T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME TEMELLİ YENİ NESİL GÜVENLİK DUVARININ TASARLANMASI

Kenan BAYSAL

DOKTORA TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Tez Danışmanı: Dr. Öğr. Üyesi Deniz TAŞKIN

EDİRNE 2022

KENAN BAYSAL'ın hazırladığı "**DERİN ÖĞRENME TEMELLİ YENİ NESİL GÜVENLİK DUVARININ TASARLANMASI**" başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından **Bilgisayar Mühendisliği** Anabilim Dalında bir **Doktora tezi** olarak kabul edilmiştir.

Jüri Üyeleri :	İmza
Dr. Öğr. Üyesi Deniz TAŞKIN	……
Dr. Öğr. Üyesi Tarık YERLİKAYA	……
Dr. Öğr. Üyesi Berrin YILMAZ	……
Dr. Öğr. Üyesi Murat Olcay ÖZCAN	……
Dr. Öğr. Üyesi Selçuk YAZAR	……

Tez Savunma Tarihi: 10 / 02 / 2022

Bu tezin Doktora tezi olarak gerekli şartları sağladığını onaylarım.

İmza

Dr. Öğr. Üyesi Deniz TAŞKIN

Tez Danışmanı ……

Trakya Üniversitesi Fen Bilimleri Enstitüsü onayı

Prof. Dr. Hüseyin Rıza Ferhat KARABULUT

Fen Bilimleri Enstitüsü Müdürü

T.Ü.FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ DOKTORA PROGRAMI
DOĞRULUK BEYANI

Trakya Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında, tüm verilerin bilimsel ve akademik kurallar çerçevesinde elde edildiğini, kullanılan verilerde tahrifat yapılmadığını, tezin akademik ve etik kurallara uygun olarak yazıldığını, kullanılan tüm literatür bilgilerinin bilimsel normlara uygun bir şekilde kaynak gösterilerek ilgili tezde yer aldığını ve bu tezin tamamı ya da herhangi bir bölümünün daha önceden Trakya Üniversitesi ya da farklı bir üniversitede tez çalışması olarak sunulmadığını beyan ederim.

10 / 02 / 2022
Kenan BAYSAL

Doktora Tezi
Derin Öğrenme Temelli Yeni Nesil Güvenlik Duvarının Tasarlanması
T.Ü. Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

İnternet kullanımının yaygınlaşması ile insan hayatında kalıcı değişimler meydana gelmiştir. Bu değişimler hem iyi hem kötü anlamda değerlendirilebilir. Web üzerindeki sınırsız bilgi akışında faydalı içeriklerin yanı sıra, zararlı içeriklere de isteyerek veya istemeyerek maruz kalınabilmektedir. Özellikle çocuk ve gençlerin üzerinde zararlı içeriklerin, telafisi zor hasarlara yol açtığı bilinmektedir. İçerik yönünden zararsız kabul edilebilecek web sayfalarının içeriğinde bile, reklam veya kişisel gönderi olarak zararlı içeriğe rastlanabilmektedir. Bu ödev, oyun ve sosyal medya sayfalarında sıkça görülebilen bir durumdur. Bu tür web sayfalarını tümden engellemek ise farklı sorunlara yol açabilmektedir. Ayrıca geleneksel yöntemler ile web sayfalarının içeriğinin düzenli olarak takip edilmesi ve değerlendirilmesi mümkün değildir.

Çalışmanın amacı, web sitelerinin bütünüyle engellenmesinden ziyade, gelen trafik üzerindeki içeriğin tespit edilip, içeriğin zararlı veya zararsız olarak sınıflandırılarak, sadece o içerik üzerinde engelleme yapılmasını sağlayacak bir yeni nesil güvenlik duvarı filtreleme algoritması geliştirmektir. Web sayfalarında görsellerin yoğunluğu ve etkisi daha fazla olduğundan, çalışma kapsamında değerlendirilen içerik sadece görseller olarak ele alınmıştır. Görsel içeriğinin yorumlanmasında, günümüzde en güçlü yöntem olan derin öğrenme yöntemi kullanılmıştır.

Zararlı içerik olarak değerlendirilen görseller sadece alkollü içecekler ile sınırlandırılmıştır. Lmdb veri tabanı ile toplamda 4,6 milyon görselden oluşan bir veri seti ile CAFFE ortamında GoogLeNet ağı kullanılarak eğitim gerçekleştirilmiştir. Eniyileyici yöntemi olarak Adam algoritması kullanılmıştır. Eğitim sonucunda %97,6469 doğruluk

değerine ulaşılmıştır. Karışıklık matrisi hesaplamalarında F1 skoru %87,75526188 olarak hesaplanmıştır.

Güvenlik duvarı mitmproxy vekil sunucusu ile uygulanmış ve web sayfalarında görsellerin başarıyla filtrelenmesi sağlanmıştır.

Yıl : 2022

Sayfa Sayısı : 146

Anahtar Kelimeler : derin öğrenme, güvenlik duvarı, evrişimli sinir ağları

Doctoral Thesis
Designing a Deep Learning Based Next Generation Firewall
Trakya University Institute of Natural and Applied Sciences
Computer Engineering Department

ABSTRACT

Permanent changes have occurred in human life with the widespread use of the Internet. These changes may both be considered as good and bad. Along with useful contents within this unlimited flow of information through the web, people may also be willingly or unwillingly exposed to harmful contents. It is well known that these harmful contents cause irrecoverable damages particularly on the young and the children. Harmful contents can even be seen in websites, which may content-wise be considered harmless, in the form of advertisements or personal posts. This can be commonly seen in homework, gaming, or social media pages. Completely blocking these websites may cause other problems. In addition, it is impossible to regularly track and review the contents of the websites through traditional methods.

The purpose of this study is to develop a next-generation firewall filtering algorithm, which would detect the contents from the inbound traffic, and categorize them as harmless or harmful, and perform a content-wise blocking process, rather than completely blocking the websites. Since the visual images on the websites are more intense and have higher impact, the contents reviewed within the scope of this study were only discussed as visual images. Today's most powerful method, deep-learning method, was used in interpretation of the contents of the visual images.

The visual images, which were considered as harmful, were limited to alcoholic beverages. The training was performed by using GoogLeNet network within Caffe environment with a data set containing 4.6 million images with Lmdb database. Adam algorithm was used as optimization method. At the end of this training, 97.6469% of

accuracy value was reached. F1 score was calculated as 87.75526188% in confusion matrix calculations.

The firewall was implemented through mitmproxy proxy server, and the visual images on the websites were successfully filtered.

Year : 2022

Number of Pages : 146

Keywords : deep learning, firewall, evrişimli sinir ağları,

TEŞEKKÜR

Doktora eğitimim boyunca bana bilgisi ve tecrübesi ile yol gösteren, değerli danışman hocam Dr. Öğr. Üyesi Deniz TAŞKIN'a teşekkür ederim.

Tez üzerinde değerli tavsiyeleri ve katkılarından ötürü Dr. Öğr. Üyesi Murat Olcay ÖZCAN, Öğr. Gör. Dr. Füsun ŞAHİN HASTÜRK ve Öğr. Gör. Dr. Funda ER ÜLKER hocalarıma teşekkür ederim.

Hayatımın her alanında olduğu gibi tez çalışma süresince de verdikleri destek ve gösterdikleri sabır için sevgili eşim Hülya ve kızım Asya'ya sonsuz teşekkür ederim.

Beni yetiştiren rahmetli annem ve muhterem babama da emeklerinden ötürü teşekkür ederim.

İÇİNDEKİLER

ÖZET....iv
ABSTRACT....vi
TEŞEKKÜR....viii
İÇİNDEKİLER....ix
SİMGELER DİZİNİ....xii
KISALTMALAR DİZİNİ....xiii
ŞEKİLLER DİZİNİ....xv
ÇİZELGELER DİZİNİ....xix
BÖLÜM 1....1
1. GİRİŞ....1
1.1. Çalışmanın Amacı....1
1.2. Çalışmanın Kapsamı....3
1.3. Önceki Çalışmalar....5
BÖLÜM 2....8
2. YAPAY SİNİR AĞLARI....8
2.1. Makine Öğrenmesi....11
2.2. Genel Kavramlar....11
BÖLÜM 3....14
3. DERİN ÖĞRENME....14
3.1. Yapılan Çalışmalar....16
3.2. Derin Sinir Ağları....19
3.3. Geri Besleme....20
BÖLÜM 4....22
4. HİPER-PARAMETRELER....22
4.1. Aktivasyon Fonksiyonu....23
4.2. Gradyan İnişi....27
4.3. Yığın Boyutu – Batch Size....28
4.4. Eniyileyici Algoritmaları....29

4.4.1. Stochastic Gradient Descent (SGD) 30
4.4.2. Momentum 30
4.4.3. Adaptive Gradient – AdaGrad 31
4.4.4. Root Mean Square Propagation - RMSProp 32
4.4.5. AdaDelta 32
4.4.6. Nesterov's Accelerated Gradient – NAG 33
4.4.7. Adaptive Moment Estimation– Adam 34
4.5. Aşırı Öğrenme – Overfitting 35
4.6. Öğrenme Katsayısı – Learning Rate 36
4.7. Tur Sayısı - Epoch 37
4.8. Seyreltme - Dropout 38
BÖLÜM 5 40
5. EVRİŞİMSEL SİNİR AĞLARI 40
5.1. Evrişim İşlemi 41
5.2. Kenar Tespiti 42
5.3. Padding İşlemi 44
5.4. Adım Aralığı 45
5.5. Havuzlama İşlemi 46
5.6. Tensor Yapısı 46
5.7. Kayıp Fonksiyonu 47
5.8. ESA Katman Yapısı 48
5.9. ESA Ağ Modelleri 50
5.9.1. AlexNet 51
5.9.2. VGG16 52
5.9.3. GoogLeNet 53
BÖLÜM 6 57
6. DERİN ÖĞRENME KULLANILARAK GELİŞTİRİLEN YENİ NESİL GÜVENLİK DUVARI 57
6.1. Materyal ve Yöntem 57
6.1.1. Uygulama Zorlukları 58
6.2. Veri Seti 59
6.2.1. Veri Tekilleştirme 62
6.2.2. Veri Çoğaltma 63
6.3. Model Eğitim Süreci 68

6.3.1. Donanım 68
6.3.2. Eğitim Ortamı 69
6.3.3. Hiper-Parametrelerin Belirlenmesi 71
6.3.4. Eğitim 82
6.3.5. Modelin Test Edilmesi 87
6.3.6. Modelin Karışıklık Matrisi Değerlendirmesi 89
6.4. Güvenlik Duvarı Uygulaması 93
BÖLÜM 7 99
7. SONUÇLAR VE TARTIŞMA 99
7.1. Sonuçlar 99
7.2. Tartışma 101
KAYNAKLAR 104
EKLER 117
ÖZGEÇMİŞ 126

SİMGELER DİZİNİ

$\boldsymbol{\beta}$	Bias değeri
$\mathbf{db_t}$	Toplam tekil veri sayısı
$\mathbf{f}$	Aktivasyon fonksiyonu
$\mathbf{i_s}$	Çoğaltma işlem sayısı
$\mathbf{L}$	Katman sayısı
$\mathbf{n}$	Nöron sayısı
$\mathbf{s_v}$	Sentetik veri miktarı
$\mathbf{p}$	Padding işlemi için eklenecek piksel boyutu
$\mathbf{r_n}$	Çoğaltma işlemi için veri başına düşen tekrar sayısı
$\boldsymbol{w}$	Ağırlık parametresi

KISALTMALAR DİZİNİ

Adam	Adaptive Moment
BM	Boltzmann Machine
B-tree	Binary Tree
CAFFE	Convolutional Architecture for Fast Feature Embedding
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
cuDNN	CUDA Deep Neural Network Library
DBN	Deep Learning Networks
DNN	Deep Neural Network
DP	Doğru pozitif
DN	Doğru negatif
DQN	Deep Q Learning Network
ESA	Evrişimsel Sinir Ağları
FF	Feed Forward
GRU	Gated Recurrent Unit
GPU	Graphics Processing Unit
HN	Hopfield Network
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
LMDB	Lightning Memory-Mapped Database
LSTM	Long short-term memory
Malloc	Memory Allocation

memcpy	Memory Copy
MC	Markov Chain
MLP	Multi Layer Perception
MSE	Mean Squared Error
NAG	Nesterov’s Accelerated gradient
SGD	Stochastic Gradient Descent
SWOT	Strengths, Weaknesses, Opportunities, Threats
SWRNN	Spiking Wavelet Radial Basis Neural Network
RBF	Radial Basis Function
ReLu	Rectified Linear Units
RNN	Recurrent Neural Network
URL	Uniform Resource Locator
Val	Validation
Vd	Ve diğerleri
VGG	Visual Geometry Group
YN	Yanlış negatif
YP	Yanlış pozitif
YSA	Yapay sinir ağları

