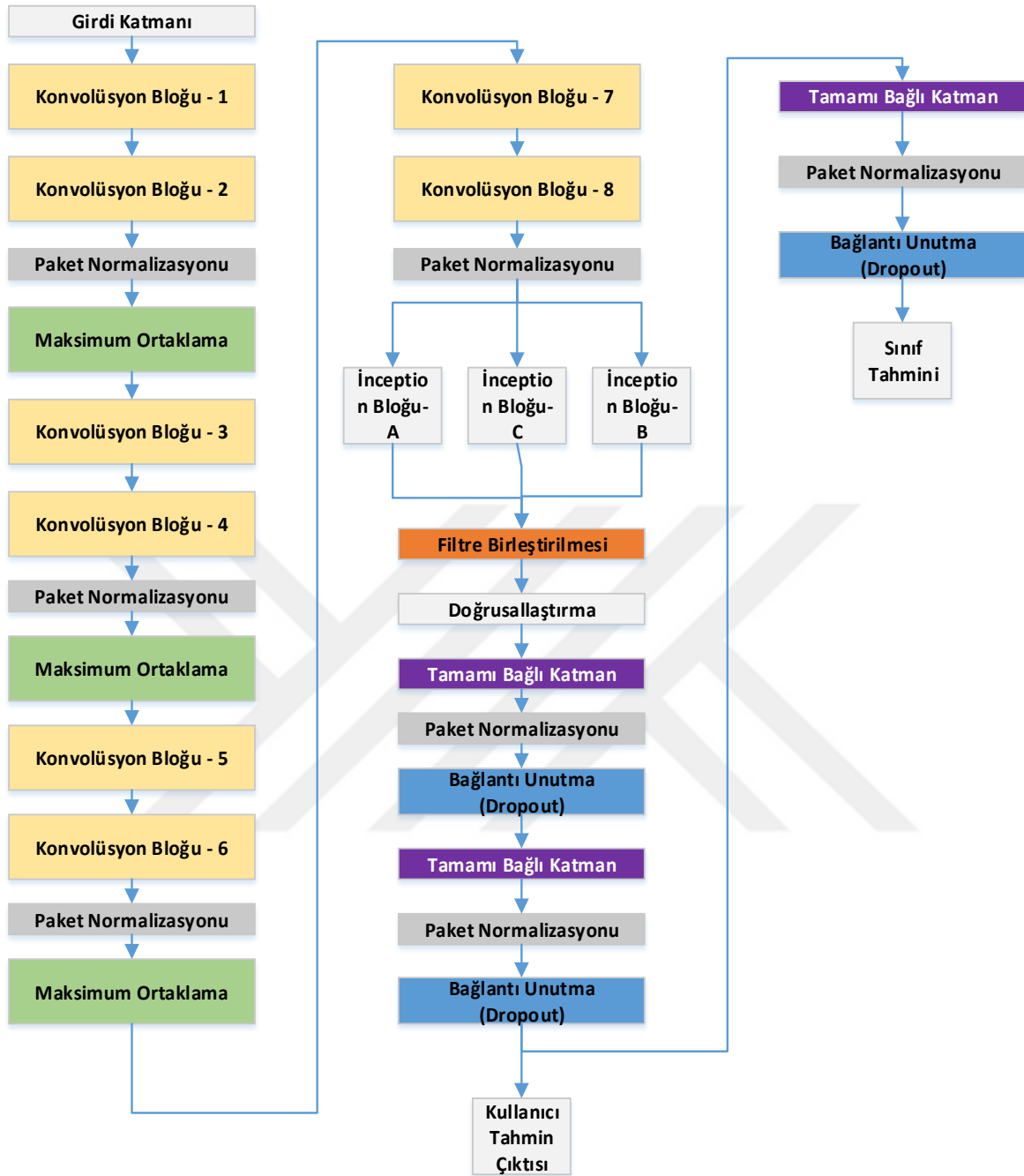
ILSVRC yarışmasında yer alan ağların derinlikleri aşağıda Şekil 5.9'da verilmiştir[58].



Şekil 5.9. ILSVRC yarışması ağ derinlikleri [58]

5.3. Çalışmada Kullanılan ve Önerilen Ağ Mimarisi

Evrişimli sinir ağı mimarilerinin başlıca katmanları farklı diziliş ve parametrelerde kullanılarak birçok ağ mimarisi ortaya çıkarılmıştır. Ortaya çıkarılan mimariler birbirlerinden alıntılar yapılarak mevcut mimariyi daha başarılı hale getirmeye yöneliktir. Bu tez çalışmasında popüler evrişimli sinir ağı mimarilerinden VGG-16 ağına benzer evrişim katmanları ve GoogleNet ağ mimarisine ait Inception blokları kullanılarak hibrid yapıda bir mimari ortaya konulmuştur. Böyle bir hibrid yapı ile iki mimarinin de kendine has avantajlarından yararlanarak daha başarılı bir mimari ortaya konulmaya çalışılmıştır. Ağ mimarisi tek bir girdi katmanına ve iki çıkış katmanına sahiptir. Önerilen ağ mimarisinin blok şeması Şekil 5.10'da gösterilmiştir.



Şekil 5.10. Önerilen ağ mimarisi

Önerilen ağ mimarisinde konvolüsyon blokları tıpkı VGG-16 mimarisindeki gibi ard arda bağlanmıştır. Önerilen mimaride inception bloklarına kadar olan blokların hiper parametre ve filtre sayıları VGG-16 mimarisindeki şekliyle korunmuştur. Burada filtre sayılarının ilk bloklarda 16, 32 gibi düşük sayılarda olup ilerleyen bloklarda artmasının sebebi görüntü boyutunun ilk bloklarda büyük ilerleyen bloklarda azalması sebeplidir. İlk bloklarda çok yüksek sayılarda filtre kullanılınca, çok daha büyük belleğe ihtiyaç duyulup, bellek

yetersizliği nedeniyle eğitimin ilerleyen epoklarında hata verilip ağı eğitimi yarım kalabilmektedir.

Ağ bloklarının detaylı açıklamaları aşağıdaki gibidir.

Evrişim(konvolüsyon) blokları, 3x3 evrişim filtre boyutuna sahiptir.

Blok-1 ve 2, sırasıyla 16 ve 32 adet filtreye sahiptir.

Evrişim blokları 3 ve 4'te sırasıyla 64 ve 96 adet filtre bulunmaktadır.

Evrişim bloğu 5 ve 6'da sırasıyla 128 ve 156 adet filtre kullanılmıştır.

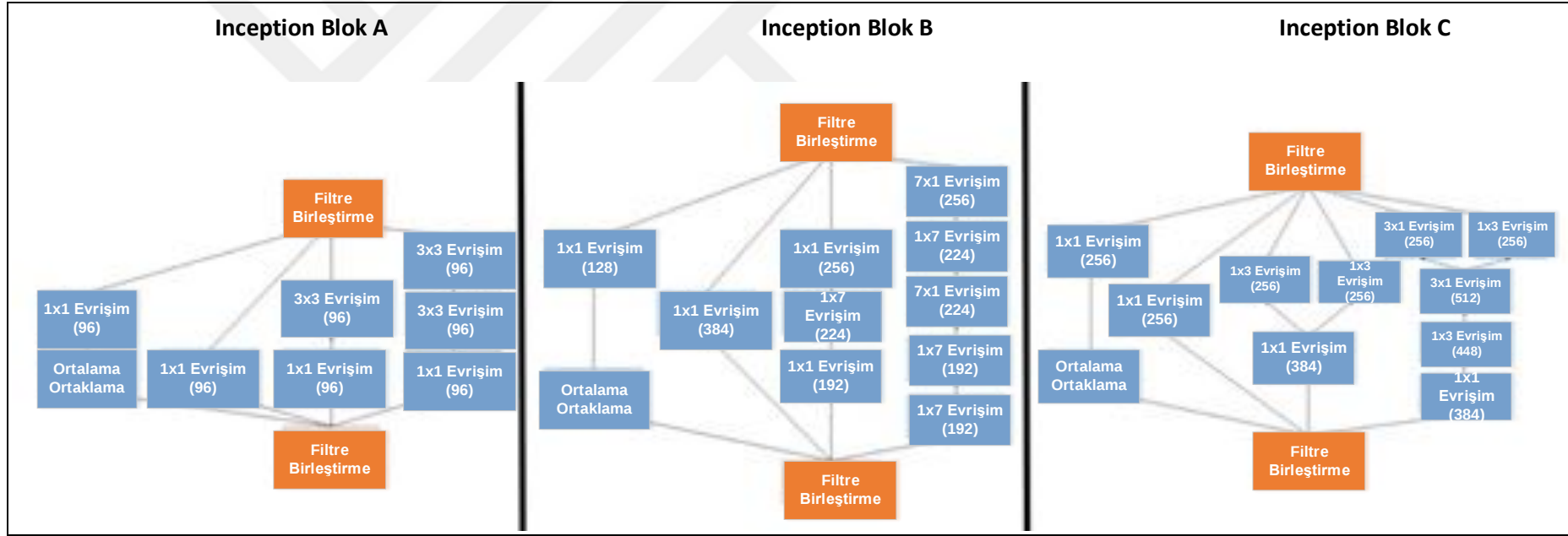
Evrişim blokları 7 ve 8'de sırasıyla 156 ve 192 adet filtreden yararlanılmıştır.

Ağ mimarisinde VGG-16 mimarisindeki şekliyle maksimum ortaklama kullanılmıştır. Ortaklama katmanı boyutu 2x2 ve adım kaydırma olarak s=2 kullanılmıştır.

VGG-16 mimarisi öncesi mimarilerde genellikle Konvolüsyon blokları sonrası Ortaklama işlemi ardından tekrar Konvolüsyon bloğu şeklinde bir yapı kullanılıyordu, VGG-16 mimarisi ile konvolüsyon blokları ard arda kullanılmaya başlanmıştır, bu şekilde kullanım ile derin öğrenmede tanıma için kullanılan özniteliklerin daha belirgin olması sağlanmıştır. Bu tezde de VGG-16 mimarisinin bu avantajı mimariyi oluşturan ilk bloklarda kullanılmıştır.

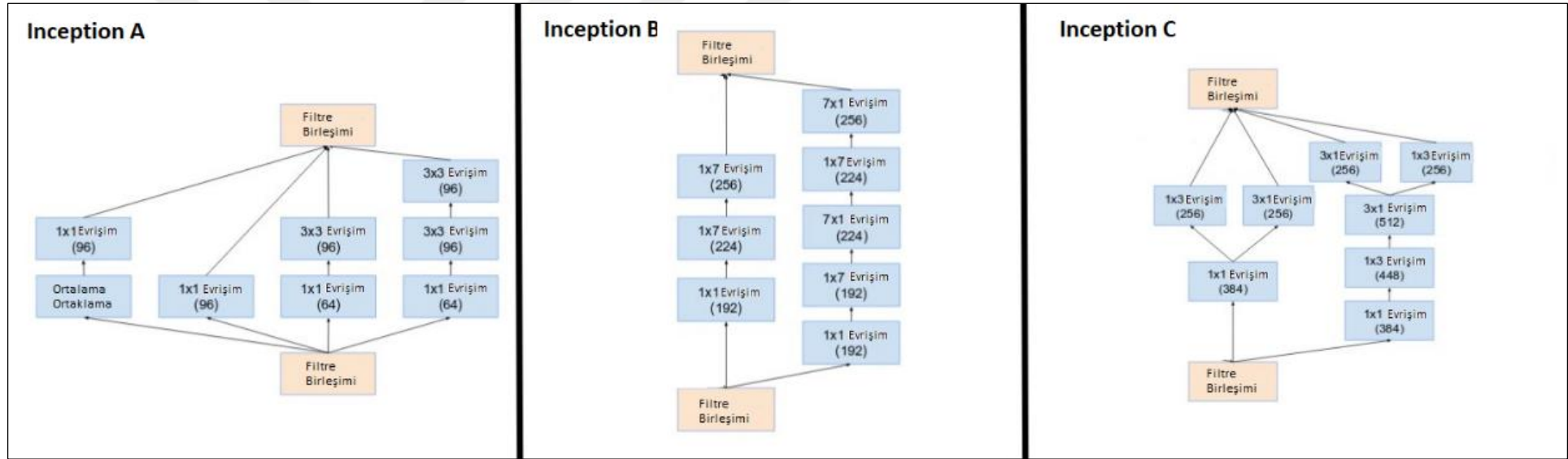
VGG-16 mimarisi blokları sonrası ağ filtreleri doğrusal hale getirilmeden önce GoogleNet ağına ait Inception bloklarından yararlanılmıştır. Hibrid ağ tasarlanırken özellikle GoogleNet(Inception) ağından yararlanılmasının sebebi ise GoogleNet yüksek boyutlu resimlerde dahi daha düşük parametre sayıları ile öznitelikler çıkarabilmektedir [56]. Normalde inception blokları kendi ağ mimarisinde ard arda bağlanırken bu çalışmadaki ağ mimarisinde bu bloklar paralel şekilde VGG-16 mimarisine bağlanmıştır.