ŞEKİLLER DİZİNİ

Şekil 2.1. Sinir hücresi yapısı (Openstax, 2021) 8

Şekil 2.2. Perseptron – algılayıcı 9

Şekil 2.3. Çok katmanlı bir yapay sinir ağı örneği 10

Şekil 2.4. Sınıflandırma (a) ve Regresyon (b) 12

Şekil 2.5. Kümeleme (a) ve Bağlantı (b) 13

Şekil 3.1. Makine Öğrenmesi (üstte) ve Derin Öğrenme (altta) 15

Şekil 3.2. Derin sinir ağlarının veri boyutu ile performans ilişkisi (Dong vd, 2021) 15

Şekil 3.3. Derin sinir ağı örneği 19

Şekil 3.4. İleri ve Geri besleme 21

Şekil 4.1. Hiper-parametrelerde iterasyon - maliyet ilişkisi 23

Şekil 4.2. Ham veri (a), Lineer aktivasyon fonksiyonu ile sınıflandırma (b), lineer olmayan aktivasyon fonksiyonu ile sınıflandırma (c) 24

Şekil 4.3. Aktivasyon fonksiyonları 26

Şekil 4.4. Gradyan inişi 28

Şekil 4.5. Eniyileyici algoritmaları arasındaki ilişki haritası (Seyyarer vd, 2020) 29

Şekil 4.6. Eniyileyici türleri kedi-köpek veri seti üzerindeki eğitiminin karşılaştırması 30

Şekil 4.7. Klasik momentum (üstte), Nesterov Hızlandırılmış Gradyan (altta).....33

Şekil 4.8. Yetersiz Öğrenme (a), İyi Öğrenme (b), Aşırı Öğrenme (c).....36

Şekil 4.9. (a) Yüksek öğrenme oranı (b) Düşük öğrenme oranı (c) Optimum öğrenme oranı (Akca, 2020) 37

Şekil 4.10. Tur sayısı ve öğrenme başarısı arasındaki ilişki 38

Şekil 4.11. Standart bir sinir ağı (a) Seyreltme işleminden sonra sinir ağı (b) 39

Şekil 5.1. Evrişim işlemi ile özellik hiyerarşisi
(Lee, Grosse, Ranganath & Ng, 2011).....41

Şekil 5.2 İki boyutlu matris üzerinde evrişim işlem örneği 42

Şekil 5.3. Kenar bulma işlemi.....43

Şekil 5.4. Yatay, dikey ve çapraz kenar bulma örnekleri 44

Şekil 5.5. 8x8 boyutundaki bir matrise 3x3 boyutunda bir filtre uygulanması durumunda uygulanacak padding işlemi.....45

Şekil 5.6. Ortalama havuzlama ve maksimum havuzlama 46

Şekil 5.7. Çok kanallı evrişim ve tensor yapısı.....47

Şekil 5.8. ESA Katman yapısı.....49

Şekil 5.9. AlexNet Mimarisi 52

Şekil 5.10. VGG16 Mimarisi 53

Şekil 5.11. (a) Doğrusal evrişim katmanı (b) MLP katmanı (c) Ağ içinde ağ yapısı 54

Şekil 5.12. (a) Görselin çoğunu kaplayan bir kedi (b) görselin bir kısmını kaplayan bir kedi (c) görselin küçük bir kısmını kaplayan kedi 55

Şekil 5.13. Naif başlangıç modülü (Szegedy vd., 2014).....55

Şekil 5.14. Boyut küçültme ile başlangıç modülü.....56

Şekil 6.1. İşlem basamakları 58

Şekil 6.2. Web crawler blok şeması.....60

Şekil 6.3. alkol_var etiketli veri seti örnekleri 61

Şekil 6.4. alkol_yok etiketli veri seti örnekleri 61

Şekil 6.5. MSE ile yapılan karşılaştırma.....63

Şekil 6.6. Veri çoğaltma örnekleri 64

Şekil 6.7. Rastgele kesme ve rastgele silme işlemlerinde yaşanabilecek veri kayıpları.....66

Şekil 6.8. Veri seti training, test ve validation bölüm oranları 67

Şekil 6.9. db_00 veri seti kullanılarak GoogLeNet (a) ve AlexNet (b) karşılaştırması 72

Şekil 6.10. 2 (a), 16 (b) ve 32 (c) yığın boyutlarının db_00 veri seti ve GoogLeNet kullanılarak kıyaslanması.....74

Şekil 6.11. db_00 veri seti ile AdaDelta, AdaGrad, Adam, NAG, RmsProp ve SGD Eniyileyici türlerinin eğitim karşılaştırmaları 75

Şekil 6.12. Test edilen öğrenme oranı sönümleme ilkeleri.....77

Şekil 6.13. Azalan Sönümleme (a) Sigmoid Sönümleme (b) Polinom Sönümleme (c) Devrik Sönümleme (d) Sabit (e) Üstel Sönümleme (f) temel öğrenme oranı ilkelerinin db_00 ile eğitim karşılaştırmaları 78

Şekil 6.14. Öğrenme katsayılarının karşılaştırılması 80

Şekil 6.15. GoogLeNet Inception Ağ Mimarisi.....83

Şekil 6.16. GoogLeNet Inception modülü 84

Şekil 6.17. Model_1 eğitim grafiği 86

Şekil 6.18. Model_2 eğitim grafiği 86

Şekil 6.19. Model_3 eğitim grafiği 87

Şekil 6.20. Digits arayüzü ile alkollü içecek tespit örnekleri 88

Şekil 6.21. Model test uygulama blok şeması 89

Şekil 6.22. Karışıklık matrisi ikili sınıflandırma tablosu 90

Şekil 6.23. Karışıklık matrisi hesaplaması için gerçekleştirilen yedi ayrı testin DP, YP, DN ve YN veri sayıları 91

Şekil 6.24. Yakın benzerlik ve uzak benzerlik DP, DN, YP ve YN değerleri 92

Şekil 6.25. Uygulama blok şeması 93

Şekil 6.26. Mitmproxy https bağlantı (Mitmproxy, 2021) 95

Şekil 6.27. Aktif güvenlik duvarı ile web içerik ekran görüntüleri ve terminal çıktısı ..96

Şekil 6.28. Güvenlik duvarı aktifken ve pasifken web tarayıcıda görüntülenen sayfaların ekran görüntüsü 97

ÇİZELGELER DİZİNİ

Çizelge 3.1. Derin öğrenme ile yapılan çalışmalar 17

Çizelge 4.1. Aktivasyon fonksiyonları 25

Çizelge 5.1. Kayıp Fonksiyonları 48

Çizelge 5.2. ILSVRC kazananları ve hata oranları (Image-net, 2021) 51

Çizelge 6.1. Augmentor kütüphanesinden uygulanan işlemler ve parametreler 65

Çizelge 6.2. db_01, db_02 ve db_03 için veri setleri için veri çoğaltma işlem sayısı ve toplam veri miktarı 66

Çizelge 6.3. Veri setleri ve test, train, validation veri miktarları 68

Çizelge 6.4. Eğitim ve Uygulama için kullanılan donanım 69

Çizelge 6.5. GoogLeNet ve AlexNet mimarilerinin 20-50 arası döngülerde accuracy (val) loss (val) ve loss (train) değerlerinin karşılaştırmaları 72

Çizelge 6.6. 2, 16 ve 32 yığın boyutlarının 20 – 50 arası döngülerde accuracy (val) loss (val) ve loss (train) değerlerinin karşılaştırmaları 74

Çizelge 6.7. Eniyileyici türleri 20 – 50 arası döngülerde accuracy (val) loss (val) ve loss (train) değerlerinin karşılaştırmaları 76

Çizelge 6.8. Öğrenme oranı sönümleme ilkelerinin 20 – 50 arası döngülerde accuracy (val), loss (val) ve loss (train) değerlerinin karşılaştırmaları 79

Çizelge 6.9. 0,1 – 0,01 – 0,001 – 0,0001 – 0,00001 öğrenme katsayı değerlerinin 20 – 50 arası döngülerde accuracy (val), loss (val) ve loss (train) değerlerinin karşılaştırmaları 81

Çizelge 6.10. Eğitilen modellerin parametre bilgileri 85

Çizelge 6.11. Karışıklık matrisi değerlendirme metrikleri 91

Çizelge 6.12. Yapılan 7 ayrı testin kesinlik, geri çağırma, doğruluk, özgüllük ve f1 skor değerleri 92

Çizelge 6.13. Yakın ve Uzak benzerlik veri setleri ile yapılan sınıflandırmada elde edilen kesinlik, geri çağırma, doğruluk, özgüllük ve f1 skoru değerleri 93

BÖLÜM 1

1. GİRİŞ

1.1. Çalışmanın Amacı

1900'lü yılların ikinci yarısından itibaren elektronik ve bilgisayar alanında gerçekleşen yüksek ivmeli gelişim sayesinde, 2000'li yıllar "Bilişim Çağı" veya "İnternet Çağı" olarak değerlendirilmektedir. Son 10 yıl içerisinde, sosyal medya kavramının hızlı gelişimi ve evrimi ile birlikte internet kullanımının insan ilişkileri üzerinde gözlemlenebilen kalıcı etkileri olmuştur. İnternetin doğası gereği var olan global yapısı nedeniyle, tüketim, iletişim, etkileşim gibi eylemlerin zamandan ve mekândan bağımsız olarak geniş kitleleri farklı biçimlerde etkilediği görülmektedir. Bu etkileşimin büyük çoğunluğunu görseller oluşturmaktadır. Wearesocial.com web sitesinin 2021 yılı verilerine göre, dünya üzerinde 4,66 milyar aktif internet kullanıcısı içerisinde 4,20 milyar aktif sosyal medya kullanıcısı mevcuttur. Yine aynı verilere göre internete bağlı cihazlar üzerinde günlük ortalama geçirilen zaman 6 saat 54 dakikadır. Türkiye için sadece sosyal medya üzerinde günlük geçirilen ortalama zaman 2 saat 57 dakika olarak belirtilmiştir (Wearesocial, 2021).

Çubukçu & Bayzan (2013), SWOT (Strengths, Weaknesses, Opportunities, Threats) analizi kullanarak yapmış oldukları "Türkiye'de Dijital Vatandaşlık Algısı ve Bu Algıyı İnternetin Bilinçli, Güvenli ve Etkin Kullanımı ile Artırma Yöntemleri" başlıklı çalışmalarında zararlı içerik üretim ve paylaşımına dikkat çekmişlerdir.

Lehu & Bressoud'tan aktaran Yaraş vd. (2018), geleneksel medyanın eskiye göre etkisinin azalmasıyla reklamcıların, ürün yerleştirme tekniğini bir pazarlama yöntemi olarak kullanmaya başladıklarını belirtmişlerdir. Yaraş vd. (2018), silah, alkol ve tütün

ürünleri için getirilen reklam kısıtlamalarını aşmak ve özellikle gençleri etkileyebilmek için firmaların ürün yerleştirme tekniklerini tercih ettiğini belirtmektedir.

Sosyal medya üzerinde yayınlanan içeriklerin önemli bir bölümü, ürün tanıtımı veya ürün yerleştirme içermektedir. Şiddet ve cinsellik gibi bazı içerikler sosyal medya platformları tarafından engellense de alkollü içecek, tütün kullanımı veya ateşli silahlar bu filtrelemeye tabi olmamaktadır. Sosyal medya haricindeki web sitelerinin içeriklerinde ise şiddet ve cinsellik gibi, gençlerin zihinsel aktivitelerini etkileyebilecek çok daha fazla görsel bulunabilmektedir.

Özellikle küçük yaştaki çocuklar, web sayfalarının uygunsuz içeriklerine karşı yetişkinlerden daha savunmasız durumdadırlar. İplikçi & Batu (2018), Türkiye'de çocuk web siteleri üzerinde yer alan reklamlara yönelik bir içerik çalışması yapmışlardır. Çalışma sonucunda reklam içeriklerinde +18 yetişkin içeriklerin yer aldığını, bununla birlikte çocuk ruh sağlığını olumsuz yönde etkileyebilecek içeriklere rastlandığını belirtmişlerdir. Kapferer'dan aktaran İplikçi vd. (2018), 5 yaşından küçük çocukların reklam görsellerinin işlevi hakkında bilgi sahibi olmadıklarını, %26'sının ise reklamları güvenilir bilgi kaynağı olarak gördüklerini ifade etmişlerdir.

Görsel medya içeriklerine yönelik belirlenmiş birçok standart ve düzenleme bulunmaktadır. Uzun (2014), çocuklara yönelik medya etik ve davranış kuralları ile ilgili yaptığı çalışmasında, BBC etik kurallarını ele almış, ancak web üzerinden yayın yapan kuruluşların bu etik kurallarını düzenlemesi ve geliştirmesi konusunda kuruluşların ticari yapıları göz önünde bulundurulduğunda, düzenleme yapılmasının beklenemeyeceğini belirtmiştir. Bu durumda tümden engellemeler ve yasaklamalar ile durumun düzeltileceği düşünülebilir.

Kanbur (2020), görsel medyanın çocuk sağlığı üzerindeki etkileri hakkında yaptığı çalışmada, günümüz dünyasında çocukların ve gençlerin görsel medya araçlarını kullanmasının tamamen yasaklanmasının çözüm olmadığını belirtmektedir. Bu tür bir kısıtlama çocukların arkadaş çevresi tarafından dışlanmasına sebep olabilmektedir. Bu durum ise çocuklar üzerinde baskı oluşturabilmektedir.

İçerikler konusunda sosyal medya kuruluşları, içerik üreticileri ve diğer dijital medya kurumlarından mevcut koşullar üzerinde bir iyileştirme ve düzenleme beklemek gerçekçi bir yaklaşım olmayacaktır. Dolayısıyla kullanıcıların bu tür zararlı içeriklerden

kaçınabilmeleri için filtreleme yazılımlarına ihtiyaçları olmaktadır. Var olan güvenlik duvarı yapısı ile filtreleme işlemleri, belirli bir kural oluşturma ve kelime bazlı veya adres engelleme biçiminde gerçekleştirilmektedir. Bu yapı, bir görselin içerisinde uygunsuz içeriğin var olup olmadığını tespit edebilecek bir çalışma biçimine sahip değildir. Dolayısıyla geleneksel yöntemler ile içerik bazlı filtreleme mümkün değildir. Önceden belirlenmiş domain listesi veya belirli bir URL (Uniform Resource Locator) ile filtreleme gerçekleştirilmektedir. Bu da bir web sitesinin tamamen engellenmesi anlamına gelmektedir. Ancak sosyal medya gibi çeşitli içerik ve ortamların bulunduğu, reklam geliri açısından ticari bir değeri olan bu mecraların bütünüyle engellenmesi, farklı sorunları ortaya çıkarmaktadır. Bu sebeple içerik bazlı filtreleme yapmaya ihtiyaç duyulmaktadır.

Bu doğrultuda çalışmanın amacı, içeriklerin akıllı bir algoritma ile tespit edilebilmesi ve internet kullanımının tamamen kısıtlanmasına gerek kalmadan güvenli bir internet kullanımının mümkün kılınabilmesidir.

1.2. Çalışmanın Kapsamı

Makine öğrenmesi ve derin öğrenme kavramları "Yapay Zekâ" kavramının bir alt çalışma sahası olarak karşımıza çıkmaktadır. Yapay Zekâ, insan davranışlarının bilgisayar gücü ile benzetiminin sağlanması, öğrenme, karar verme ve değerlendirme süreçlerinin insan benzeri aşamalarla gerçekleşmesi temelini esas almaktadır (Zhang & Lu, 2021). Bilgisayar kabiliyeti ile insan zekasına benzer bir yapı geliştirme fikri yeni bir fikir değildir. Bu fikrin geçmişi esasen 1950'li yıllara kadar uzanmaktadır. Bilim kurgu dünyasında ise 1920'li yıllara kadar dayanmaktadır. Sinema sektöründe ilk yapay zekâ fikri, 1927 yılında çevrilen "Metropolis" filminde işlenmiştir. Yapay zekanın bir alt kolu olan makine öğrenmesi kavramı ise, bu alanda öncü kabul edilen, Arthur Samuel tarafından 1959 yılında yayımlanan makalede "makinelerin özel olarak programlandıkları sonuçları öğrenebilme yeteneği" olarak tanımlanmıştır (Samuel, 1959). Makine öğrenmesi, mevcut problemin, o problemin çözümünde kullanılabilecek nitelikteki veri ile modellemesini gerçekleştiren bilgisayar algoritmalarıdır (Atalay & Çelik, 2017).

1990'lı yıllarda donanımsal gelişmelerin sağladığı imkanlar ile veri madenciliği kavramı bir araya geldiğinde, makine öğrenmesi üzerinde yapılan çalışmalar artmıştır (Bose & Mahapatra, 2000). Finans, sağlık, pazarlama ve eğitim gibi birçok alanda

uygulamalar gerçekleştirilmiştir. Mevcut veri eşliğinde eğitilen algoritmalar, karar, analiz veya tespit aşamasında kullanılmıştır.

Derin öğrenme kavramı, makine öğrenmesinin bir alt alanı olarak, arka planda birçok disiplinin katkılarıyla oluşum göstermiştir. Derin öğrenme için ilk gerçek çalışan algoritma Ivakhnenko & Lapa'nın (1967) çalışması kabul edilmektedir. Hinton vd. tarafından 2006 yılından itibaren yayınlanan çalışmalar ve son yıllarda artan GPU (Graphics Processing Unit) gücü sayesinde ise yaygın kullanım alanı bulmuştur. Bu çalışmalarda çok katmanlı ileri beslemeli bir sinir ağının her seferinde bir katmanının etkili bir şekilde önceden eğitilebileceği gösterilmiştir (Hinton, 2007), (Hinton, Osindero & Teh, 2006). Geleneksel yöntemlere göre derin öğrenme algoritmasının eğitilmesi için çok büyük boyutlarda veri setleri gerekmektedir. Veri seti büyüklüğü arttıkça, eğitilen algoritmanın çalışma doğruluğu ve verimi artmaktadır.

Derin öğrenme, önümüzdeki yılları etkileyecek en yüksek potansiyele sahip araştırma alanı olarak gösterilmektedir (Du, Cai, Wang & Zhang, 2016). CAFFE, Tensorflow, Torch, Keras yaygın olarak kullanılan derin öğrenme kütüphane yığınları arasındadır. Açık kaynaklı ve kolay ulaşılabilir bu kütüphane yığınları sayesinde görsel tanıma ve sınıflandırma konusunda başarılı sonuçlar elde edilmiştir (Pathak, Pandey & Rautaray, 2018). Bu alandaki başarısını internet trafiği üzerindeki görsel paketlerinin incelenmesi ve sınıflandırılması için de kullanmak mümkündür.

Bu çalışmada ile web sitesini engellemeden, sadece zararlı kabul edilen görselleri engelleyebilecek bir yapı geliştirilmiştir. Zararlı içerik olarak alkollü içecekler kategorisi seçilmiştir. Tasarlanan web crawler sayesinde veri seti oluşturmak için çeşitli web sayfalarından görseller yedi ay gibi bir sürede temin edilmiştir. Temin edilen görseller etiketlenerek algoritma derin öğrenme yöntemi ile eğitilmiş ve alkollü içecek içeren görsellerin tanınması konusunda başarı sağlanmıştır. Web görsel filtreleme olarak uygulaması gerçekleştirilmiştir. Bu şekilde özellikle sosyal medya gibi web sitesinin tamamını engellemenin zor olduğu durumlarda, işlevsel olarak sadece zararlı kabul edilen görseller filtrelenmiştir.

1.3. Önceki Çalışmalar

Güvenlik duvarı, kişisel ağ ile açık internet erişimi arasında, belirlenen kurallar dahilinde, giren çıkan bütün paketlerin üzerinden geçtiği bir süzgeç görevi görmektedir. Aşağıda konuya ilişkin yapılan bazı çalışmalara yer verilmiştir.

Gouda & Liu (2006), yapmış olduğu "Yapılandırılmış güvenlik duvarı tasarımı" adlı çalışmalarında, bir güvenlik duvarının tasarımında üç temel sorundan bahsetmişlerdir. Bunlar; tutarlılık sorunu olarak kuralların doğru bir şekilde düzenlenmesi, bütünlük sorunu olarak her trafiğin tamamının gözden kaçırılmadan süzgeçten geçirilmesi ve kuralların sayısının düşük tutulması zorluğudur.

Dawoud, Shahristani & Raun (2018), nesnelerin interneti için derin öğrenme temelli, algılayıcı modülü olarak kısıtlı Boltzmann makinası kullanan bir saldırı tespit mekanizması tasarlamışlardır.

Le, Boydel, Namee & Scanlon (2018), derin öğrenme tabanlı, uzmanlık bilgisi gerektirmeyen karmaşık model ve özellik tanımlaması için tamamen veriye dayalı bir yaklaşıma sahip kötücül yazılım sınıflandırma uygulama çalışması yapmışlardır.

Zuo, Wu, Min & Cui (2018), derin öğrenme algoritmaları kullanarak kısıtlı koşullar altında ağ planlaması için yazılım tanımlı ağ (SDN - Software Defined Networking) kontrolöründe kullanılmak üzere veri odaklı bir model önermişlerdir.

Özcan (2014), çok düşük çözünürlükteki görsellerden kimlik tespiti yapılabilmesi için, derin öğrenme algoritmaları kullanarak bu görselleri yüksek çözünürlüğe dönüştüren bir çalışma yapmıştır.

Özhan (2013) ağ üzerinde güvenliği arttırma amacını güttüğü çalışmasında, veri madenciliği ve makine öğrenmesi algoritmalarını kullanarak, güvenlik duvarı günlüklerini analiz etmiş ve modellemiştir.

Makine öğrenmesi ve derin öğrenme kullanılarak, web crawler araçları ile web siteleri üzerinde çeşitli sınıflandırma ve güvenlik çalışmaları da yapılmıştır.

Mulahuwaish, Gynorick, Ghafoor, Maghdid & Rawat (2020), haber sitelerinin içeriklerini tarayarak, ekonomi, teknoloji, bilim, sağlık ve eğlence başlıkları altında makine öğrenmesi yoluyla sınıflandırma çalışması yapmışlardır.

Hannousse & Yahiouche (2021), web sayfalarından phishing tespiti için makine öğrenmesi yöntemlerini kullanarak, deneysel bir model için tarama ve sınıflandırma çalışması yapmışlardır.

Lopez - Sanchez, Arrieta & Corchado (2019) web sayfalarını, içerdikleri görsellere göre derin öğrenme ve metrik öğrenme tekniklerini kullanarak sınıflandırma çalışması yapmışlardır. Böylelikle web sitesinin dilinden bağımsız olarak web sitelerini sınıflandırmayı başarmışlardır. Görsel içerik tabanlı web sayfası sınıflandırması sorununa metrik öğrenme ile birlikte transfer tekniklerini de uygulayarak farklı açıdan yaklaşmışlardır.

Uzair, Beg, Mujtaba & Majeed (2019), web sayfalarının aygıt üzerindeki enerji tüketimini arttıran görselleri makine öğrenmesi ile tespit ederek, web sayfalarının aygıt üzerinde daha düşük güç tüketmesini sağlamışlardır.

Kumar, Solanki & Singh (2020), kullanıcının web geçmişi günlüklerini makine öğrenmesi ile işleyerek kullanıcının hareketlerini tahmin eden bir model çalışması yapmışlardır. Çalışmanın amacı, kullanıcının hareketlerinden yola çıkarak, web sitesinin dinamik yapısının verimliliği artıracak şekilde tekrar biçimlenmesini sağlamaktır.

Nengroo & Kuppusamy (2018), makine öğrenmesi tekniklerini kullanarak web sayfalarında reklamları filtreleme ile ilgili bir çalışma yapmışlardır. Tasarlamış oldukları bu filtreleme yapısının %98,6 doğruluk oranına sahip olduğunu belirtmişlerdir.

Ertam (2019) yaptığı çalışmada, güvenlik duvarı aracılığı ile elde edilen verileri derin öğrenme yöntemi kullanarak bir sınıflandırma yapısı önermektedir. Çalışmasında uzun kısa süreli bellek ve çift yönlü uzun kısa süreli bellek sinir ağlarını, tek tek ve birbirine bağlanarak kullanmış, performans testlerinde %97 doğruluk elde etmiştir.