Inception A bloğu tamamen aynı şekilde iken B ve C blokları alınırken sadece diğer bloklardan farklı olarak uygulanan evrişim işlemleri korunmuştur. Bu bloklarda özellikle farklı filtre boyutlarına sahip evrişim işlemleri kullanılmaya özen gösterilmiştir, bu sebeple bu filtreler içerisinde benzer işlemi yapan evrişim katmanları çıkartılmıştır. Bu yolla farklı özniteliklerin saptanabilmesi amaçlanmıştır.



Şekil 5.11. Orijinal inception bloklarının yapısı [56]

Şekil 5.11’de Inception orijinal blok dizilimleri yer almaktadır. En sol kısımda Inception A, ortada B, sağda ise C bloğu yer almaktadır. Bu bloklar şekil 5.12’deki gibi değiştirilerek kullanılmıştır.



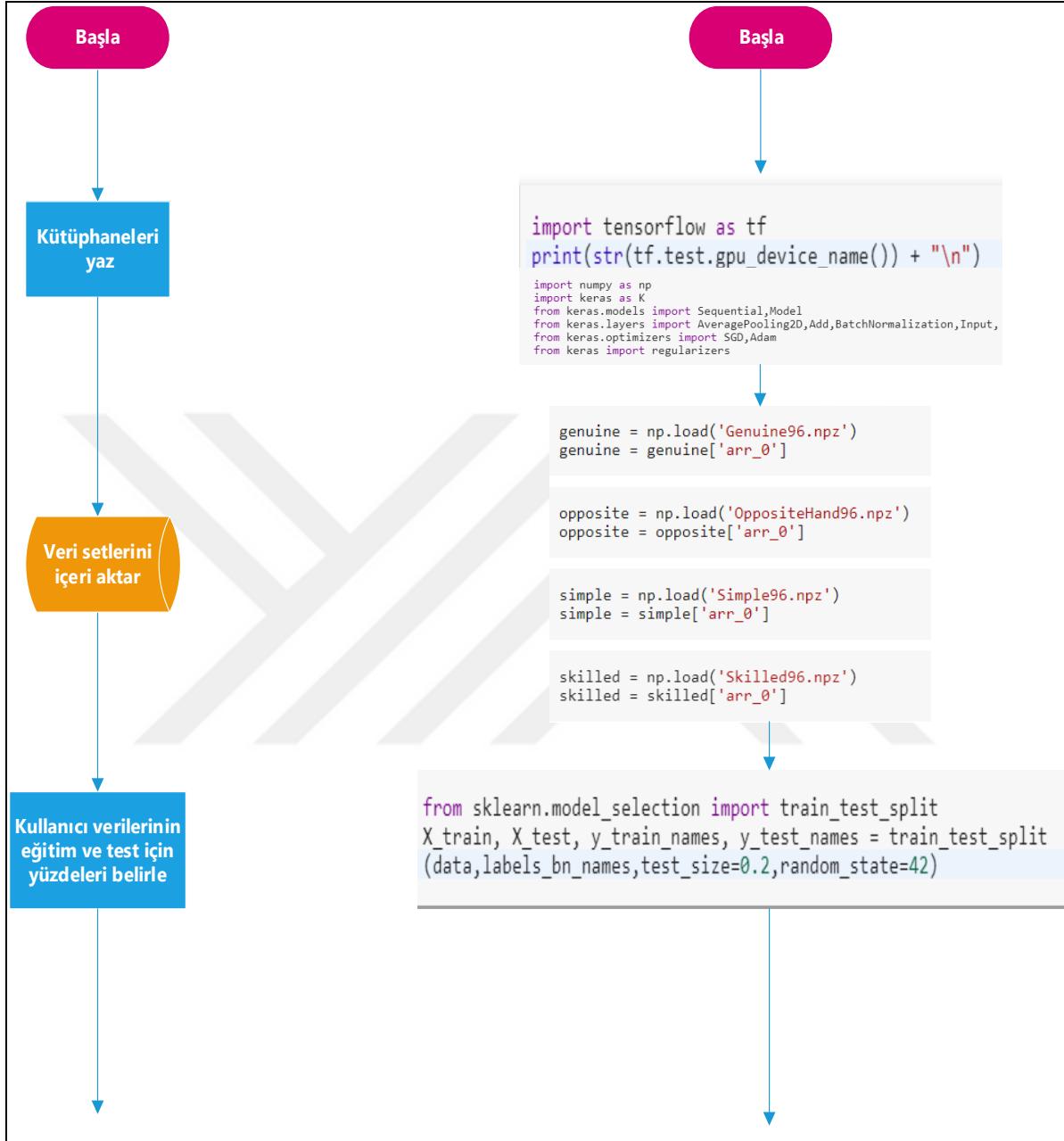
Şekil 5.12. Çalışmada kullanılan inception blok yapıları

Şekil 5.12'deki gibi yapılan değişiklik ile aynı filtre cevaplarının üçer kez kullanılmasının önüne geçilmiştir. Normalde kendi mimarisinde seri bağlanan inception bloklarının bu çalışmada paralel bağlanması sebebiyle ve bloklarında 1x1'lik filtrelerin A,B ve C bloklarında ayrı ayrı kullanılmasının aynı öznelilikleri tekrar tekrar çıkarmasına bunun da boş yere hesap sayısının artmasına neden olması sebebiyle Şekil 5.11'deki orijinal inception bloklarında Şekil 5.12'deki gibi değişikliğe gidilmiştir. Aynı zamanda B bloğunda evrişim işleminde 1x5 ve 5x1 olarak düzenlemeye gidilmiştir. Inception blokları ile filtre sayısı artırılmış, böylece daha farklı özneliliklerin de çıkarılması sağlanmıştır.

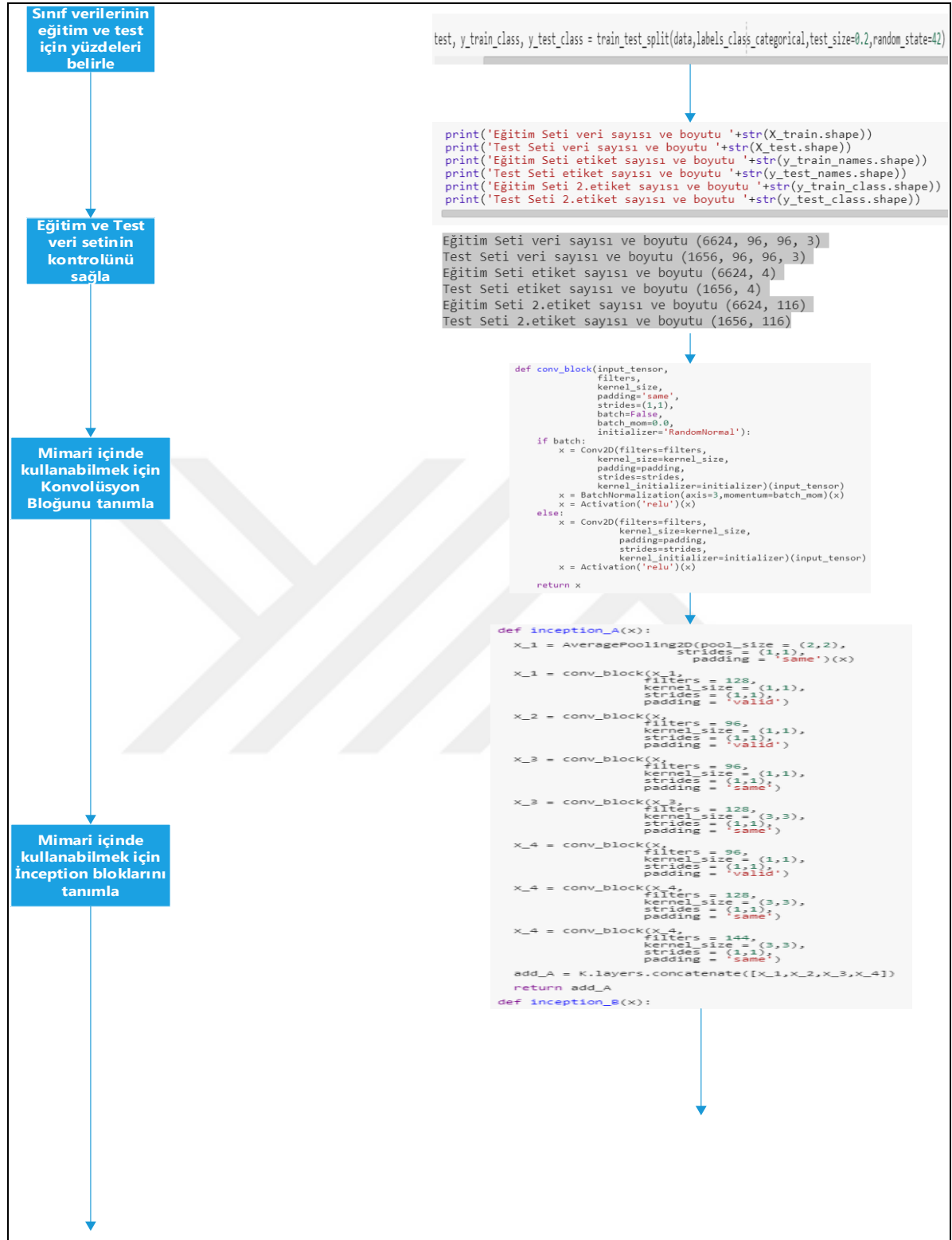
Inception blokları ardından iki adet tamamı bağlı katman kullanılmıştır. Tamamı bağlı katmanlar derin öğrenme mimarilerinde mimarilerin sonuna doğru çıkan sonuçları optimize etmek için kullanılırlar. Bu katmanlar da 256 adet nöron bulunmaktadır. Ağ çıktılarından ilki iki adet tamamı bağlı katmanın olduğu kısma bağlanmıştır ve 115 adet sınıf tahmini için softmax aktivasyonu kullanılmıştır. İkinci çıkış ise 4 adet sınıf tahmini yapan sigmoid aktivasyonlu bir tahmin katmanıdır.

Tamamı bağlı katmanlar arasında ezberleme riskini azaltmak için seyreltme katmanına yer verilmiştir. Seyreltme değeri 0,4 olarak ayarlanmıştır. Modelin yüksek epok değerlerinde seyreltme katmanına rağmen ezberleme riski hala var olduğundan tamamı bağlı katmanlarda düzeltme (regularization) metodu da kullanılmıştır. Bu değer 10^{-5} olarak ayarlanmıştır. Kullanılan 0,4 veya kullanılabilecek 0,6 seyreltme değeri ile 10^{-5} düzeltme değerleri derin öğrenme mimarisindeki varsayılan değerlerdir. Mimarilerde seyreltme ve düzeltme kullanılacaksa genellikle bu ve yakın değerler seçilir.