Hu vd. (2019), yapmış oldukları çalışmada, sosyal medya paylaşımlarında mahremiyet içerebilecek fotoğrafları tespit eden ve filtreleyebilen bir güvenlik duvarı önermişlerdir. Veri setinde “öpme”, “sarılma” ve “el ele tutuşma” temalı görselleri etiketleyerek eğitim gerçekleştirmişlerdir. Derin öğrenme ile gerçekleştirdikleri bu görsel sınıflandırma çalışmasında VGG-16 ağını kullanmışlardır. Oluşturdukları güvenlik duvarı yapısı, kullanıcı için 1 ile 7 arası hassasiyet seviyesi ayarlamasına izin vermektedir. Eğer iki kişinin dahil olduğu bir görselde hassasiyet seviyesi yüksek olarak tespit edilirse görselin paylaşılması güvenlik duvarı tarafından engellenmektedir.

Nasir, Khan ve Varlamis (2021), özellikle sosyal medya üzerinden yayılan sahte haberlerin tespiti için hibrit bir CNN-RNN modeli kullanarak metin sınıflandırma çalışması yapmışlardır. ISO ve FA-KES sahte haber veri setleri kullanmışlardır. Hibrit olmayan diğer çalışmalara göre önemli ölçüde daha iyi sonuçlar elde ettiklerini bildirmişlerdir.

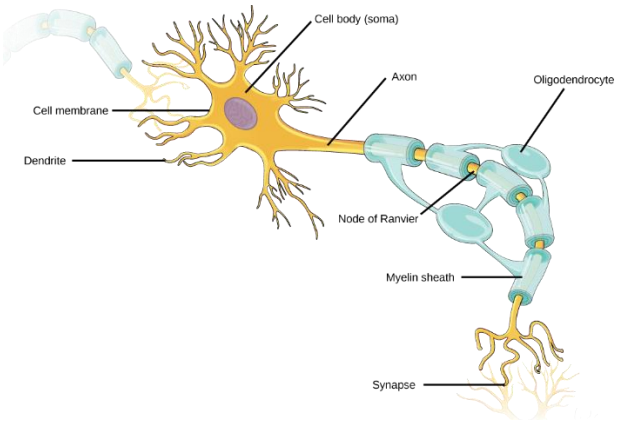
Hafiz, Haque, Rakshit ve Uddin (2020), yayımladıkları çalışmada derin evrişimsel sinir ağları kullanarak, alkolsüz içecekleri kalorisi, enerjisi, içerdiği şeker miktarı gibi kategorilere göre görsel üzerinden sınıflandırma yapabilen bir algoritma geliştirmişlerdir. Araştırıcılar, InceptionV3, VGG19, MobileNet, ve ResNet50 öğrenme metotlarını kullanmışlar ve en yüksek doğruluk oranını %99.88 ile Resnet50 ile elde etmişlerdir.

Niu ve arkadaşları (2020), yapmış oldukları çalışmada, büyük veri setleri için derin öğrenme tabanlı bir model kullanarak geleneksel olmayan görüntülerin sınıflandırılması ve akıllı filtrelenmesi için bir yaklaşım önermişlerdir. Yapmış oldukları bu çalışmanın temelinde, web üzerindeki geleneksel olmayan ve pornografi gibi zararlı içeriğe sahip olabilecek görsellerin doğru bir şekilde tespit edilmesi ve filtrelenmesini sağlayabilmeyi amaçlamışlardır. Bu amaçla, eğitilecek modelin başarım yüzdesini arttırabilmek için farklı bir yaklaşım denemişlerdir. Bu yöntemin, tıbbi görüntü analizi gibi farklı alanlarda da uygulanabilir olduğunu çalışmalarının sonuç kısmında ifade etmişlerdir.

BÖLÜM 2

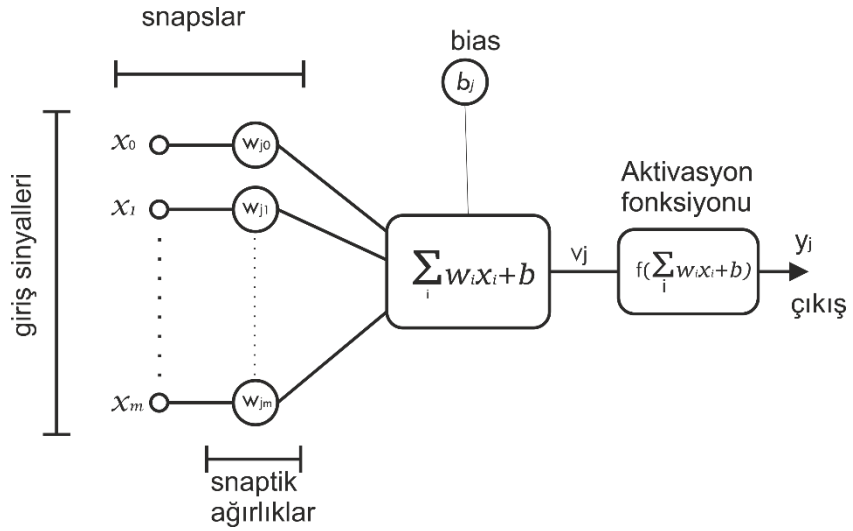
2. YAPAY SİNİR AĞLARI

İnsan beyninin öğrenme süreci beyinde bulunan sinir hücrelerinin arasında kurulan elektriksel bağlar ile açıklanmaktadır. Yapılan çalışmalarda beynin çalışma yapısı henüz kesin olarak açıklanamasa da öğrenme süreci ve bilginin yayılımının sinir hücrelerinde bulunan akson, dendrit ve sinapslar aracılığı ile gerçekleştiği sonucuna varılmıştır. Sinir hücresinde, aksonlar çıkış darbesi üreten tek yönlü yapıdadırlar. Dendrit ise diğer hücrelerden gelen iletim darbelerini toplayan kısımlardır. Dendrit sadece gelen elektrik darbelerini algılar, pasif bir yapıya sahiptir ve sinir hücresinin giriş kısmı olarak kabul edilebilir. Sinapslar diğer hücrelerin dendritleri ile aksonlarını birbirine bağlayan bağlantı kısımlarıdır.



Şekil 2.1. Sinir hücresi yapısı (Openstax, 2021)

Yapay sinir ağları, biyolojik sinir dokusu örneklerinden esinlenilerek oluşturulan matematik modelleri ile kendi kendine karar verebilen makineler oluşturmayı hedeflemektedir. Beynin biyolojik yapısına, bilgisayar ortamında matematiksel olarak benzer bir model oluşturma fikri, Warren vd. (1943) tarafından ortaya atılmıştır. Mantıksal elektrik devreleri kullanarak oluşturdukları bu ilk yapay sinir hücresi ile mantıksal ifadelerin oluşturulabileceğini göstermişlerdir. Bu çalışma ile ilk yapay sinir hücresinin oluşturulduğu kabul edilmektedir. Kullanışlı bir yapı oluşturmak için birden fazla yapay sinir hücresinin kullanılması gerektiği sonucuna varılmış ve yapay sinir ağlarının temeli atılmıştır.



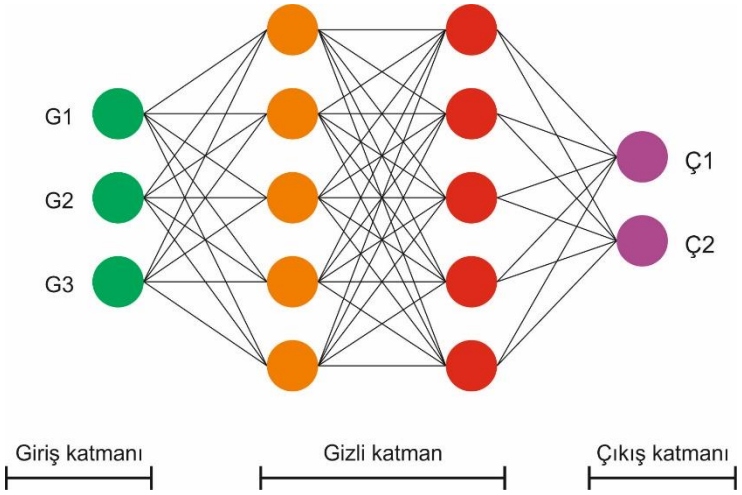
Şekil 2.2. Perseptron – algılayıcı

Şekil 2.2'de bir yapay sinir hücresi örnek modeli verilmiştir. Tek katmanlı bu yapı yapay sinir ağının en temel birimidir ve perseptron yani algılayıcı olarak isimlendirilir. Burada *x* girdi değeri olarak bağımsız değişkendir. *W* değeri ağırlık parametresidir. Ağırlık başlangıç değeri olarak rastgele bir değer olabilir veya ayarlanabilmektedir. Rastgele bir değer olarak atanırken sıfırdan farklı bir değer olarak artı (+) veya eksi (–) bir değer belirlenebilir veya önceki eğitim verilerinin değerleri kullanılabilir. Ağırlık değeri her girdinin önem durumunu belirleyen bir değerdir. Bu değer ne kadar fazla olursa girdinin toplam fonksiyonu üzerindeki etkisi de ona göre fazla olacaktır. Ayrıca uzun

süreli kalıcı hafıza oluşmasını sağlayan kısım, ağırlık değerlerinin öğrenim durumunda yeniden değerlenmesi ile gerçekleşmektedir. Başka bir ifadeyle, ağ üzerindeki bilgi, ağırlıklar ile temsil edilir. Bias değeri ise aktivasyon fonksiyonun ötelenmesini sağlayarak giriş değerlerinin sürekli 0 olması durumunu engeller. Giriş değerleri sürekli 0 olursa öğrenme gerçekleşmez. Çıkış durumu denklem 2.1'de verilmiştir.

$$y = f(b + \sum_{i=1}^{m} x_i \, w_i) \tag{2.1}$$

Şekil 2.3'te giriş, çıkış ve iki gizli katmanı olan çok katmanlı bir yapay sinir ağı modeli verilmiştir. İleri beslemeli bu modelde düğüm yapıları arasındaki bağlantılar çizgiler ile ifade edilmektedir. Şekilde giriş katmanları hariç 5+5+2 olmak üzere toplamda 12 nöron bulunmaktadır. [3 x 5] + [5 x 5] + [5 x 2] olmak üzere 50 ağırlık ve 5+5+2 =12 bias değeri olarak 62 adet öğrenilmesi gereken parametre vardır.



Şekil 2.3. Çok katmanlı bir yapay sinir ağı örneği

Aynı katmanda bulunan nöronların birbiri ile bağlantısı bulunmamaktadır. Her bir nöron kendisine gelen girdinin ağırlık parametresiyle birlikte bias değeri aracılığıyla çıkış değerinin ötelenmesi sayesinde bir sonraki katmanda bulunan nöronlara bilgiyi aktarır. Buradan hareketle modeldeki katmanlarda bulunan nöron sayısı öğrenme durumunu

doğrudan veya dolaylı olarak etkileyebilir sonucuna varılabilir. O halde katman sayısının artması öğrenme performansını arttırabilir gibi görünse de öğrenmenin gerçekleşmesini sağlayan birçok hiper-parametre ve seçilecek aktivasyon fonksiyonunun etkisi büyüktür. Ağ için seçilecek en uygun mimari, optimum seviyede öğrenme yapabilecek kadar büyük, güncelleme yapabilmek içinse optimum seviyede küçük olabilmelidir (Arı & Berberler, 2017). Gereğinden küçük bir ağ, öğrenmeyi istenen seviyede gerçekleştiremez ve gereğinden fazla büyük olarak tasarlanan bir ağ ise öğrenme durumunda aşırıya gideceğinden ezberleme gerçekleşir.

2.1. Makine Öğrenmesi

Makine öğrenmesi, bilgisayar programının kesin yönergeler olmadan eldeki veri setinden çıkarımlar yaparak öğrenebilmesini amaçlamaktadır. Makine öğrenmesi, yapay zekanın bir dalı olarak kabul edilmektedir, ancak istatistik, bilişsel bilim ve bilgi teorisi gibi birçok disiplin ile kesişmektedir (Ławrynowicz & Tresp, 2014).