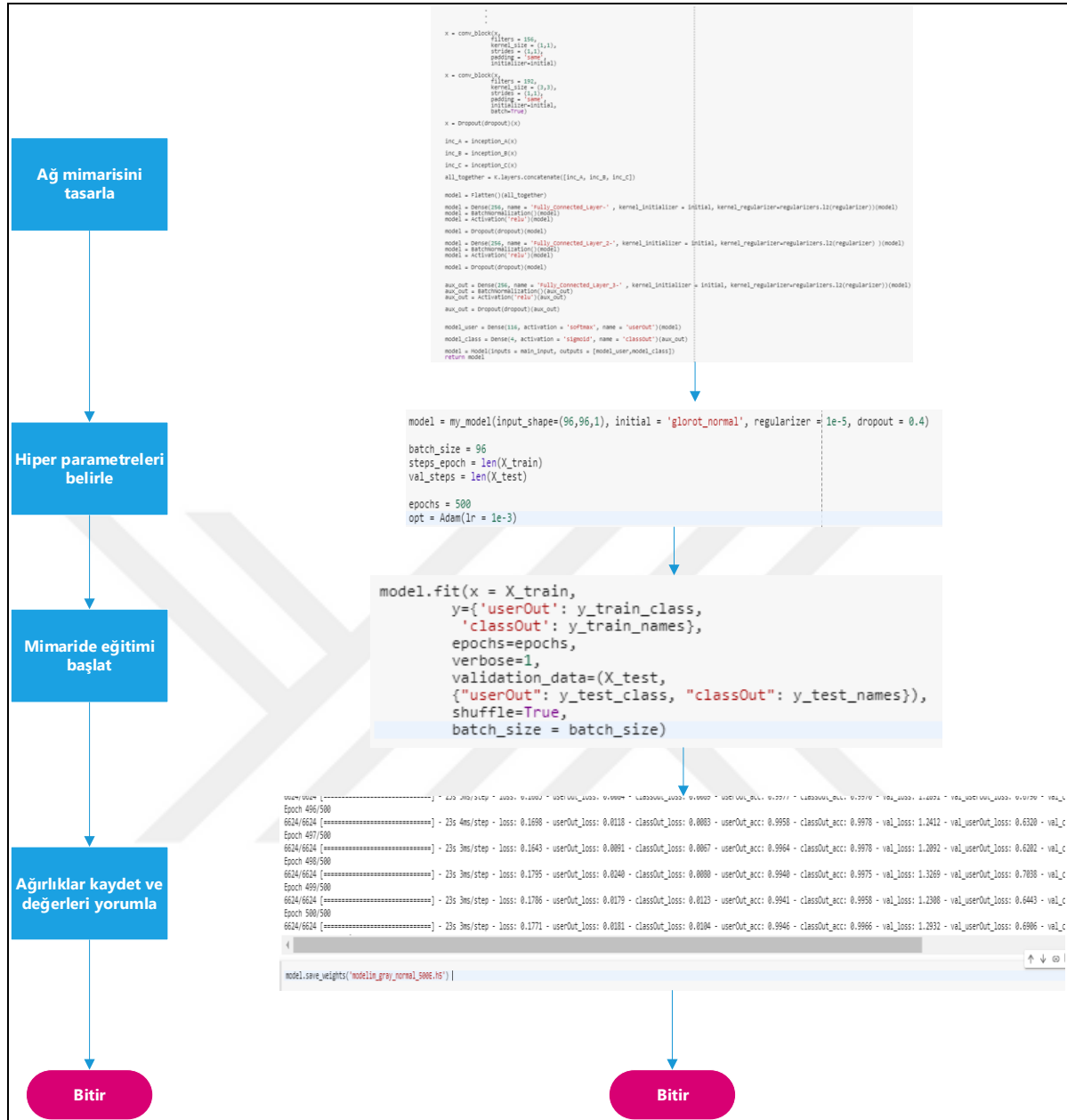
Önerilen ağ mimarisi diyagramı şekil 5.13’de verilmiştir.



Şekil 5.13. Önerilen ağ mimarisi diyagramı



Şekil 5.13. (devam) Önerilen ağ mimarisi diyagramı



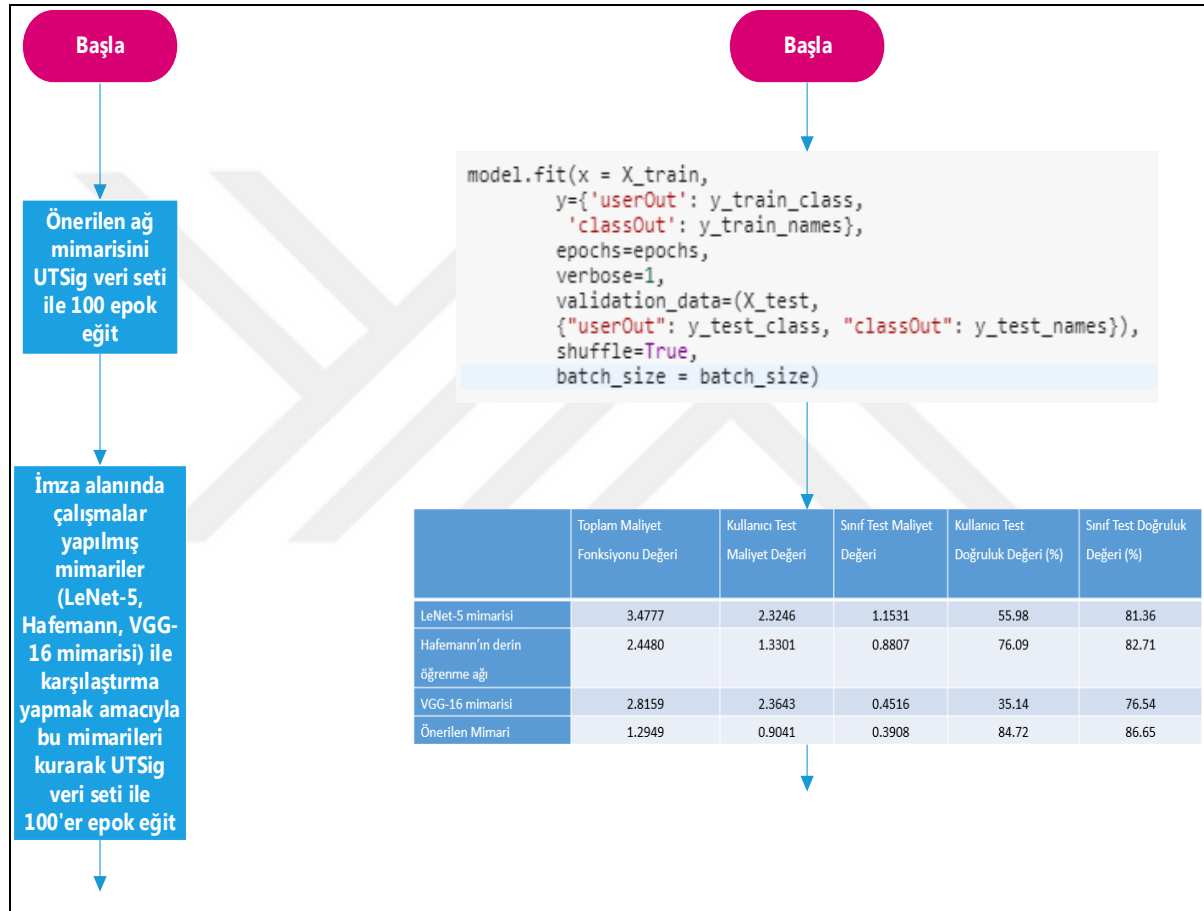
Şekil 5.13. (devam) Önerilen ağ mimarisi diyagramı



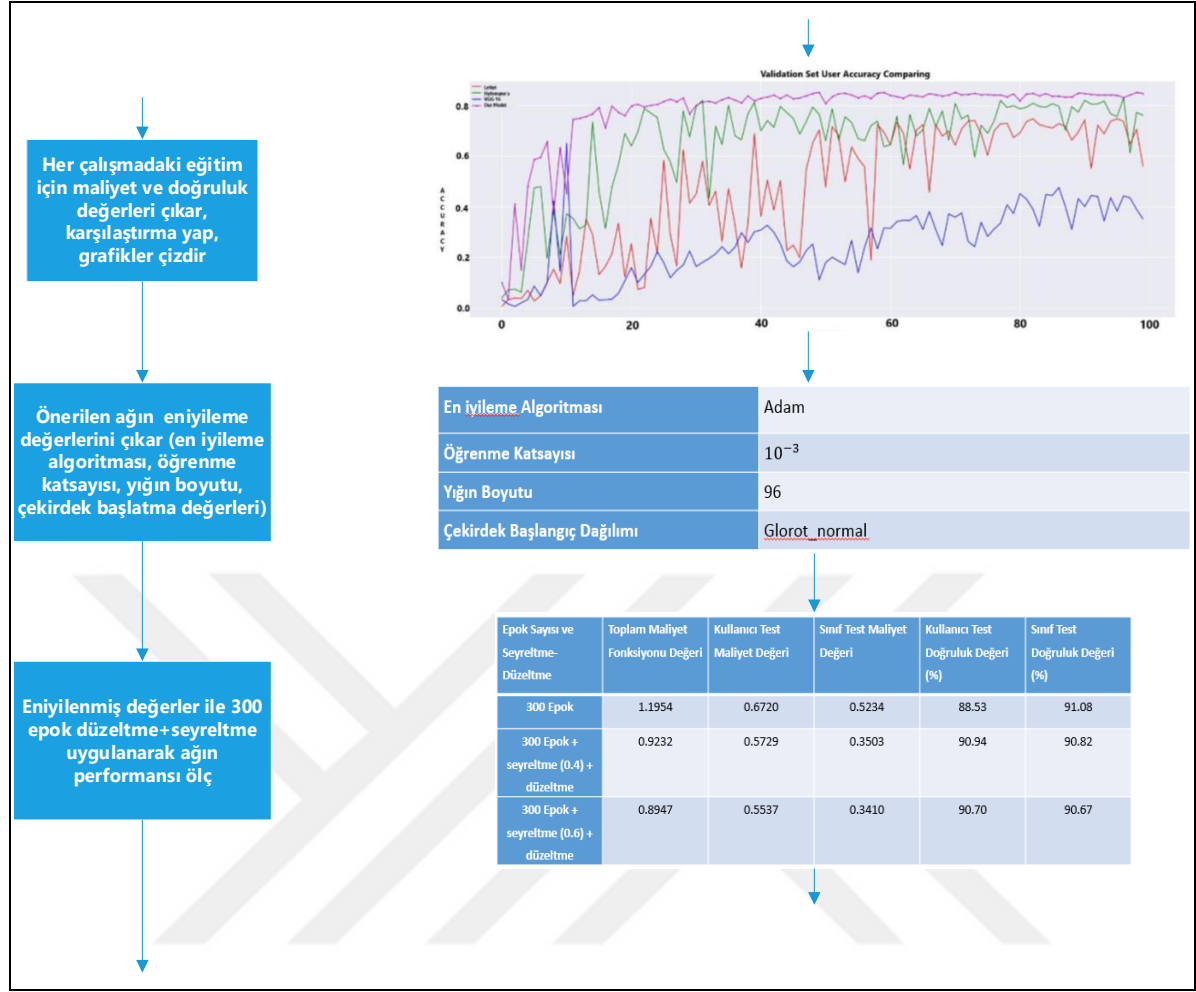
6. İMZA VERİ SETİNİN ÖNERİLEN AĞ MİMARİSİNE UYGULANMASI

Önerilen ağ mimarisi oluşturulduktan sonra uygulamalara geçilmiştir.

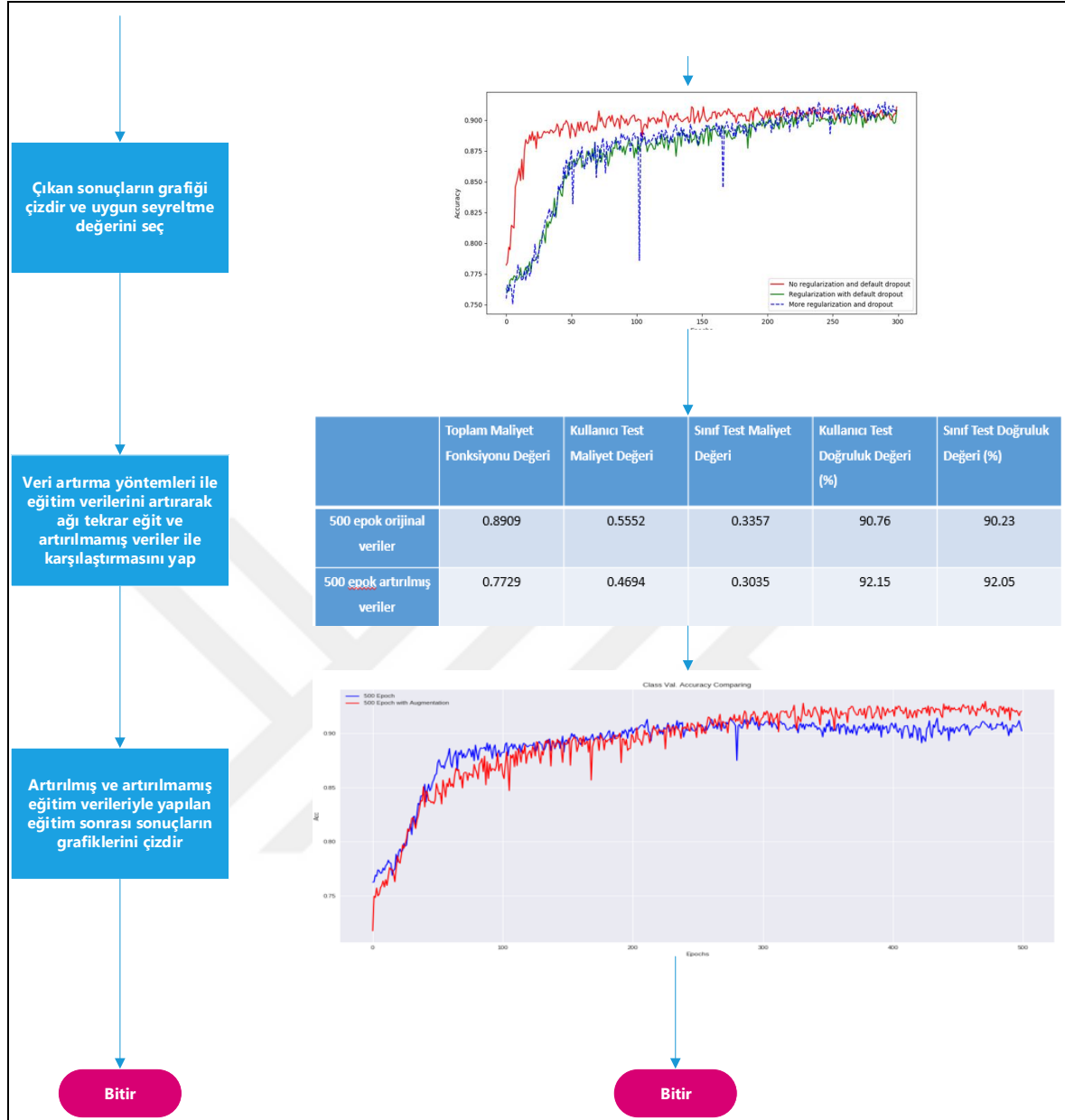
Uygulama olarak Şekil 6.1'deki gibi bir akış izlenmiş olup, detay sonuç ve grafikler alt başlıklarda verilmiştir.



Şekil 6.1. Önerilen ağ mimarisi üzerinde yapılan çalışmaların akış diyagramı



Şekil 6.1. (devam) Önerilen ağ mimarisi üzerinde yapılan çalışmaların akış diyagramı

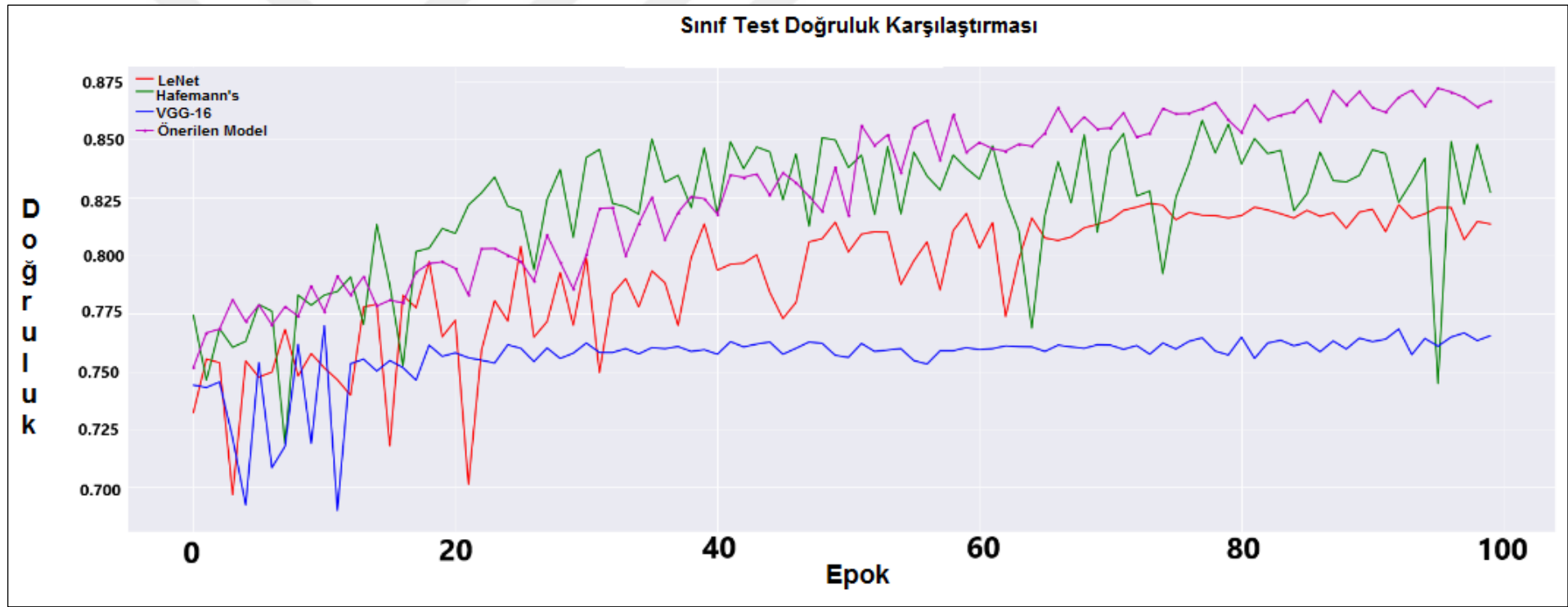


Şekil 6.1. (devam) Önerilen ağ mimarisi üzerinde yapılan çalışmaların akış diyagramı

6.1. Önerilen Ağ Mimarisi ile Diğer Mimarilerin Karşılaştırılması

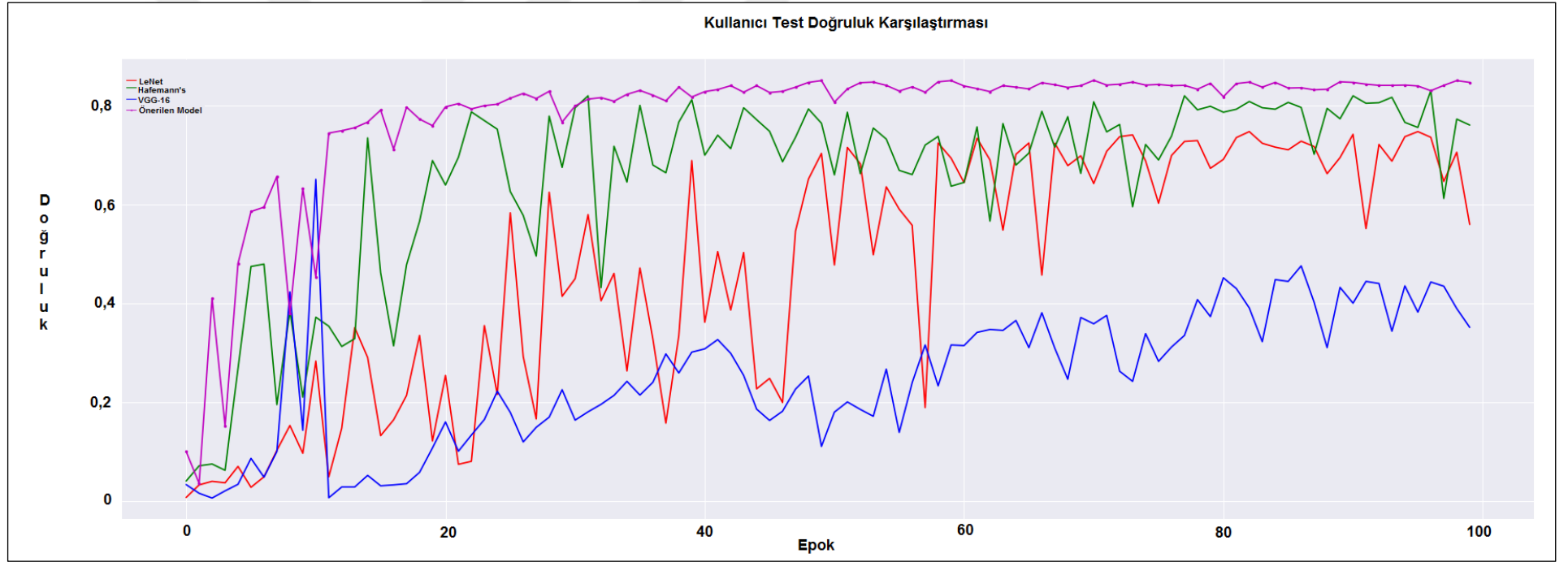
Önerilen ağ mimarisi diğer mimari ve çalışmalarla karşılaştırılmak amacıyla VGG-16 mimarisi, Hafemann çalışmasındaki derin öğrenme ağı ve LeNet-5 mimarileri ile karşılaştırılmıştır. Karşılaştırma yapılan mimarilerde doygunluk 100 epoka doğru tamamlandığı için her mimari için 100'er epok işlem yapılmıştır. Yapılan çalışmalarda, problemin önerilen ağdaki yaklaşımına göre diğer ağlarda iki çıkışlı hale getirilmiştir. Çıkışlardan birisi gerçek-sahte imza ayrımını yaparken diğer çıkış imzanın hangi kullanıcıya ait olduğunu tahmin etmektedir. Şekil 6.2'de söz edilen dört ağın karşılaştırılması grafiksel olarak verilmiştir.





Şekil 6.2. UTSig veri seti ile modellerin gerçek-sahte imza tahmin doğruluk değerleri

Şekil 6.2’de yer alan grafikte gerçek - sahte imza tahmini yapan çıkışa ait doğruluk değerlerini göstermektedir. Bu grafikte mor renk ile gösterilen önerilen ağ modelinin, yeşil renk ile gösterilen Hafemann’a ait modelin, kırmızı renk ile gösterilen Khalajzadeh’a ait modelin, mavi renk ile gösterilen ise VGG-16 ağ modeline ait doğruluk değerleridir. Aynı modellere ait aynı renkler ile Şekil 6.3’de modellerin hangi kullanıcıları doğru tahmin ettiğine dair sonuçlar yer almaktadır.



Şekil 6.3. UTSig veri seti ile modellerin kullanıcı tahmin doğruluk değerleri

Söz konusu tüm değerler 100 epok için alınmıştır. Sonuçlara ilişkin çizelge Çizelge 6.1’de yer almaktadır.

Çizelge 6.1. UTSig veri seti kullanılarak önerilen model ile karşılaştırılan mimarilerin doğruluk ve maliyet fonksiyonu değerleri

	Toplam Maliyet Fonksiyonu Değeri	Kullanıcı Test Maliyet Değeri	Sınıf Test Maliyet Değeri	Kullanıcı Test Doğruluk Değeri (%)	Sınıf Test Doğruluk Değeri (%)
LeNet-5 mimarisi	3,4777	2,3246	1,1531	55,98	81,36
Hafemann’ın derin öğrenme ağı	2,4480	1,3301	0,8807	76,09	82,71
VGG-16 mimarisi	2,8159	2,3643	0,4516	35,14	76,54
Önerilen Mimari	1,2949	0,9041	0,3908	84,72	86,65

Çizelge 6.1’de görüldüğü üzere önerilen metot 100 epok sonunda diğer mimarilere göre en iyi sonuçları vermiştir.

6.2. Önerilen Ağ Mimarisinin Optimize Edilmesi ile En iyi Sonuçlarının Elde Edilmesi

Derin öğrenmede, veriden öğrenen mimariler tasarlanırken, mimaride kullanılan algoritma ya da tekniklerin ne olması gerektiğine karar vermek mimarinin performansını en uygun seviyeye taşımak için önemlidir. Tüm bu algoritma ve tekniklerin mimari performansı için en uygununun bulunması ise eniyileme olarak adlandırılmaktadır. Seçilmesi gereken parametrelere hiper-parametre denilmektedir.

Önerilen ağ mimarisinde ilişkin 100 epok sonuçlarının alınmasının ardından ağ mimarisinin en iyilemesi yapılmıştır. En iyileme hiper parametrelerinin tespiti için tüm metotların 50 epok olarak çalıştırılması yeterli olmuştur. En iyileme yapılan adımlar şu şekilde sıralanmıştır; en iyileme algoritmasının seçilmesi, öğrenme katsayısının seçilmesi, yığın boyutunun seçilmesi, evrişim blokları ve tamamı bağlı katmanlar için çekirdek başlatma değerlerinin seçilmesi şeklindedir.

En iyileme algoritması

En iyileme algoritmasının seçimi için Keras API'nin sunduğu algoritmalar sırayla uygulanmıştır. Uygulanan en iyileme algoritmaları; “*RMSprop*”, “*Adam*”, “*Adadelta*”, “*Nadam*” ve “*SGD*” dir. Tüm en iyileme algoritmalarının her biri için 50 epok önerilen ağ mimarisinde çalıştırılmıştır. Aşağıda ilgili çizelgede alınan değerler yer almaktadır.

Çizelge 6.2. Önerilen mimari için en iyileme algoritmalarının seçiminde doğruluk ve maliyet fonksiyonu değerleri

En iyileme Algoritması	Toplam Maliyet Fonksiyonu Değeri	Kullanıcı Test Maliyet Değeri	Sınıf Test Maliyet Değeri	Kullanıcı Test Doğruluk Değeri (%)	Sınıf Test Doğruluk Değeri (%)
RMSprop	1,2726	0,8203	0,4523	87,86	89,78
Adam	1,0070	0,5806	0,4264	88,83	88,80
Adadelta	1,0964	0,6964	0,4000	86,17	87,44
Nadam	0,9817	0,5901	0,3916	88,59	88,63
SGD	1,0865	0,6525	0,4340	86,78	85,27

Çizelge 6.2’de yer alan değerlere bakılarak Adam en iyileme algoritması seçilmiştir. Her ne kadar en düşük maliyet fonksiyonu değerlerine sahip olmasa da Adam en iyileme algoritması ile elde edilen doğruluk daha yüksek olduğu için bu algoritma seçilmiştir.