1950'li yıllardan günümüze gelen süreçte, makine öğrenmesi zaman zaman sekteye uğrasa da bugün birçok alanda karşılaşılan problemlere çözüm üretmek için başvurulan bir modeldir. Hinton'ın 22 Mayıs 2016'da Royal Society'de yaptığı konuşmada, geçmişte derin öğrenme gelişiminin sekteye uğramasının temel sebepleri olarak, eğitimde kullanılacak olan veri setlerinin aşırı küçük olması, bilgisayarların yeteri kadar güçlü olmamaları ve yanlış non-lineer fonksiyonların kullanılmasını göstermiştir. Bugün derin öğrenme, konuşma tanıma, nesne tespiti ve resim sınıflandırma gibi birçok zorlu görevde başarılı performans sergilemektedir. Hatta bazı durumlarda insan performansının üzerine çıkmaktadır (Sun, Zhang, Zhang & Hu, 2017)

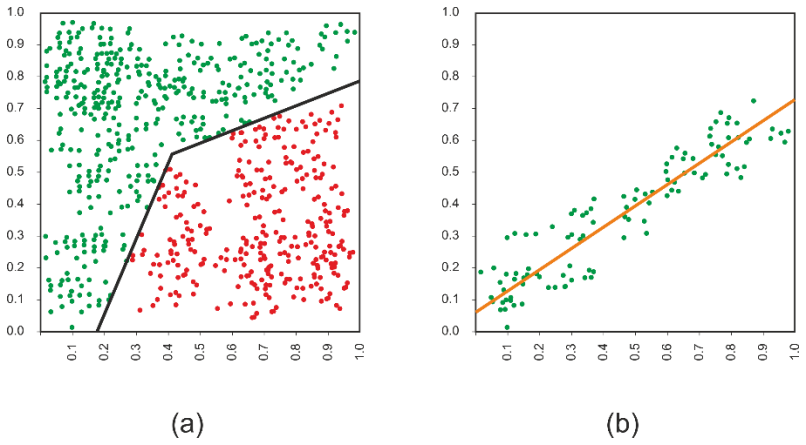
2.2. Genel Kavramlar

Makine öğrenmesinde öğrenme yöntemleri; denetimli öğrenme, denetimsiz öğrenme, yarı denetimli öğrenme ve pekiştirmeli öğrenme olarak gruplandırılmaktadır.

Denetimli öğrenme, eldeki verinin etiketlendiği ve çıkışta ne istediğimizi bildiğimiz öğrenme türüdür. Hem giriş hem de çıkış parametreleri belirlidir. Algoritma makul bir performans seviyesine ulaşana kadar, veri seti üzerinden çıkarımlar yapar, sonrasında öğrenme sonlanır (Aydın & Özkul, 2015).

Denetimli öğrenme, istenen çıktı durumuna göre sınıflandırma ve regresyon olarak gruplandırılmaktadır. Sınıflandırma probleminde, veri, aranan kritere uyup uymadığı durumuna bakılarak gruplara ayrılır. Örneğin gelen e-postaların spam olup olmadığını belirlemek için kullanılabilir.

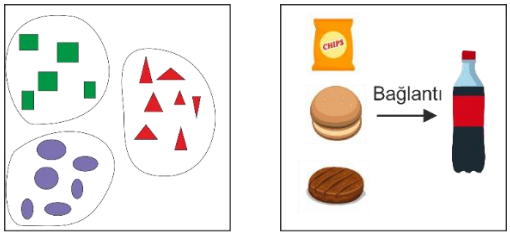
Regresyon ise, istenen özellik sayısal bir değer ve süreklilik içeriyorsa tercih edilir. Bağımsız ve bağımlı değişken veya değişkenler arasındaki ilişkiyi incelemek için kullanılmaktadır. Bağımlı ve bağımsız değişkenlerin modelde bir tane olması durumunda tek değişkenli regresyon, birden çok bağımlı ve bağımsız değişken bulunması durumunda ise çok değişkenli regresyon modelleri olarak isimlendirilmektedir (Sel, 2020). Örnek olarak, ev fiyatlarının tahmin edilme sürecinde evin büyüklüğü, konumu, yaşı gibi değerlere bakılarak evin değerinin tahmin edilmesi durumu verilebilir. Eşit bir örnek üzerinden aralarındaki farkı açıklamak gerekirse; hava durumu üzerinden sınıflandırma algoritması ile ertesi günün sıcak ya da soğuk olacağı bilgisi elde edilebilirken, regresyon algoritması ile ertesi günün hava sıcaklığı santigrat cinsinden tahmin edilebilir.



Şekil 2.4. Sınıflandırma (a) ve Regresyon (b)

Denetimsiz öğrenme de ise veriler etiketsiz olarak verilmektedir. Sistem herhangi bir etiket ile yönlendirme olmadan ham verileri organize verilere dönüştürerek öğrenmeye çalışmaktadır. Eldeki veriden tam olarak ne istediğimizi bilmediğimiz durumlarda, veri üzerinde bir örüntü yakalamak için kullanılmaktadır (Xu, Wang, Li, Ouyang & Shao, 2021). Örneğin kedi ve köpek fotoğrafları karışık olarak verildiğinde,

denetimsiz öğrenme ile bunlar arasındaki örüntü farkı algılanıp öğrenilebilir ve gruplandırılabilir, ancak çıkışta gruplandırılan bu görsellere kedi veya köpek etiketi verilemez. Problem çözümünde ihtiyaca göre kümeleme ve bağlantı (associaton) yöntemlerini kullanılır.



Şekil 2.5. Kümeleme (a) ve Bağlantı (b)

Kümeleme yönteminde, veri setindeki örüntüye göre veri gruplandırılmaktadır. Bağlantı yönteminde ise çok büyük miktarda olan veri setinden, veriler arasındaki bağlantı üzerinden çıkarımda bulunarak, tahminler yapabilecek bir öğrenme durumu gerçekleşmektedir. Örneğin, market alışverişinde patates cipsi, hamburger ekmeği ve köfte alan bir kişinin içecek olarak kola alma eğiliminde olacağı çıkarımı yapılabilir.

Yarı denetimli öğrenme modelinde, veri setinde küçük bir miktar etiketli veri ile büyük miktarda etiketsiz veri kullanılarak öğrenme işlemi gerçekleştirilmektedir. Büyük miktarda veri ile eğitimin gerçekleştirilmesi, modelin öğrenme başarısını olumlu yönde etki etmektedir. Ancak yüksek miktarda verinin etiketlenmesi zaman ve maliyet olarak denetimli öğrenme sürecini olumsuz yönde etkilemektedir. Yarı denetimli öğrenme ile denetimsiz öğrenmenin dezavantajı sayılabilecek veri etiketleme yükü azaltılarak aynı zamanda denetimli öğrenmenin doğruluk oranı avantajı bir araya getirilmiştir (Lan, Zou, Kang & Wu, 2021).

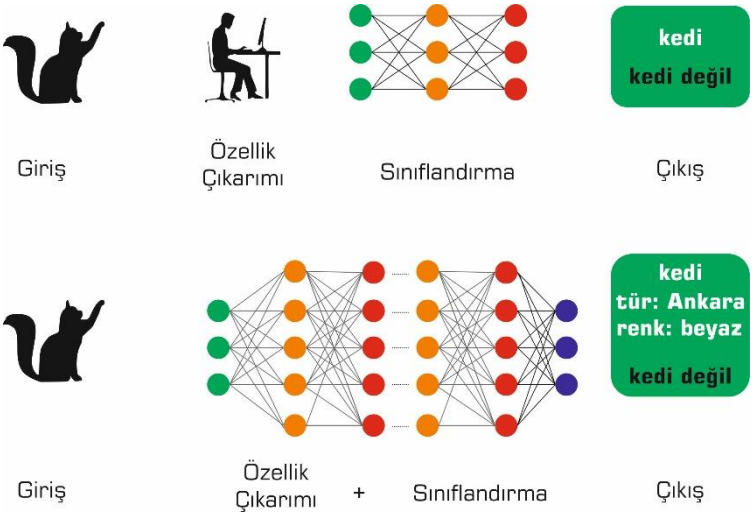
Pekiştirmeli öğrenmede, denetmen kişi eğitim süresince çıktıları değerlendirir. Çıktının doğru veya yanlış olması durumuna göre sistemi ödüllendirerek veya cezalandırarak ağırlıkları etkiler (Mekrache, Bradai, Moulay & Dawaliby, 2021).

BÖLÜM 3

3. DERİN ÖĞRENME

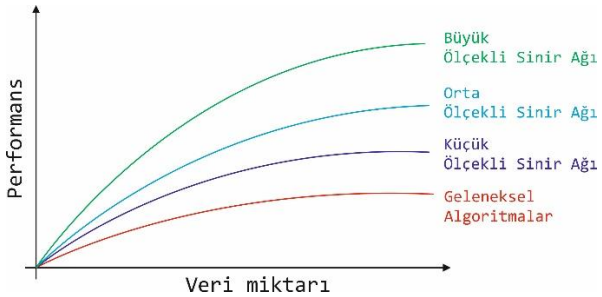
Derin öğrenme, yapay sinir ağları katman yapısını kullanır. İnsan beyninin öğrenme süreçlerini matematiksel modeller ile taklit ederek verileri işler ve anlamsal kavramlar oluşturur (Hu, Pang & Shi, 2016). Her bir katman diğer katman ile bağlantılı, çok katmanlı derin bir yapıya sahip olduğundan derin öğrenme olarak adlandırılmaktadır. Makine öğrenmesinin bir alt dalı olarak kabul edilmektedir. Makine öğrenmesinden farklı olarak, derin öğrenmenin yapay sinir ağları katman yapısında birçok gizli katman kullanılmaktadır. Herhangi bir özellik çıkarımı yapılmadan çok yüksek miktarda veri ile eğitim gerçekleşir.

Eldeki veri setleri kullanılarak eğitilen algoritma, çok katmanlı yapay sinir ağları ile kendi kurallarını oluşturmakta ve istenilen bilgiyi ayrıştırabilmektedir. Geleneksel yöntemlere göre derin öğrenme algoritmasının eğitilmesi için çok büyük boyutlarda veri setleri gerekmektedir. Veri seti büyüklüğü arttıkça, eğitilen algoritmanın çalışma doğruluğu ve verimi artmaktadır (Tiken, 2015).



Şekil 3.1. Makine Öğrenmesi (üstte) ve Derin Öğrenme (altta)

Derin öğrenmede 90'lı yıllarda donanım gücü yetersizliği sebebiyle çıkmaza girilmişse de son on yıl içerisinde GPU işlem gücünün artması ile birçok alanda kullanılmaya başlanmıştır. Derin öğrenme ile, bilgisayar oyunları, konuşma duygu tanıma, doğal dil işleme, sürücüsüz araç gibi alanlarda başarılı sonuçlar elde edilmiştir (Lauriola, Lavelli & Aiolli, 2021), (Gupta, Anpalagan, Guan & Khwaja, 2021), (Fahad, Ranjan, Yadav & Deepak, 2021), (Possemiers & Lee, 2021). Şekil 3.2'de gösterildiği gibi veri boyutu arttıkça sinir ağının büyüklüğünün de artması performansı olumlu yönde etkilemektedir. Küçük bir sinir ağı, küçük miktarda veri üzerinde, geleneksel sınıflandırıcılardan daha iyi sonuç vermekle birlikte, büyük boyutlu sinir ağları, büyük miktarda veri üzerinde daha yüksek doğruluk performansı elde edebilmektedir (Dong, Wang & Abbas, 2021).



Şekil 3.2. Derin sinir ağlarının veri boyutu ile performans ilişkisi (Dong vd, 2021)

3.1. Yapılan Çalışmalar

Son yıllarda derin öğrenme algoritmalarının temel alınmasıyla yapılan çalışmalar aşağıdaki başlıklarda incelenebilir (Dong vd, 2021).