Öğrenme katsayısı

Derin öğrenmede parametrelerin güncellenmesi geri yayılım işlemi ile yapılmaktadır. Geri yayılım işleminde bu güncelleme geriye doğru türev olarak farkın bulunması ve bulunan fark değerinin “öğrenme katsayısı” parametresiyle çarpılması, çıkan sonucun ağırlık değerlerinden çıkarılarak yeni ağırlık değerinin hesaplanmasıyla yapılmaktadır. Bu işlem esnasında kullanılan “öğrenme katsayısı” parametresi sabit değer olarak belirlenebilir, ya da adım adım değiştirilebilir.

Adam en iyileme yöntemi ile mimari öğrenme katsayısı seçimi en iyilemesine tabi tutulmuştur. Öğrenme katsayıları için seçilen değerler; 10^{-6} , 10^{-5} , 10^{-4} , 10^{-3} değerleridir. Bu değerler ile alınan sonuçlar çizelge 6.3’de verilmiştir.

Çizelge 6.3. Önerilen mimaride öğrenme katsayısının en iyilemesinde doğruluk ve maliyet fonksiyonu değerleri

Öğrenme Katsayısı Değeri	Toplam Maliyet Fonksiyonu Değeri	Kullanıcı Test Maliyet Değeri	Sınıf Test Maliyet Değeri	Kullanıcı Test Doğruluk Değeri (%)	Sınıf Test Doğruluk Değeri (%)
10^{-6}	4,0567	3,5052	0,5515	42,93	75,03
10^{-5}	2,6478	2,2165	0,4313	70,83	80,86
10^{-4}	1,2396	0,7403	0,4993	83,57	85,21
10^{-3}	0,9527	0,5603	0,3924	88,59	88,65

Çizelge 6.3'te yer alan değerlere bakılarak 10^{-3} öğrenme katsayısı seçilmiştir. Çizelge, tüm öğrenme katsayıları için mimari 50 epok sonuçlarını göstermektedir. Adam en iyileme algoritması adaptif bir metottur. Bu nedenle ilerleyen epoklarda öğrenme katsayısı kendiliğinden daha küçük değerlere inmektedir.

Yığın boyutu

Derin öğrenmede, veri setinde bulunan verilerin tamamını tek seferde işleyerek öğrenme, hem zaman hem de bellek açısından maliyetli bir iştir. Çünkü öğrenmenin her aşamasında geri yayılım ile eğitim hesaplaması yapılmaktadır, bu işlemin tüm verilere aynı anda yapıldığını düşünürsek bu işlem bellekte yetersiz alana sebep olacaktır. Bu sebeple veri seti küçük gruplara ayrılarak öğrenme bu gruplar üzerinden yapılmaktadır.

Mimaride seçilen yığın boyutu değeri veriyi hangi büyüklükte parçalar halinde ele alınacağını belirtmektedir.

Yüksek yığın boyutları genellikle hafıza sınırının aşılmasına neden olabilmektedir. Bu nedenle küçük yığın boyutlarına düşülmektedir ancak bu seferde çok küçük yığın boyutları toplam eğitim süresini oldukça artırabilmektedir. Önerilen mimari için 32, 64, 80, 96, 128 yığın boyutları denenmiştir. Bu değerlere ilişkin sonuçlar aşağıdaki çizelgede yer almaktadır. Çizelge 6.4'de önerilen ağ mimarisinin yığın boyutu en iyilemesinde doğruluk ve maliyet fonksiyonu ve 1 epok eğitim süresi verilmiştir.

Çizelge 6.4. Önerilen mimarinin yığın boyutu en iyilemesinde doğruluk ve maliyet fonksiyonu ve 1 epok eğitim süresi

Yığın Boyutu Değeri	Toplam Maliyet Fonksiyonu Değeri	Kullanıcı Test Maliyet Değeri	Sınıf Test Maliyet Değeri	Kullanıcı Test Doğruluk Değeri (%)	Sınıf Test Doğruluk Değeri (%)	Epok başına süre (saniye)
32	0,9987	0,5983	0,4004	88,04	89,40	40
64	1,0408	0,5739	0,4670	87,86	88,99	28
80	1,0362	0,5858	0,4504	88,95	89,15	26
96	0,9023	0,5363	0,3660	88,95	90,01	24
128	1,0233	0,5848	0,4386	87,44	89,24	22

Çizelge 6.4’te yer alan değerlere göre, 96 yığın boyutu hem süre bakımından uygun olduğu hem de doğruluk değerleri bakımından diğerlerine göre daha iyi olduğundan seçilmiştir. Epok başına sürenin kısa olması istenen bir durumdur.

Çekirdek Başlangıç Dağılımı

Bu adımların ardından evrişim işlemleri ve tamamı bağlı katman için çekirdek başlangıç değerlerinin seçilmesine ilişkin en iyilemeye geçilmiştir. Keras API, çekirdek başlangıç değerleri olarak çeşitli dağılımlara sahiptir. İstatistiksel birer dağılım olan çekirdek başlangıç değerleri bazı durumlarda mimarilerin daha iyi performans sergilemesine katkı sağlamaktadır. Mimari için en iyilemeye tabi tutulan çekirdek başlangıç dağılımları; “RandomNormal”, “RandomUniform”, “TruncatedNormal”, “lecun_uniform”, “glorot_normal”, “glorot_uniform”, “he_normal”, “lecun_normal”, “he_uniform” olmak üzere dokuz adettir. Bu dağılımlar içerisinde “RandomNormal” varsayılan değerdir. Yani herhangi bir dağılım belirtilmeyen evrişim işlemlerinde bu dağılım varsayılan olarak ilk değerleri belirlemektedir. Çekirdek başlangıç dağılımlarına ilişkin en iyileme sonuçları çizelge Çizelge 6.5’de verilmiştir.

Çizelge 6.5. Önerilen mimarinin çekirdek başlangıç değerlerinin en iyilemesinde doğruluk ve maliyet fonksiyonu değerleri

Çekirdek Başlangıç Dağılımı	Toplam Maliyet Fonksiyonu Değeri	Kullanıcı Test Maliyet Değeri	Sınıf Test Maliyet Değeri	Kullanıcı Test Doğruluk Değeri (%)	Sınıf Test Doğruluk Değeri (%)
RandomNormal	0,9221	0,5442	0,3779	88,77	89,37
RandomUniform	0,9934	0,5548	0,4386	88,47	89,30
TruncatedNormal	0,9555	0,5498	0,4057	88,47	89,46
Lecun_uniform	0,9059	0,5297	0,3763	88,35	90,35
Glorot_normal	0,9007	0,5434	0,3572	88,53	90,67
Glorot_uniform	0,9875	0,5508	0,4367	88,83	89,28
He_normal	0,9783	0,5397	0,4386	87,86	88,99
Lecun_normal	1,0134	0,5380	0,4753	88,29	88,95
He_uniform	0,9443	0,5502	0,3942	87,56	89,99

Çizelge 6.5'te yer alan değerlere göre “*Glorot_normal*” çekirdek başlangıç değeri seçilmiştir. Son en iyileme adımı olan çekirdek başlangıç değerinin de seçilmesinden sonra mimariye ilişkin en iyi değerleri üreten yöntemler Çizelge 6.6'da yer almaktadır.

Çizelge 6.6. Önerilen mimarinin tüm en iyileme değerleri

En iyileme algoritması	Adam
Öğrenme Katsayısı	10^{-3}
Yığın Boyutu	96
Çekirdek Başlangıç Dağılımı	Glorot_normal

Epok sayısı ve seyreltme-düzeltilme

Epok sayısı da mimaride en iyileme için kullanılan yöntemlerden biridir. Çizelge 6.6'da yer alan 50 epok sonucu ulaşılan en iyileme değerleri kullanılarak 300 epok ağ eğitilmiş ayrıca 300 epok 0,4-0,6 seyreltme, 10^{-5} düzeltme değerlerinde de eğitim sonuçları alınmıştır.

300 epok seçimin sebebi gerçek-sahte imza doğruluk değerlerinin denenen seyreltme ve düzeltme işlemlerinde son epoka kadar net bir şekilde belli olmaması sebebiyledir.

Çizelge 6.7'de Önerilen mimarinin 300 epokta normal hali ve seyreltme - düzeltme eklenmiş hali yer almaktadır.

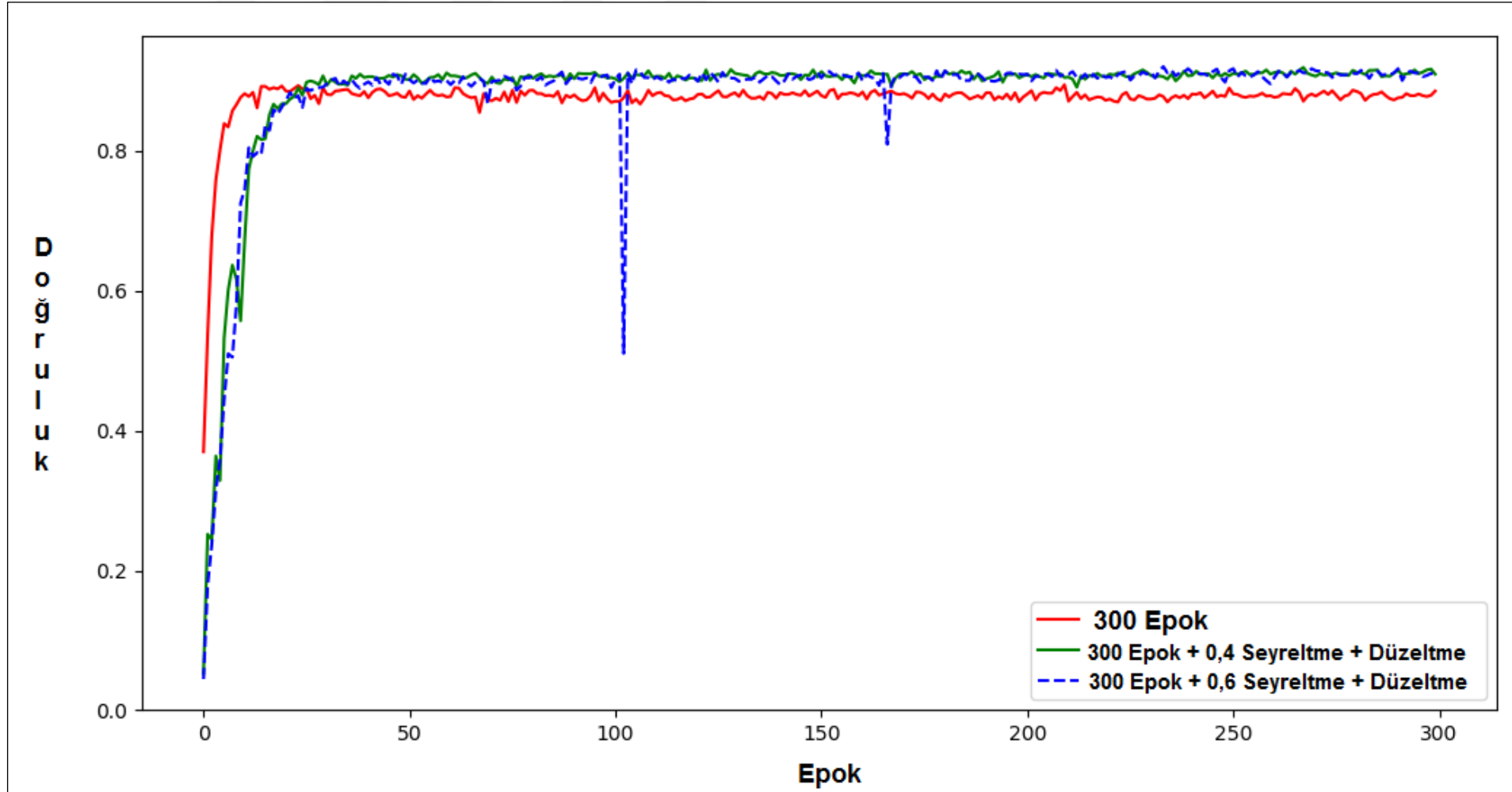
Çizelge 6.7. Önerilen mimarinin 300 epokta normal hali ve seyreltme - düzeltme eklenmiş hali

Epok Sayısı ve Seyreltme-Düzeltilme	Toplam Maliyet Fonksiyonu Değeri	Kullanıcı Test Maliyet Değeri	Sınıf Test Maliyet Değeri	Kullanıcı Test Doğruluk Değeri (%)	Sınıf Test Doğruluk Değeri (%)
300 Epok	1,1954	0,6720	0,5234	88,53	91,08
300 Epok + seyreltme (0,4) + düzeltme	0,9232	0,5729	0,3503	90,94	90,82
300 Epok + seyreltme (0,6) + düzeltme	0,8947	0,5537	0,3410	90,70	90,67

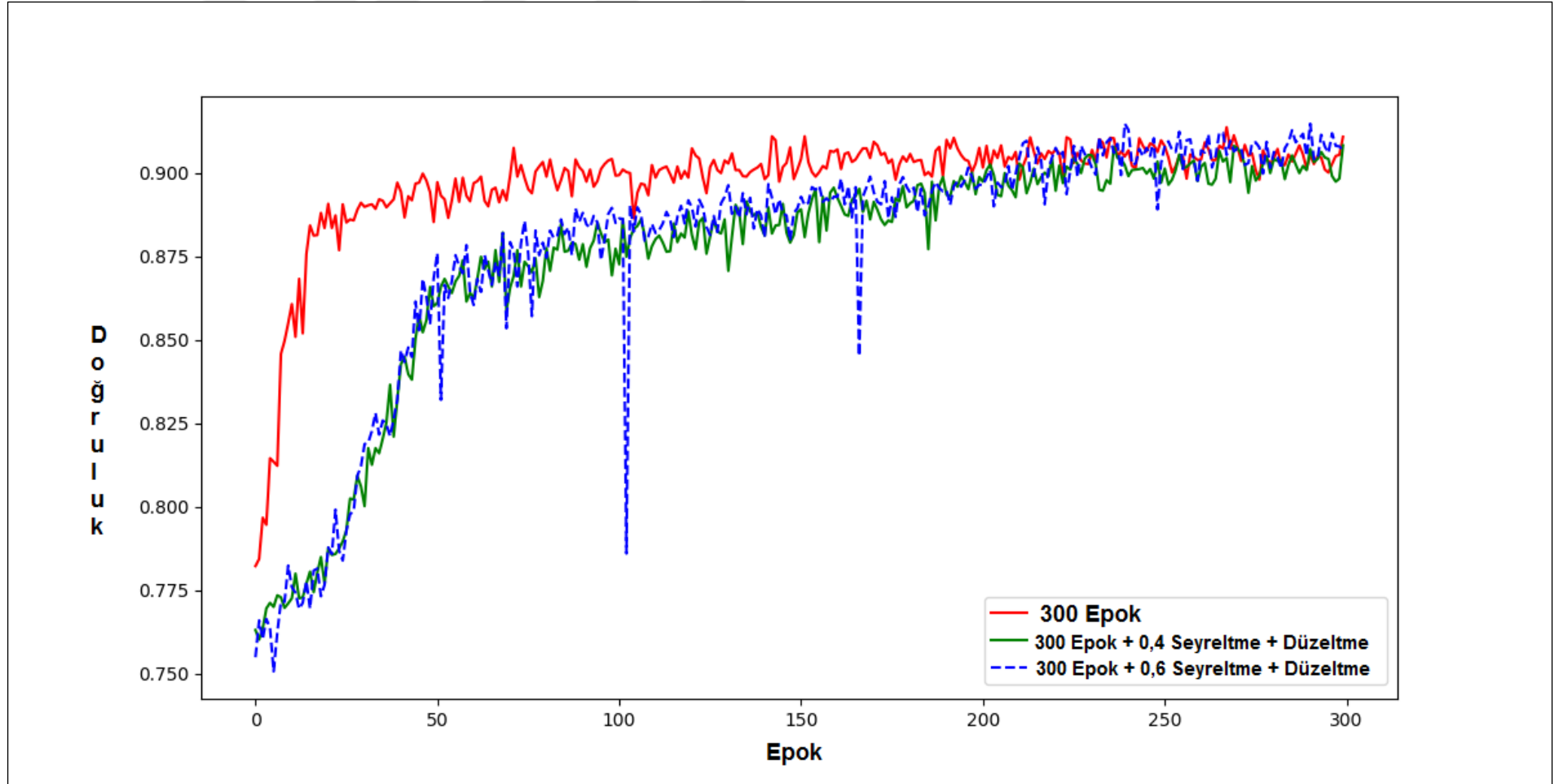
Çizelge 6.7.'de yer alan değerlere ilişkin görsel Şekil 6.4'te yer almaktadır.

Şekil 6.4'te mimarinin 300 epok sonunda kullanıcı tahminine ilişkin grafik gösterimi yer almaktadır. Şekilde kırmızı renk Çizelge 6.7'deki ilk değerlerden, yeşil renk Çizelge 6.7'deki seyreltme değerinin 0,4 olduğu değerlerden, mavi renk ise Çizelge 6.7'deki seyreltme değerinin 0,6 olduğu değerlerden elde edilmiştir.

Şekil 6.5'te mimarinin 300 epok sonunda gerçek-sahte imza ayrımına ilişkin elde edilen değerlerin grafik gösterimi yer almaktadır. Şekilde kırmızı renk, Çizelge 6.7'deki ilk değerlerden, yeşil renk Çizelge 6.7'deki seyreltme değerinin 0.4 olduğu değerlerden, mavi renk ise Çizelge 6.7'deki seyreltme değerinin 0.6 olduğu değerlerden elde edilmiştir.

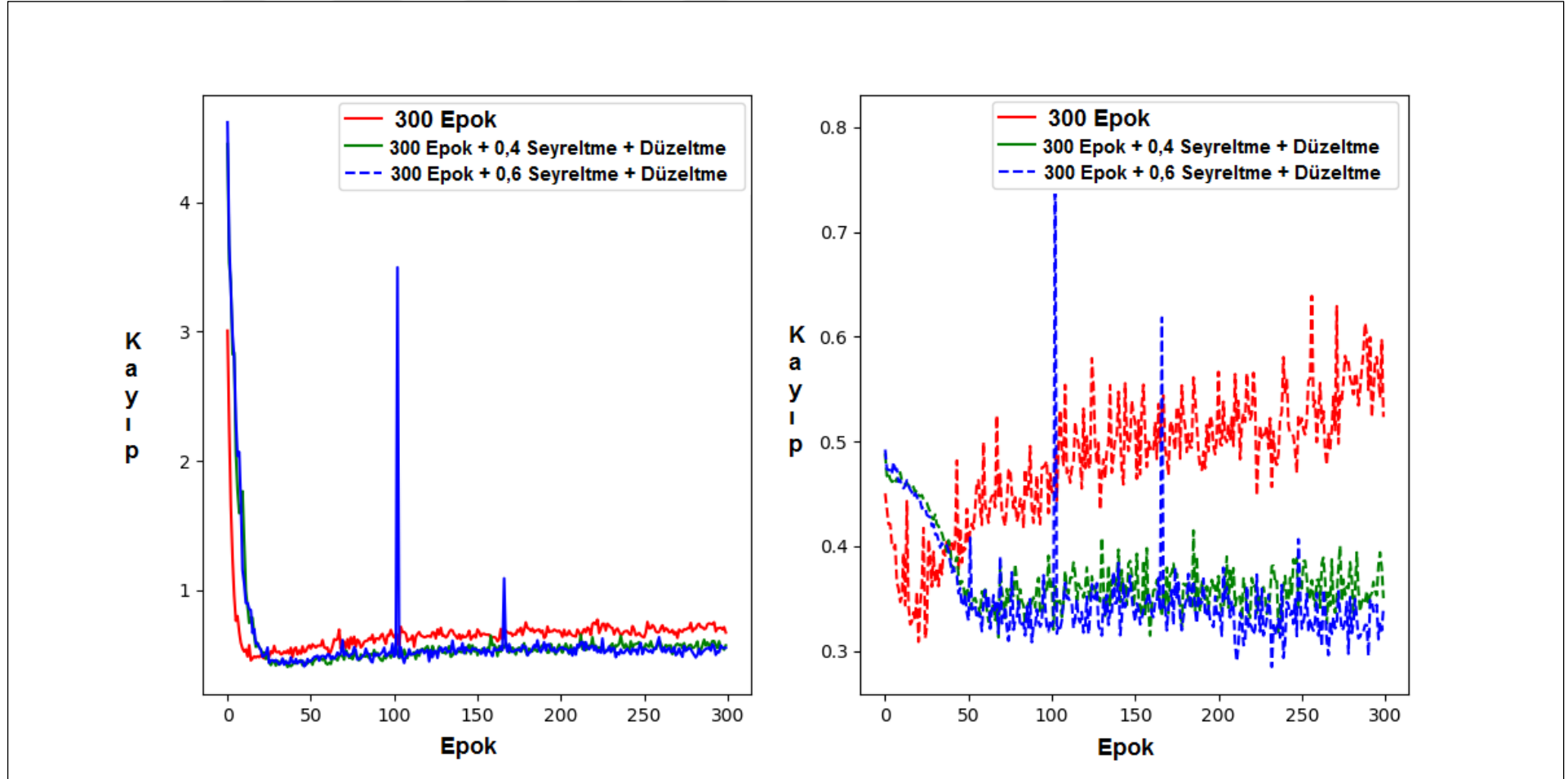


Şekil 6.4. 300 epok kullanıcı tahmini doğruluk sonuçları



Şekil 6.5. 300 epok gerçek-sahte imza ayırımına ilişkin doğruluk değerleri

Şekil 6.5 ve Şekil 6.6 incelendiğinde ulaşılan son değerlerin yakın olduğu ancak ara değerlerde eklenen düzeltme ve seyreltme değerlerinin etkisi açıkça görülmektedir. Derin öğrenme yöntemlerinde özellikle doğruluğun doğrusal olarak artması idealdir. Ancak bu durum her zaman sağlanamaz. Örneğin şekillerdeki kırmızı renk ile gösterilen mimari, çok hızlı şekilde yüksek doğruluk değerine ulaşmış ancak bu değer ilerleyen yaklaşık 200 epok boyunca neredeyse değişmediği görülmektedir. Şekillerdeki yeşil ve mavi renk ile gösterilen değerler birbirlerine oldukça yakındır. Ancak mavi renk ile gösterilen mimari için son doğruluk değeri daha düşüktür. Burada söylenilmesi gereken başka bir durum daha ortaya çıkmaktadır. Özellikle bu durumda yeşil renk ile gösterilen değerin daha iyi sonuçlar verdiği yanılıgısına düşülebilir. İki farklı yaklaşım arasından daha doğru olanı seçebilmek için maliyet fonksiyonu değerlerine bakmak gereklidir. Özellikle mavi renk ile gösterilen değerin maliyet fonksiyonu değeri daha düşüktür. Bu istenen bir durumdur. Maliyet fonksiyonlarına ilişkin görseller aşağıda yer almaktadır.



Şekil 6.6. Maliyet fonksiyonlarına ilişkin değerler

Şekil 6.6’da solda kullanıcı tahminine ilişkin, sağda gerçek-sahte imza tahminine ilişkin maliyet fonksiyonlarının görselleri yer almaktadır. Burada görülmektedir ki mavi değerlere ilişkin maliyet fonksiyonları daha düşüktür.

Maliyet fonksiyonlarına ilişkin değerlere bakıldığında mavi renk ile gösterilen mimari parametreleri seçilmiştir. Özellikle gerçek-sahte imza ayırımına ilişkin maliyet fonksiyonu çıktısı önemli bir etken olmuştur.

Mimariye ilişkin nihai parametreler, mavi renk ile gösterilen parametreler olarak belirlenmiştir. Şekil 6.6’da yer alan maliyet fonksiyonlarındaki sıçrama ve Şekil 6.6’da yer alan doğruluk değerlerindeki düşüş birbiriyle korelasyon içerisindedir. Bu durum zaten beklenen durumdur. Özellikle korelasyonun bozulduğu durumlarda ezberleme olduğu söylenebilir. Ezberlemenin var olduğunun en büyük kanıtı doğruluk değerlerinde sabitlik veya artışa göre, maliyet fonksiyonunda da artışın meydana gelmesidir. Özellikle Şekil 6.5 ve Şekil 6.6’da kırmızı renk ile gösterilen mimari parametrelerinde ezberleme meydana geldiği söylenebilir. Çünkü maliyet fonksiyonun artmasına rağmen doğruluk azalmamış ve yer yer artış göstermiştir. Yani aralarındaki korelasyon bozulmuştur.