- Doğal dil işleme
- Konuşma tanıma
- Sağlık uygulamaları
- Anlamsal bölütleme
- Sahne etiketleme
- Yüz tanıma
- Nesne tanıma
- Video nesne bölütleme
- Ön plan, arka plan ayrıştırma
- Grafik temelli uygulamalar
- Akıllı ulaşım sistemleri
- IoT uygulamaları
- Bilgisayar ağları alanında uygulamalar

Çizelge 3.1. Derin öğrenme ile yapılan çalışmalar

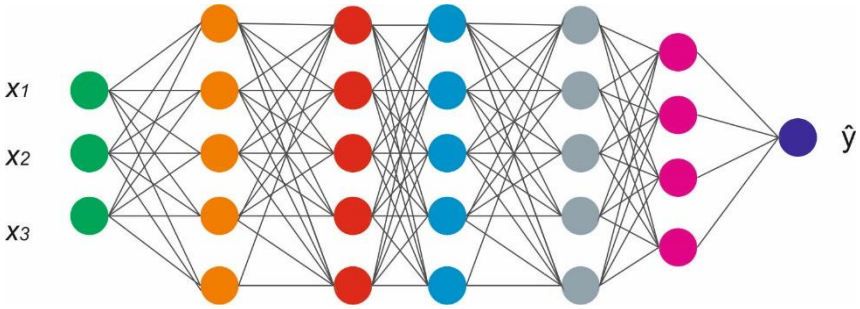
Uygulama Alanı	*Referans*	*Yöntem*	*Uygulama*
Doğal dil işleme	Kasthuri & Balaj (2021)	RNN - Uzun Kısa – Süreli Bellek (Long Short-Term Memory - LSTM)	Derin öğrenme sohbet robotu
	Alhogail & Alsabih (2021)	Grafik Evrişimsel Sinir Ağları (graph convolutional network -GCN)	Kimlik avı e-postası algılama
	Plappert, Mandery & Asfour (2018)	RNN	Vücut hareketi ile dil arasında çift yönlü bir ilişki çalışması
Konuşma Tanıma	Bertini, Allevi, Lutero, Calza & Montesi (2022)	Autoencoder	Günlük konuşma diline dayalı Alzheimer hastalığı sınıflandırma
Sağlık Alanında Uygulamalar	Shambhu, Koundal, Das & Sharma (2022)	CNN	Covid-19 ct görüntülerininin ikili sınıflandırması ile Covid teşhisi
	Hassan vd. (2021)	SVM, Adaboost, K-NN	Ultrason ve MRI görüntüleri ile Prostat kanseri sınıflandırması
	Eweje vd. (2021)	EfficientNet deep learning mimarisi	Kemik lezyonlarının MRI görüntüleri kullanılarak sınıflandırılması
	Jiang, Zhai & Kong (2021)	DDU-net	DDU-net ile beyin tümörü segmentasyonu
	Sranya, Priya, Jayalakshmi & Pavithra (2021)	CNN	Derin öğrenme ile beyin tümörü sınıflandırma
	Yildirim, Bozdag, Talo, Yildirim, Karabatak & Acharya (2021)	CNN	CT görüntülerinden Böbrek böbrek taşı tespit edilmesi
	Wai, Haishan, alpers, Rak & Hansen (2021)	CNN	2B ve 3B CT/MR görüntülerinden karaciğer tümörü ablasyonu
	Grassman vd. (2018)	CNN	Yaşa bağlı makula dejenerasyou için şiddet ölçeği çalışması
	Koh, Yoon, Kim, Han & Kim (2021)	CNN	CT üzerinden meme kanseri tespiti
Anlamsal Segmentasyon	Jeon, Kim, Park, Kwak & Choi (2021)	CNN	Deniz otlaklarının drone görüntüleri segmantasyonu
	Han, Dong & Yang (2021)	KNN	MLS Nokta bulutlarının anlamsal segmantasyonu
	Hacıefendioğlu, Başağa, Yavuz & Karimi (2021)	CNN	Rüzgar türbini kanatlarında akıllı buz tespiti

Çizelge 3.1. Devamı

Uygulama Alanı	*Referans*	*Yöntem*	*Uygulama*
Yüz Tanıma	Iqbal, Sameem, Naqvi, Kanwal & Ye (2019)	CNN	Açısal olarak ayırd edici özelliklere dayalı yüz tanımlaması
	Dhomne, Kumar & Bhan (2018)	CNN	Yüz tanıma ile cinsiyet tespiti
	Kumar, Kasiviswanadham, Indira, Palesetti & Bhargavi (2021)	MTCNN	Suçlu yüz tanıma sistemi
	Alhanaee, Alhammadi, Almenhali & Shatnawi (2021)	CNN	Katılımcı yüz tanıma sistemi
Nesne Tanımlama	Kakehi vd. (2021)	ResNeXt	Pasifik istiridyesi "Crassostrea gigas larvare" tanıma ve sayma
	Hespeler, Nemati & Niri (2021)	CNN	Robotik hasat sistemleri için biber nesne tanımlama termal görüntüleme
	Pathak vd. (2018)	CNN	Derin öğrenme ile nesne tanımlama uygulaması
	Loh, Yong, Yapeter, Subburaj & Chandramohanadas (2021)	MRCNN	Sıtma enfeksiyonun taranmasına yönelik nesne tanıma ile hızlı hücre sayımı
	Jeon, Lee, Shin, Jang & Kim (2019)	R-CNN	Su altı nesne tanıma
Akıllı ulaşım sistemleri	Li, Yang, Qu, Cao & Li (2021)	CNN	Gece otonom sürüş için görüntü iyileştirme yaklaşımı
	Fujiyoshi, Hirakawa & Yamashita (2019)	CNN	Otonom sürüş için görsel tanıma
	Zhao, Qu & Xu (2020)	DQN	Otoyolda otonom sürüş için DRL yaklaşımı
	Huang, Wen, Fu, Peng & Tang (2020)	FCNN, LSTM, 3D CNN	Demiryolu sistemlerinde gecikme tahmini
	Zhu & Li (2021)	SWRNN	Yakıt yükleme kararları için uçuş süresi tahmini
	Yu, Guo, Asian, Wang & Chen (2019)	DBN	Uçuş rötar tahmini

3.2. Derin Sinir Ağları

Derin sinir ağları ile çok katmanlı yoğun bir yapıya sahip sinir ağı modeli oluşturularak öğrenme sürecinin en iyi şekilde optimize edilmesi amaçlanmaktadır. Her katmanda veri üzerinde bir özellik çıkarımı yapılmaktadır. Böylelikle katman sayısının artmasıyla daha çok özellik çıkarımı mümkün olmaktadır. Daha çok özellik çıkarımı ile görselde bir nesnenin diğerlerinden ayrılması, uygun koşullar sağlandığında, daha kolay olmaktadır.



Şekil 3.3. Derin sinir ağı örneği

Şekil 3.3'te katman sayısı L, L katmanındaki nöron sayısı $n^{[l]}$, girişler ile ağırlık değerinin bias değeri çarpım sonucu çıkış değeri olan z değeri aktivasyon fonksiyonuna (f) tabi tutulduğunda L. katmanındaki a değeri elde edilmektedir.

Her hücre diğer hücreler ile birbirine bağlanarak katman yapısı oluşturur. Gizli katman sayısı tek ise tek katmanlı, birden fazla ise çok katmanlı yapay sinir ağları olarak isimlendirilir.

$$a^{[l]} = f^{[l]}\,(z^{[l]}) \tag{3.1}$$

Şekil 3.3'te giriş katmanı, toplam katmanlara dahil edilmemektedir. Bu durumda çıkış katmanı dahil L=6 olmak üzere 6 katmanlı bir ağ yapısı mevcuttur. Her katman için nöron sayısı; $n^{[1]} = 5$, $n^{[2]} = 5$, $n^{[3]} = 5$, $n^{[4]} = 5$, $n^{[5]} = 4$, $n^{[6]} = 1$ şeklinde ifade edilmektedir.

Sinir ağı çeşitleri, İleri beslemeli (FF), Tekrarlayan sinir ağları (RNN), Derin ileri beslemeli (DFF), Uzun/kısa süreli bellek (LSTM), Yarıçapsal temelli ağ (RBF), Kapılı

tekrarlayan hücre (GRU), Markov ziniri (MC), Hopfield ağı (HN), Boltzman makinesi (BM), Derin inanç ağları (DBN), Evrişimsel sinir ağı (CNN) şeklinde sıralanabilir.

3.3. Geri Besleme

Çıkış değeri ile istenen değer arasındaki sapmayı hesaplayıp, bu sapma değerinin yönü ve miktarına göre ağırlıkların yeniden düzenlenmesi, geri yayılım algoritması ile sağlanmaktadır. Rumelhart, Hinton ve Williams'ın 1986 yılında yaptıkları çalışma ile kullanımı artmıştır (Rumelhart, Hinton & Williams, 1986). Çıkıştan itibaren her ağırlık için türev hesabı yapılarak ağırlıklar güncellenmektedir. Denklem 3.2'de türev hesabının formülü verilmiştir.

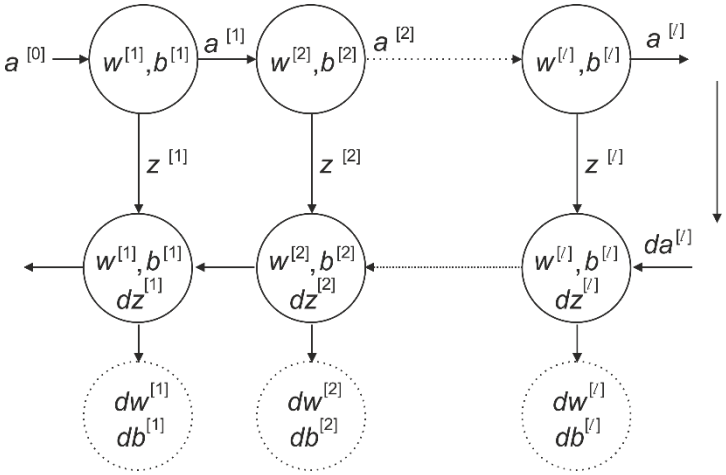
$$w \leftarrow w - \alpha \frac{\partial L(z,y)}{\partial w} \tag{3.2}$$

İleri beslemeli yapıda $l.$ katmanın giriş değeri bir önceki katmanın çıkış değeri olan $a^{[l-1]}$ değeridir. $l.$ katmanın ara çıkış değeri olan $z^{[l]} = w^{l}.a^{[l-1]} + b^{[l]}$ ile hesaplanır. $l.$ katmanın nihai çıkış değeri $z^{[l]}$ değerinin, $f^{[l]}$ aktivasyon fonksiyonuna tabi tutulması ile elde edilmektedir.

$l.$ Katmanın geri yönde beslemesi için elde edilen $a^{[l]}$ değerinin türevi alınmaktadır. $a^{[l]}$ değerinin türevinin alınmasıyla l katmanının ağırlık değerlerinin ve bias değerinin türevi elde edilmektedir. O katmana ait z değerinin türevini elde etmek için $da^{[l]}$ ile $f^{[l]}\ (z^{[l]})$ değeri evrişime tabi tutulmaktadır.

$$dz^{[l]} = da^{[l]} * f^{[l]}\ (z^{[l]}) \tag{3.3}$$

Geri yönde besleme ile son çıkışta elde edilen yitim değerinin minimuma indirgenmesi hedeflenmektedir. Böylece çıkışların türevleri ile önceki katmanının ağırlık değerleri güncellenmiş olmaktadır.



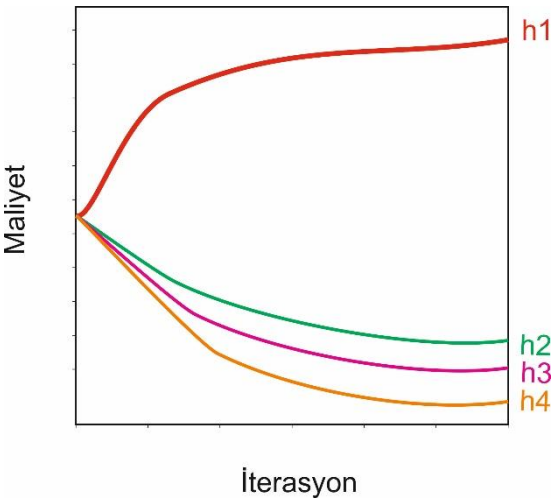
Şekil 3.4. İleri ve Geri besleme

BÖLÜM 4

4. HİPER-PARAMETRELER

Hiper-parametreler, veri setine ve donanım yapısına bağlı olarak değişebilen parametrelerdir. Ağırlık değerleri, bias değerleri gibi değerler model için parametreler olarak değerlendirilirken, öğrenme oranı, iterasyon sayısı, gizli katman sayısı, aktivasyon fonksiyonu çeşidi, küme boyutları ve regülarizasyon yöntemleri hiper-parametreler olarak değerlendirilmektedir. Derin öğrenme modelini tasarlarken bu hiper-parametreleri göz önünde bulundurmak gerekmektedir. Her modelde her zaman aynı hiper-parametre değerleri uygun sonuç vermeyebilmektedir. Derin öğrenme model yapısı çok kararlı bir yapıya sahip olmadığından, model eğitiminde deneme yanılma yoluna da başvurulabilmektedir. Ancak kimi zaman aranan optimize değerleri bulmak, hiper-parametre havuzu çok geniş olabileceğinden mümkün olamayabilmektedir. Bu gibi durumlarda parçacık sürü optimizasyonu, diferansiyel gelişim algoritmaları ve genetik algoritmalar gibi sezgisel yöntemlerden faydalanılabilmektedir (Gülcü & Kuş, 2019).