Grafikler incelendiğinde 100. ve 160.epoklardan sonra ani sıçramalar olmuştur. Bu durum derin öğrenme problemlerinde yaşanabilen normal bir durumdur. İleri yayılımda atılan adımlar öğrenme katsayısı ile ilişkilidir. Öğrenme katsayısının adaptif olduğu durumda her epok sonrasında öğrenme katsayısı belli bir oranda düşmektedir. Bazı durumlarda öğrenme katsayısı düşse dahi global minimum ötesinde bir noktaya sıçrama gerçekleşebilir. Bu gibi durumlarda doğruluk oranı aniden düşebilir ancak takip eden epokta hata oranı tekrar hesaplandığı için bir önceki epokta atılan büyük öğrenme adımı geri alınarak daha küçük bir değere getirilmektedir. Takip eden epoklarda doğruluk ve maliyet değerlerinin tekrar beklendiği gibi yoluna girdiği Şekil 6.5 ve Şekil 6.6’da görülebilir.

Derin öğrenme problemlerinde ezberleme çok sık görülebilmektedir. Ezberleme sorununu çözmek için seyreltme ve/veya düzeltme işlemleri ile yapılabileceği gibi veri artırma yöntemlerine de başvurulabilir.

Veri Artırma

Ezberleme riskini ortadan kaldırmak ve başarı oranını artırmak için kullanılan veri artırma yöntemleri genel olarak görüntünün belirli oranda kaydırılması, döndürülmesi, görüntü matrisinin normalizasyon işlemine sokulması gibi yöntemlerden oluşmaktadır. Bunların dışında da veri artırma yöntemleri bulunmaktadır. Çalışmada UTSig veri setinde, veri artırma olarak iki işlem ele alınmıştır. Bunlardan ilki görüntülerin 90° döndürülmesidir, ikincisi ise minimum filtresinden geçirilmesidir. İmza görüntüleri sayfayı kaplayacak şekilde düzenlenmiş olduğu için 90° döndürülmesi imzada herhangi bir kısmın kaybolmasına sebep olmamaktadır, örneğin 45° döndürülerek veri artırımı yapılsaydı köşelerden imza görüntülerinde kayıplar oluşacaktı. Bu yüzden döndürme işlemi 90° olarak seçilmiştir. Diğer veri artırma yöntemi minimum filtresinin seçilme sebebi ise minimum filtresi görüntü işleme de kenar bulma işlemlerinde kullanılmaktadır. Veri setindeki bazı verilerin bu filtreden geçirilerek kenarları belirginleştirilmiştir. Bu durum öğrenmeye pozitif yönde katkı sağlamıştır. Ancak bu işlemler söz konusu bir kullanıcıya ait yirmi yedi adet gerçek imzanın her birine uygulanmamıştır. İlgili kullanıcıya ait verilerden rastgele olarak seçilen on adet imzaya veri artırma yöntemleri uygulanmıştır. Artırılmış veriler ile mimari 500 epok eğitilmiştir. Özellikle artırılmış veriler ve orijinal veriler arasındaki farkın anlaşılması için orijinal veriler ile de 500 epok sonuçları alınmıştır.

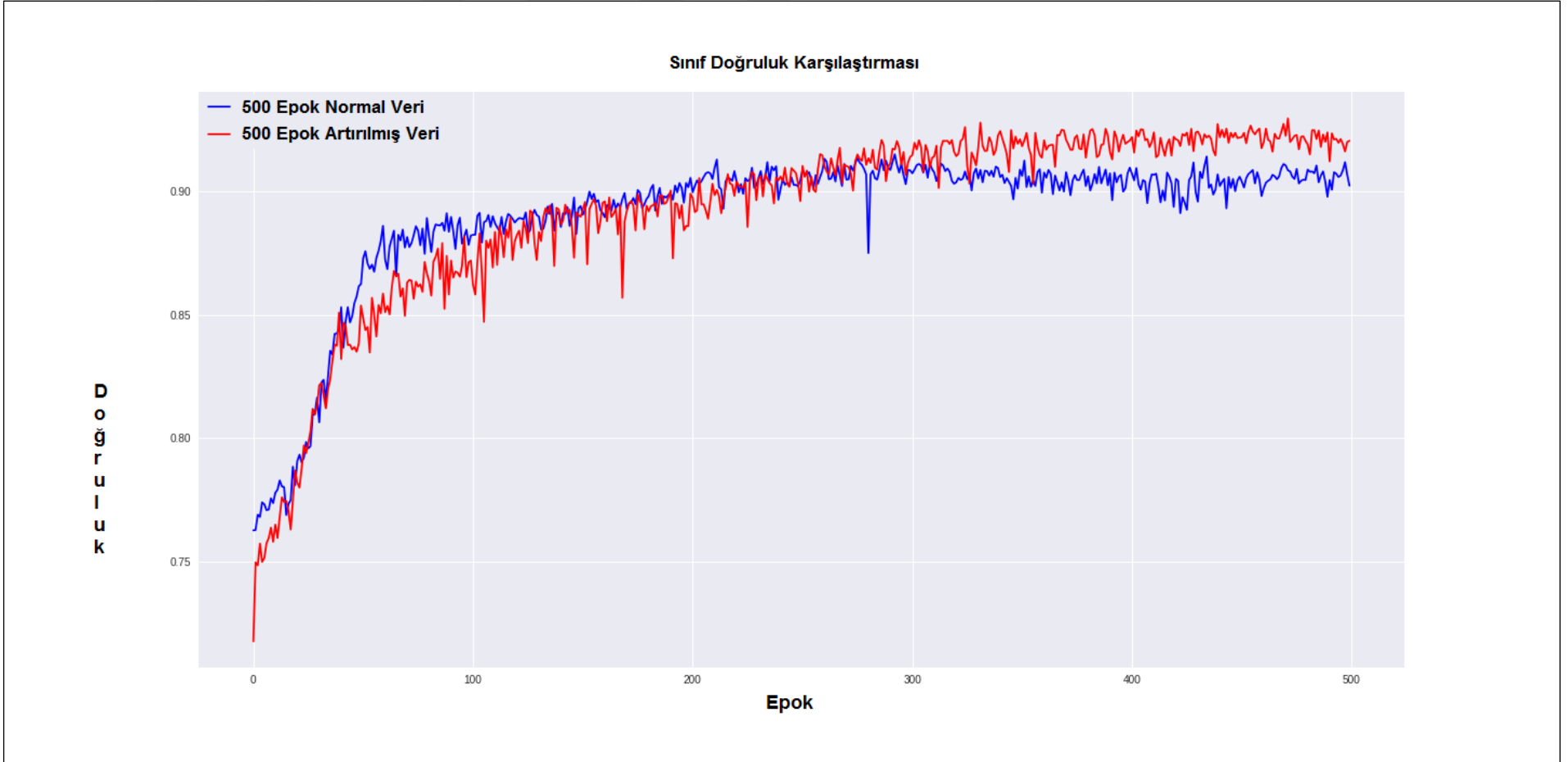
Özellikle 500 epok eğitim yapılmasının sebebi ise 300 epoktan sonra hala gerçek-sahte imza doğruluğun artırılmış veriler ile artma eğilimi içinde olmasıdır. Artırılmış veriler ile alınan doğruluk sonuçları 300 epok sonrası da artmıştır. Bu yüzden epok sayısı 500'e çıkarılmış böylece epok sayısının da ağırlık eğitiminde doğruluk üzerindeki etkisi bir kez daha görülmüştür. Şekil 6.7 üzerinde bu etki açıkça görülebilmektedir.

Çizelge 6.8. Önerilen mimaride 500 epokta orijinal ve artırılmış veriler ile alınan doğruluk ve maliyet fonksiyonu değerleri

	Toplam Maliyet Fonksiyonu Değeri	Kullanıcı Test Maliyet Değeri	Sınıf Test Maliyet Değeri	Kullanıcı Test Doğruluk Değeri (%)	Sınıf Test Doğruluk Değeri (%)
500 epok orijinal veriler	0,8909	0,5552	0,3357	90,76	90,23
500 epok artırılmış veriler	0,7729	0,4694	0,3035	92,15	92,05

Çizelge 6.8’de yer alan değere ilişkin artırılmış verilerin doğruluk değerlerini artırdığı ve maliyet fonksiyonu değerlerini düşürdüğü görülmektedir. Çalışmada veri artırma sonucunda elde edilen bir diğer kazanım ise ara değerlerde daha önce meydana gelen maliyet fonksiyonundaki sıçramaların ve doğruluk değerlerindeki ani düşüşlerin artık olmayışıdır.





Şekil 6.7. 500 epok gerçek-sahte imza tahminine ilişkin doğruluk değerleri

Şekil 6.7’de gerçek – sahte imza ayrımı yapan çıkışa ilişkin doğruluk değerlerinin görseli yer almaktadır. Şekilde kırmızı renk ile gösterilen artırılmış veriler, mavi renk ile gösterilen orijinal verilerdir.



Şekil 6.8. 500 epok kullanıcı tahminine ilişkin doğruluk değerleri

Şekil 6.8’de kullanıcıların doğru tahmin edilmesine ilişkin görsel yer almaktadır. Özellikle daha önce meydana gelen doğruluk değerindeki ani düşüş 500 epok orijinal veriler ile yapılan eğitimde de görülmektedir. Mavi renk ile gösterilen orijinal veriler ile eğitimde özellikle ilk epoklar içerisinde meydana gelen ani düşüş görülmektedir. Ancak kırmızı renk ile gösterilen artırılmış veriler ile yapılan 500 epok eğitimde yer ani düşüş görülmemektedir. Benzer durumlar maliyet fonksiyonları için de geçerlidir.



7. SONUÇ VE ÖNERİLER

Bu tez çalışmasında derin öğrenme tekniklerinden olan evrişimli sinir ağları ile çalışılmış, VGG-16 ve GoogleNet mimarilerinden de yararlanılarak yeni bir hibrid ağ mimarisi önerilmiştir.

Çalışmada önerilen hibrid ağ için İran kökenli UTSig veri seti kullanılmıştır. Veri setinin %80'i ile ağ eğitilmiş ve öznitelikler elde edilmiş, %20'lik kısmı ile de ağın başarı oranı test edilmiştir. Önerilen ağ mimarisinin diğer mimarilerdeki başarı oranı ile karşılaştırılması amacıyla aynı veri seti 100 epok olarak LeNet-5, Hafemann'ın derin ağ mimarisi ve VGG-16 ağ mimarilerine uygulanmıştır. Önerilen ağ mimarisi bu karşılaştırmada en yakın rakibine göre kullanıcı tahmininde %8 civarı, doğru-sahte sınıf tahmininde ise %4 civarı daha iyi sonuç vermiştir.

Önerilen ağ mimarisinden en iyi sonuçları elde edebilmek için modelin en iyilemesi yapılmış, bunun için hiper parametreler aynı epok sayısında sırayla denenip en iyi hiper parametrenin seçimi yapılmıştır. Bu amaçla en iyileme algoritması, öğrenme katsayısı, yığın boyutu, çekirdek başlangıç dağılımı parametreleri üzerinde testler yapılmıştır. Bu testlerin sonuçlarına göre *Adam* en iyileme algoritması, 10^{-3} öğrenme katsayısı, 96 yığın boyutu *Glorot_Normal* çekirdek başlangıç dağılımı seçilmiştir. Her parametre için çıkan sonuçlar kaydedilip bunların karşılaştırmasının verilmesi ile hiper parametre seçiminin derin öğrenme mimarilerinde önemli olduğu ortaya konulmuştur.

Derin öğrenme mimarileri için literatürde doğruluk performanslarını artırmada kullanılan seyreltme, düzeltme, veri artırma teknikleri kullanılmış ve ağın başarı oranında artış görülmüştür. Seyreltme ve düzeltme için en iyi sonuçlar 0,4 ve 10^{-5} olarak elde edilmiştir. Veri setinin artırılması amacıyla 8280 olan toplam imza görüntü sayısının bir kısmı için imzanın orijinalliğini bozmayacak şekilde işlemler uygulanarak veri seti 12420'ye çıkartılmıştır. Bu şekilde ağın eğitilmesi ile yapılan testlerde başarı oranının arttığı gözlemlenmiştir. Artırılmamış veriler ile ağ test edildiğinde imzanın hangi kullanıcıya ait olduğuna ilişkin %90,76 başarı elde edilirken imzanın doğru-sahte ayrımına ilişkin %90,23 başarı elde edilmiştir. Aynı test artırılmış veriler ile yapıldığında bu sonuçların %92,15 ve %92,05 olarak iyileştiği görülmüştür. Böylece yapılan karşılaştırmalar ile derin öğrenme sistemlerinde veri artırma, düzeltme ve seyreltme katmanının etkisinin göz ardı edilemeyecek kadar fazla olduğu gözlemlenmiştir.

Sonuç olarak önerilen ağ mimarisi geleneksel imza doğrulama sistemlerinden farklı olarak hem hangi imzanın hangi kullanıcıya ait olduğuna dair, hem de imzanın doğru ya da sahte olduğuna dair çıktı üretmektedir. Önerilen ağ mimarisi ile yapılan imza doğrulama hukuki süreçlerde bilirkişiye yardımcı olabilecek bir çıktı üretmektedir.