Hiper-parametre belirlenirken maliyet ve iterasyon arasındaki ilişkiye bakılır. İterasyon sayısı ilerledikçe maliyet artıyorsa hiper-parametrelerin yeniden düzenlenmesi gerekmektedir. Şekil 4.1'de verilen grafikte h1 hiper-parametreleri ile maliyetin arttığı gözlenmektedir. Bu durumda h1 parametreleri ile öğrenme gerçekleşmiyor anlamına gelmektedir. Örnek olarak gizli katman sayısının değiştirildiği, farklı aktivasyon fonksiyonları ile h2, h3, h4 hiper-parametrelerinin iterasyon maliyet durumunun değiştiği gözlenmiştir.



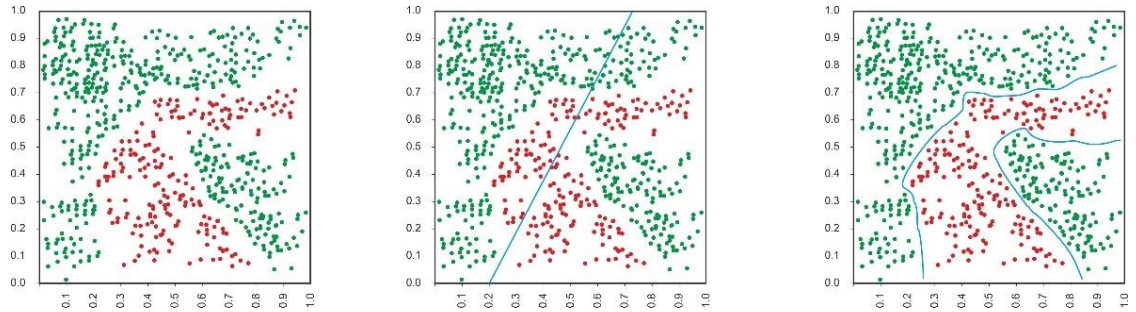
Şekil 4.1. Hiper-parametrelerde iterasyon - maliyet ilişkisi

4.1. Aktivasyon Fonksiyonu

Öğrenmenin verimli şekilde gerçekleşebilmesi için lineer olmayan bir aktivasyon fonksiyonuna ihtiyaç duyulmaktadır. Sinir ağının doğrusal olmayan durumları da öğrenmesi beklenmektedir. Aktivasyon fonksiyonunun amacı gerçek hayatta karşılaşılabilecek video, görüntü, ses gibi karmaşık durumlara uyum sağlayabilecek şekilde öğrenme işleminin gerçekleştirilmesini sağlamaktır.

Şekil 4.2'de (a) durumunda verilen kırmızı ve yeşil noktaların sınırlarının çizilmesi için lineer bir fonksiyon kullanılması durumunda sonuç (b) durumundaki gibi olacaktır. Doğrusal fonksiyonların çıktısı da doğrusal olmaktadır. Birden fazla dereceye sahip fonksiyonlar ile (c) durumundaki doğrusal olmayan bir ayrım yapmak mümkündür. Bu şekilde yapay sinir ağının öğrenim kapasitesi çok daha karmaşık sorunları çözebilecek seviyeye ulaşmaktadır (Apicella, Donnarumma, Isgro & Prevete, 2021).

Sinir ağlarında kullanılan birçok aktivasyon fonksiyonu vardır. Aktivasyon fonksiyonu seçerken, ağın derinliği, ağın işlem kapasitesi, fonksiyonun doğrusal olmaması, türevlenebilir olması, monoton artan veya azalan olması, alt ve üst sınırlarının olması, orijin durumunda kendine yakınsaması veya kullanılacağı katmanın hangi konumda olduğu durumlarının değerlendirilmesi gerekir (Mhaskar & Micchelli, 1993).



Şekil 4.2. Ham veri (a), Lineer aktivasyon fonksiyonu ile sınıflandırma (b), Lineer olmayan aktivasyon fonksiyonu ile sınıflandırma (c)

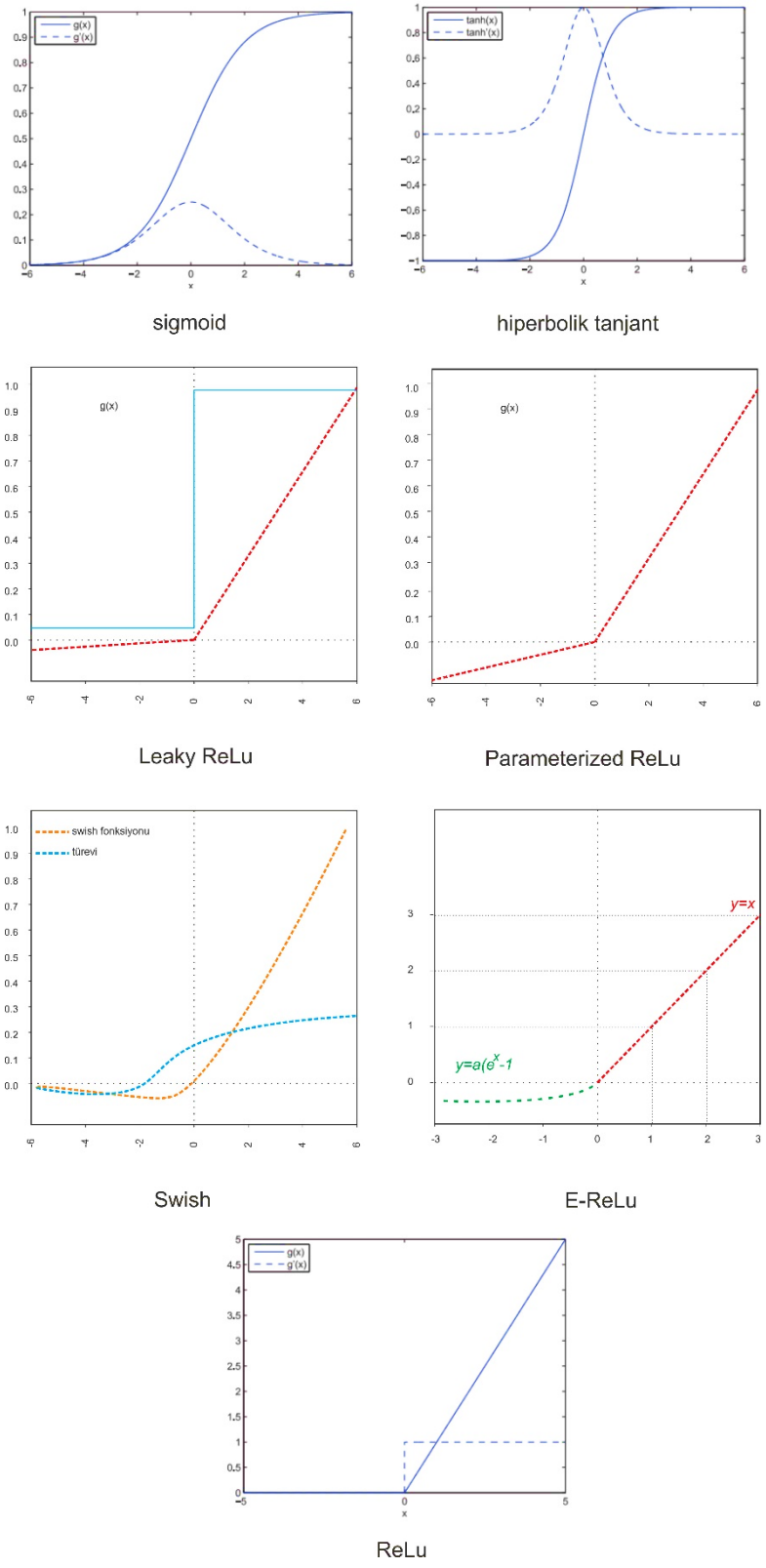
En sık tercih edilen aktivasyon fonksiyonlarından biri sigmoid fonksiyonudur. Sigmoid fonksiyonu türevlenebilir bir fonksiyondur ve çıkış aralığı [0,1] aralığında olduğu için çıktının hangi sınıfa ait olduğuna dair bir sonuç verir. Çok katmanlı ağlarda tercih edilmemesinin sebebi, türevinin uç noktalarda sıfıra yakınsamasıdır. Bu durum "vanishing gradient" olarak tanımlanmaktadır ve geri yayılım esnasında öğrenmenin yavaşlamasına veya durmasına sebep olabilmektedir (Abuqaddom, Mahafzah & Faris, 2021).

Softmax fonksiyonu ile sigmoid fonksiyonu birbirine benzer yapıdadırlar. Softmax fonksiyonu, sınıflandırma konusunda oldukça iyi bir performans sergilemektedir. İkiden fazla sınıflandırma yapılması gereken derin öğrenme modellerinin çıkış katmanlarında tercih edilmektedir.

Hiperbolik tanjant fonksiyonu da Sigmoid fonksiyonuna benzer bir yapıdadır. Çıkış aralığı [-1,1] aralığındadır. İkili sınıflandırma için tercih edilebilecek bir fonksiyondur ancak türevinin uç noktalarda sıfıra yakınsaması sebebiyle sigmoid fonksiyonu gibi çok katmanlı ağlarda tercih edilmesi sakıncalı olabilmektedir.

Çizelge 4.1. Aktivasyon fonksiyonları

Aktivasyon fonksiyonu	*Denklem*	*Aralık*
Sigmoid	$f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$	*(0,1)*
Hiperbolik Tanjant	$f(x) = \tanh(x) = \frac{(e^{x}-e^{-x})}{(e^{x}+e^{-x})}$	*(-1,1)*
ReLu	$f(x) = \begin{cases} 0 \text{ için } x < 0 \\ x \text{ için } x \geq 0 \end{cases}$	*[0, ∞)*
Leaky ReLu	$f(x) = \begin{cases} 0.001 \text{ için } x < 0 \\ x \text{ için } x \geq 0 \end{cases}$	*(-∞, ∞)*
Exponential Linear Unit	$f(x) = \begin{cases} x \text{ için } x \geq 0 \\ a(e^{x-1}) \text{ için } x < 0 \end{cases}$	*(-∞, ∞)*
Swish	$f(x) = 2x\sigma(\beta x) = \begin{cases} \beta = 0 \text{ için } f(x) = x \\ \beta \to \infty \text{ için } f(x) = 2max(0,x) \end{cases}$	*(-∞, ∞)*



Şekil 4.3. Aktivasyon fonksiyonları

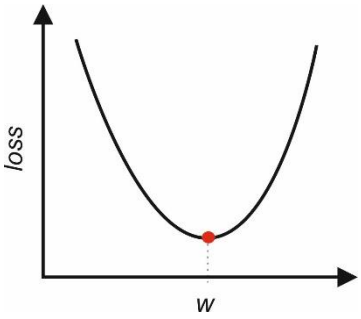
Çok katmanlı sinir ağlarında, özellikle de evrişimli sinir ağlarında tercih edilen bir aktivasyon fonksiyonu olan ReLu, aynı anda tüm nöronları aktive etmediğinden ara katmanlarda tercih edilmektedir. [0, +∞] aralığında değer almaktadır. Hiperbolik tanjant ve sigmoid fonksiyonları gibi türevi iki uç noktada sıfıra yakınsamamasına rağmen, -1 durumunda sıfıra yakınsar. Bu durumda da önceki bahsedilen aktivasyon fonksiyonlarında olduğu gibi geri yayılım esnasında öğrenme işlemi duraksamaktadır. Bu durum "Dying ReLu" olarak isimlendirilmektedir (Lu, Shin, Su & Karniadakis, 2020). ReLu fonksiyonu üzerinde yapılan küçük değişiklikler ile oluşturulan Leaky ReLu, ReLU ve Exponential Linear Unit gibi fonksiyonlar ile bu durumun üstesinden gelinmeye çalışılmıştır.

Leaky ReLu, adından da anlaşılacağı üzere, negatif kısımda sıfıra yakınsar ancak bir sızıntıya benzer şekilde tam sıfır olmaz. Bu durumda geri yayılım esnasında öğrenme süreci devam etmektedir (Liu, Wang, Wang & Liu, 2019). Parameterized ReLu ve Exponential Liner Unit fonksiyonları da Leaky ReLu fonksiyonuna benzer yapıdadırlar.

Swish fonksiyonu da negatif bölgede yer almaktadır ancak ReLu fonksiyonlarından farklı olarak swish fonksiyonu negatif bölgede doğrusal değildir.