Ağ modelinde özellikle veri setine bağımlılığın araştırılması için farklı veri setleri üzerinde de yöntemin denenmesi önerilmektedir. Mevcut veri setinde bulunan dört farklı sınıf yerine gerçek ve sahte imza sınıfı olmak üzere toplam iki sınıflı sonuçların alınması önerilen başka bir çalışma alanıdır.



KAYNAKLAR

1. Atalay, M , Çelik, E . (2017). Büyük veri analizinde yapay zekâ ve makine öğrenmesi uygulamaları. *Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 9(22), 155-172.
2. Yardımcı, T. (2011), *Makine Öğrenmesi Teknikleri İle Rss Besleme Yönetimi*, Yüksek Lisans Tezi, Gazi Üniversitesi Bilişim Enstitüsü, Ankara.
3. Öztemel, E. (2003), *Yapay Sinir Ağları*, İstanbul: Papatya Yayıncılık, 13-15.
4. Yalçın, N., Gürbüz F. (2015). Biyometrik Güvenlik Sistemlerinin İncelenmesi. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 3, 398-413.
5. Aydoğdu E., Ataç Y. (2011). İmza Sahteciliğinin Türleri, Tespiti ve Önlenmesi, *Polis Bilimleri Dergisi*, 13(2), 49-71.
6. İnternet: Başer, F. Google Colab free GPU tutorial, URL: <https://medium.com/deep-learning-turkey/googlecolab-free-gpu-tutorial-e113627b9f5d>, Son Erişim Tarihi: 01.03.2019.
7. Şenol, C. (2004), *Standart ve Hibrid Yapılar Kullanarak Yapay Sinir Ağları ile İmza Tanıma*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
8. Başer, E., Şatır, E. (2017). Signature Verification Using Artificial Neural Network. *International Advanced Researches & Engineering Congress*, 1309.
9. Erten, M. (2012). Sınıflandırma Yöntemleri Kullanılarak İmza Biyometriğine Dayalı kişi Tanıma, Yüksek Lisans Tezi, T.C. Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.
10. Baykal, B., Aktaş, T., Yıldız, O., (2017). Makine Öğrenmesi Yöntemi ile Otomatik Çevrimdışı İmza Tanıma ve Doğrulama Sistemi, *International Artificial Intelligence and Data Processing Symposium (IDAP)*, Malatya.
11. Kaymaz, S. (2010). Çevrimdışı İmza Tanıma, Yüksek Lisans Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

12. Taylan Das, M., L Canan Dulger, L. (2007). Off Line Signature Verification with PS0-NN Algorithm. *Twenty Second International Symposium on Computer and Information Sciences*, 7(9),1-6.
13. Fazli, S., Pouyan, S., Moghaddam, M. (2014). High-Performance Signature Recognition Method using SVM. *International Journal of advanced studies in Computer Science and Engineering IJASCSE*, 3(11).
14. Randhawa, M., Sharma. K, (2012). Off-line Signature Verification based on Hu's Moment Invariants and Zone Features using Support Vector Machine. *International Journal of Latest Trends in Engineering and Technology (IJLTET)*,16-23.
15. Radhika, K., Gopika, S. (2014). Online and Offline Signature Verification: A Combined Approach. *International Conference on Information and Communication Technologies(ICICT)*,1593-1600.
16. Kumar, P., Singh, S., Garg, A. ve Prabhat N. (2013). Hand Written Signature Recognition & Verification using Neural Network. *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*.
17. Al-Omari, M., Sheikh A.N. H. and Omar K. (2011). State-of-the-Art in offline signature verification system. *International Conference on Pattern Analysis and Intelligent Robotics*, Malezya, 59-64.
18. Kisku, D. R., Gupta, P. and Sing, J. K. (2010). Offline Signature Identification by Fusion of Multiple Classifiers Using Statistical Learning Theory. *International Journal of Security and Its Applications*,35-40.
19. Wai, H. and Aung, S. L. (2014). Off-Line Signature Verification using Thresholding Technique. *International Journal of Scientific Engineering and Technology Research*, 3(35),7032-7035.
20. Kudlacik, P., Porwik, P. (2014). A new approach to signature recognition using the fuzzy method. *Pattern Analysis and Application*, 17-451
21. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition. *IEEE*, 86(11), 2278-2324.

22. Khalajzadeh, H., Mansouri, M., Teshnehlal, M. (2012). Persian Signature Verification using Convolutional Neural Networks. *International Journal of Engineering Research and Technology (IJERT)*, 1(2), 2278-0181.
23. Hafemann L.G., Sabourin, R., Oliveira, L.S. (2017). Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition*, 70, 163-176.
24. Fierrez-Aguilar, J., Alonso-Hermira, N., Moreno-Marquez, G., Ortega-Garcia, J. (2004). An off-line signature verification system based on fusion of local and global information. *Springer*, 295–306.
25. Ferrer, M., Alonso, J.B., Travieso, C. (2005). Offline geometric parameters for automatic signature verification using fixed-point arithmetic. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(6), 993–7.
26. Vargas, F., Ferrer, M.M.A., Travieso, C. (2007). Off-line Handwritten Signature GPDS-960 Corpus, *Proc. of the 9th ICDAR*, 764–768.
27. Blankers, V.L., Heuvel, C., Franke, K.Y., (2009). Icdar 2009 signature verification. *ICDAR'09. 10th International Conference on' (IEEE, 2009)*, 1403–1407.
28. Liwicki, M., Heuvel, C.E. Van Den, Found, B., Malik, M.I. (2010). Forensic Signature Verification Competition 4NSigComp2010 - Detection of Simulated and Disguised Signatures. *12th International Conference on Frontiers in Handwriting Recognition*, 715–720.
29. Liwicki, M., Malik, M.I., van den Heuvel, C.E., (2011). Forensic signature verification competition 4nsigcomp2010 - detection of simulated and disguised signatures, *2011 International Conference on (Ieee, 2011)*, 1480–1484.
30. Liwicki, M., Malik, M.I., Alewijnse, L., van den Heuvel, E., Found, B. (2012) 'ICFHR 2012 Competition on Automatic Forensic Signature Verification. *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on (Ieee, 2012)*, 823–828.
31. Malik, M.I., Liwicki, M., Alewijnse, L., Ohyama, W., Blumenstein, M., Found, B. (2013) 'ICDAR 2013 Competitions on Signature Verification and Writer Identification

- for On-and Offline Skilled Forgeries. *2013 12th International Conference on (Ieee, 2013)*,1477–1483.
32. Soleimani, A., Fouladi, K., Araabi, B.N., UTSig (2016), A Persian Offline Signature Dataset. *IET Biometrics*, 6, 1-8.
 33. Allahverdi, N., (2002). *Uzman Sistemler- Bir Yapay Zeka Uygulaması*, Ankara: Atlas Yayıncılık, 1-70.
 34. İnternet: Tozkan, S. Yapay Sinir Ağları. Bitirme Ödevi, Fırat Üniversitesi Mühendislik Fakültesi, URL: <http://www.hasanbalik.com/projeler/bitirme/24.pdf>, Son Erişim Tarihi: 11.12.2018.
 35. İnternet : Yapay Zekanın Tarihçesi ve Gelişim Süreci, *Medium blog*, URL: <https://medium.com/türkiye/yapay-zekanın-tarihçesi-ve-gelişim-süreci-cb4c73deb01d>, Son Erişim Tarihi: 14.01.2019.
 36. İnternet : Gartner Hype Curve, URL: <https://view.ceros.com/gartner/digital-marketing-and-advertising-hype-cycle/p/2>, Son Erişim Tarihi: 14.02.2019.
 37. Tezbaşaran, E., Gelbal, S. (2018). Temel Bileşenler Analizi ve Yapay Sinir Ağı Modellerinin Ölçek Geliştirme Sürecinde Kullanılabilirliğinin İncelenmesi. *Mersin Üniversitesi Eğitim Fakültesi Dergisi*, 14 (1), 225-252.
 38. McCulloch, W.S., Pitts, W., (1943). A logical calculus of ideas imminent in nervous activity. *Bulletin of Methematics Biophysics*, 5, 115-133.
 39. Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Reviev*,65(6), 386-408.
 40. Hornik, K., Stinchcombe, M. and White, H. (1989), Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5), 359-366.
 41. Rumelhart, E., Hinton, E., Williams, R. (1986). Learning representations by back-propagating errors. *Nature*,323.
 42. Nadezhda, A., Andreeva, V. (2015). Global minimum search via annealing: Nanoscale gold clusters. *Chemical Physics Letters*,622, 75-79.
 43. İnternet: CS231n Convolutional Neural Networks for Visual Recognition, URL: <http://cs231n.github.io/neural-networks-1/#actfun>, Son Erişim Tarihi: 22.01.2019.

44. Clevert, D., Unterthiner, T., Hochreiter, S. (2016). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *Johannes Kepler University Institute of Bioinformatics, Austria*.
45. Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S. (2017). Self-Normalizing Neural Networks. *LIT AI Lab. & Institute of Bioinformatics*.
46. He, K., Zhang, X., Ren, S., Sun, J. (2015). Delving Deep into Rectifiers: Suprass,ng Human-Level Performance on Imagenet Classification. *Microsoft Research*.
47. Gao, B., Pavel, L., (2018). On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning,1-10.
48. Kurt, F., (2018). Evriřimli Sinir ađlarında Hiper Parametrelerin Etkisinin İncelenmesi, Yüksek Lisans Tezi, *Hacettepe Üniversitesi Bilgisayar Mühendisliđi Ana Bilim Dalı, Ankara*.
49. Smith, W. (2016). Chapter 13. Convolution, *California Technical Publishing*.
50. Internet: Karpathy, A. (2015), The Unreasonable Effectiveness of Recurrent Neural Networks, <http://karpathy.github.io/2015/05/21/rnneffectiveness/>
51. Internet: Kızrak, A. Derine Daha Derine: Evriřimli Sinir Ađları, URL: <https://medium.com/deep-learning-turkiye/deri%CC%87ne-daha-deri%CC%87ne-evri%C5%9Fimli-sinir-a%C4%9Flar%C4%B1-2813a2c8b2a9>, Son Eriřim Tarihi: 14.03.2019.
52. Krizhevsky, A., Sutskever, I., Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks, *NIPS*.
53. Simonyan, K., Zisserman, A. (2015). Very Deep Convolutional Networks for Large Scale Image Recognition, *Visual Geometry Group, Deparment of Engineering Science at Oxford University*.
54. İnternet: Visual Geometry Group, *Department of Engineering Science University of Oxford*, URL: http://www.robots.ox.ac.uk/~vgg/research/very_deep/, Son Eriřim Tarihi: 14.03.2019.
55. İnternet: VGG in TensorFlow, URL: <https://www.cs.toronto.edu/~frossard/post/vgg16/>, Son Eriřim Tarihi: 21.02.2019.

56. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2014). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1-9.
57. He, K., Zhang, X., Ren, S., Sun, J., (2015). Deep Residual Learning for Image Recognition, *Microsoft Research*.
58. İnternet: The ImageNet ILSVRC challenge results, URL: https://www.researchgate.net/figure/The-ImageNet-ILSVRC-challenge-results-top-5-error-from-2010-to-2015-AlexNet_fig1_319271964, Son Erişim Tarihi: 11.03.2019.



ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, Adı : TURAN, Bilal
 Uyruğu : T.C.
 Doğum tarihi ve yeri : 01.06.1987, Gaziantep
 Medeni hali : Evli
 Telefon : +90 (507) 605 62 60
 E-mail : bilal.turan@gazi.edu.tr



Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek Lisans	Gazi Üniversitesi / Bilişim Enstitüsü / Adli Bilişim A.B.D	2019
Lisans	Karadeniz Teknik Üniversitesi/ Elektrik-Elektronik Mühendisliği	2011

İş Deneyimi

Yıl	Yer	Görev
2018 - Halen	Türk Telekomünikasyon	Servis Yönetimi ve Tahsilat Opr. Kıdemli Uzmanı
2015 - 2018	Türk Telekomünikasyon	Sabit-Genişbant Hatlar Gelir Yönetimi Uyg. Opr. Uzmanı
2014 - 2015	Türk Telekomünikasyon	İş Zekası ve Kurumsal Uygulamalar Uzmanı
2013 - 2014	Türk Telekomünikasyon	Gelir Yönetimi Uygulamaları Operasyon Uzman Yardımcısı
2012 - 2013	Etiya Bilgi Teknolojileri A.Ş	Uygulama Destek Uzmanı

Yabancı Dil

İngilizce

Hobiler

Bilgisayar ve bilişim teknolojileri, Gitar çalmak ve müzik dinlemek



GAZİLİ OLMAK AYRICALIKTIR..