4.2. Gradyan İnişi

Makine öğrenmesinde öğrenme durumunun doğruluğu için hata (loss) değerinin düşük olması istenir. Gradyan inişteki amaç, en uygun ağırlık değerini bulup hata değerlerini aza düşürmektir (Abuqaddom vd, 2021). Gradyan iniş fonksiyonu iteratif bir fonksiyon olup öğrenme sürecinde birden fazla defa çalışmaktadır. Fonksiyon üzerinde rastgele bir noktadan çalışmaya başlayarak adım adım fonksiyondaki en düşük noktaya ulaşmaya çalışır. Şekil 4.4'teki grafikten anlaşılacağı üzere ağırlık değerinin optimum seviyeden az veya fazla olması hata değerini arttırmaktadır.



Şekil 4.4. Gradyan inişi

Momentumlu gradyan inişi fonksiyonunda önceki adımların bilgisi ile bir sonraki adımın yönü tahmin edilir. Bu şekilde y yönündeki salınımlar kısıtlanarak, hedefe daha hızlı bir şekilde ulaşılabilmektedir (Seyyarer, Ayata, Uçkan & Karci, 2020).

4.3. Yığın Boyutu – Batch Size

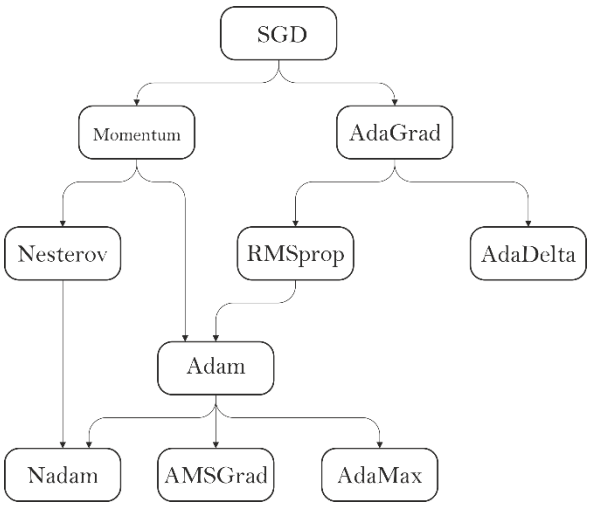
Yığın boyutu, aynı anda işlenecek veri miktarıdır. Yığın boyutu arttıkça loss değeri artar eğitim süresi düşer. Yığın boyutu düştükçe aynı anda işlenecek olan veri miktarı da düşer. Her seferinde veri yenileneceğinden dolayı öğrenim sürecinde gürültü oranı artar (Kandel & Castelli, 2020). Yığın boyutu seçilirken GPU üzerindeki bellek boyutu dikkate alınmaktadır. Bu sebepten yığın değerleri 2^n şeklinde belirlenir. Yığın boyutu değerindeki küçük değişiklikler evrişimli sinir ağlarında oldukça farklı etkiler oluşturabilmektedir (Tian, Xu & Zuo, 2020).

Evrişimli sinir ağlarında yığın boyutu değerinin yüksek verilmesi durumunda aşırı öğrenme olarak da bilinen, ezberleme (overfitting) durumları ile karşılaşılabilmektedir (Mutasa, Sun & Ha, 2020).

- **Toplu Gradyan İnişi**: Örnekleme sayısı ile yığın boyutu değeri birbirine eşittir. Örneğin 1000 adet veri içeren bir veri seti için yığın boyutu 1000 seçilirse 1 tur (epoch) içerisinde 1000 adet verinin tamamı işleme alınır.
- **Mini-Toplu Gradyan İnişi:** Bu yöntemde yığın boyutu 1 ile veri sayısı arasında seçilir. Örneğin 1000 adet veri için yığın boyutu 20 belirlendiğinde, ağ her seferinde 20 adet veriyi işleyecektir. Bu durumda tur sayısı da 50 olacaktır.

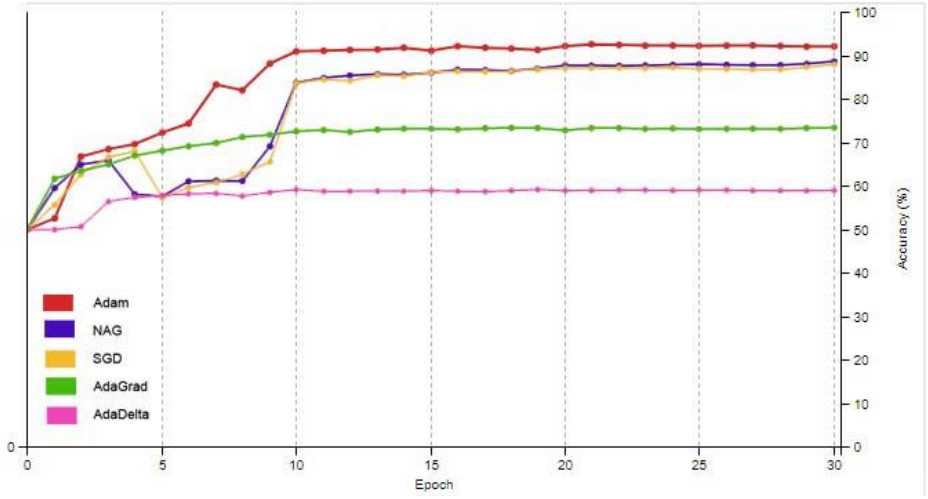
4.4. Eniyileyici Algoritmaları

Makine öğrenmesi doğrusal olmayan problemlere en verimli çözümü üretmeyi hedeflemektedir. Bunun için hata oranı en aza indirgenmeli ve optimum değer ile çözüme ulaşılmalıdır. Bu amaçla kullanılan bazı eniyileyici algoritmaları vardır. Yaygın olarak kullanılan eniyileyici algoritmaları Stochastic Gradient Descent (SGD), AdaDelta, Adaptive Gradient (AdaGrad), Adaptive Moment (AdaM), Nesterov's Accelerated Gradient (NAG), RMSprop olarak sayılabilir (Seyyarer vd, 2020). Bu eniyileyici algoritmaları özünde gradyan azaltma işlemini temel almaktadır. Bu algoritmalar arasındaki ilişki Şekil 4.5'te gösterilmektedir.



Şekil 4.5. Eniyileyici algoritmaları arasındaki ilişki haritası (Seyyarer vd, 2020)

Şekil 4.6'te 2000 adet test, 8000 adet eğitim görseli olmak üzere toplam 10000 adet kedi ve köpek görselleri veri seti üzerinde eniyileyici algoritmalarının eğitim doğruluk grafiklerinin karşılaştırması verilmiştir. CAFFE ortamında GoogLeNet ağı ile gerçekleştirilen bu testte en iyi doğruluk sonucunu Adam algoritmasının verdiği görülmüştür.



Şekil 4.6. Eniyileyici türleri kedi-köpek veri seti üzerindeki eğitiminin karşılaştırması

4.4.1. Stochastic Gradient Descent (SGD)

Gradyan inişinin bir çeşidi olan bu yöntem ile model parametreleri daha sık güncellenebilmektedir. Gradyan iniş yönteminde veri kümesindeki her noktadan türev alınarak hesaplama yapılmaktadır. Bu sebeple gradyan iniş büyük verilerde oldukça yavaş çalışmaktadır. SGD ise veri hesaplama süresini azaltmak için her yinelemede tüm veri kümesinden rastgele bir nokta seçer. Burada stokastik (stochastic) kelimesi rastgele bir olasılıkla bağlantılı sistem veya süreç anlamına gelmektedir (Wang & Knap, 2020).

SGD resim tanımada kötü sonuç vermesine ve diğer yöntemlere göre yavaş çalışmasına rağmen momentum ile kullanıldığında yüksek başarı sağlayabilmektedir (Wilson, Roelofs, Stern, Srebro & Recht, 2018). SGD'nin momentum ile kullanılması optimum nokta arayışındaki gürültü salınımını önlemektedir (Seyyarer vd, 2020).

4.4.2. Momentum

Gradyan yerine momentum kullanılması, SGD yönteminde meydana gelen salınımları en aza indirgemek için önerilmektedir. Denklemde verilen β değeri 0 – 1 arasında seçilmektedir. β değeri geçmiş gradyanların işlem üzerindeki ağırlığını belirlemek için kullanılmaktadır (Seyyarer vd., 2020).

$$w_{t+1} = w_t - aV_t \tag{4.1}$$

$$V_t = \beta V_{t-1} + (1-\beta)\frac{\partial L}{\partial w_t} \tag{4.2}$$

4.4.3. Adaptive Gradient – AdaGrad

AdaGrad, gradyan inişinin uzantısı olarak adaptif bir algoritmadır. Öğrenme sürecinde değişkenler için gerekli olan öğrenme oranlı farklılık gösterebilmektedir. Optimum değere yaklaşıldığında daha düşük bir öğrenme hızı yeterliyken, optimum değerden çok daha uzak olunan durumlarda daha büyük bir öğrenme oranına ihtiyaç duyulabilmektedir. AdaGrad, öğrenme oranını parametrelere göre ayarlayabilmektedir. Örneğin sık olmayan parametreler için daha büyük güncellemeler gerçekleştirirken, sık parametreler için daha düşük güncellemeler gerçekleştirmektedir (Ruder, 2017). Resim tanıma işleminde kullanılmaya uygun bir algoritmadır. Her parametrenin kendi adaptif öğrenme hızı vardır. Öğrenme oranını manuel olarak ayarlama ihtiyacının ortadan kalkması bu algoritmanın faydalarından biri olarak görülür. Algoritmanın özelliklerine bağlı olarak bu öğrenme oranı azalmaktadır. Ancak bir noktadan öğrenme durumu ilerlememektedir (Zhang, Lei & Zhao, 2019). Bu ise Adagrad'ın olumsuz tarafı olarak görülmektedir.

Denklemde verilen S değeri başlangıçta 0 olarak alındığından, sıfıra bölüm konusunda hata meydana gelmemesi amacıyla ϵ için denklemde 10^{-7} gibi çok düşük bir değer seçilmektedir (Seyyarer vd., 2020).

$$w_{t+1} = w_t - \frac{a}{\sqrt{S_t + \epsilon}} . \frac{\partial L}{\partial w_t} \tag{4.3}$$

$$S_t = S_{t-1} + \left[\frac{\partial L}{\partial w_t}\right]^2 \tag{4.4}$$

4.4.4. Root Mean Square Propagation - RMSProp

AdaGrad yöntemi "belirli bir süre sonra öğrenme sürecinin ilerlememesi" problemini çözmek amacıyla önerilmiş bir yöntemdir (Hinton, Srivastava & Swersky, 2012). AdaGrad ve RmsProp başlangıç adımlarında birbirine benzer. Bu yöntemde her tur için dw ve db kullanmak yerine dw ve db karelerinin üstel ağırlıklı ortalamaları dikkate alınır.

$$S_{dw} = \beta . S_{dw} + (1 - \beta) . dW^2 \tag{4.5}$$

$$S_{db} = \beta . S_{db} + (1 - \beta) . db^2 \tag{4.6}$$

Bir sonraki Hiper-parametre değerleri geçmiş değerlerin ortalaması ile mevcut değerlerin karesi arasındaki ağırlığın atanmasıyla hesaplanır (Vani & Rao, 2019).

$$W = W - \frac{\text{öğ}renme\ katsayısı . dw^2}{\sqrt{S_{dw}}} \tag{4.7}$$

$$b = b - \frac{\text{öğ}renme\ katsayısı . db^2}{\sqrt{S_{db}}} \tag{4.8}$$

4.4.5. AdaDelta

Adaptif gradyan yöntemindeki, paydada biriken gradyanların öğrenim oranını azaltarak zamanla sıfıra indirgemesinin önüne geçebilmek için Hiper-parametre seçimine duyarlılığın üstesinden gelebilecek bir yöntem olan AdaDelta çözümü önerilmiştir (Zeiler, 2012). RMSprop yönteminin bir türevidir. Bu yöntem ile AdaGrad eniyileyicisinde bir sorun olan, öğrenme hızının düşüşü önlenmektedir (Vani vd., 2019).

Denklemde Keras üzerinde varsayılan değerler β için 0.95, ε için 10^{-6} olarak belirlenmiştir (Karim, 2018).

$w_{t+1} = w_t - \frac{\sqrt{D_{t-1}+\varepsilon}}{\sqrt{v_t+\varepsilon}} . \frac{\partial L}{\partial w_t}$ için; (4.9)

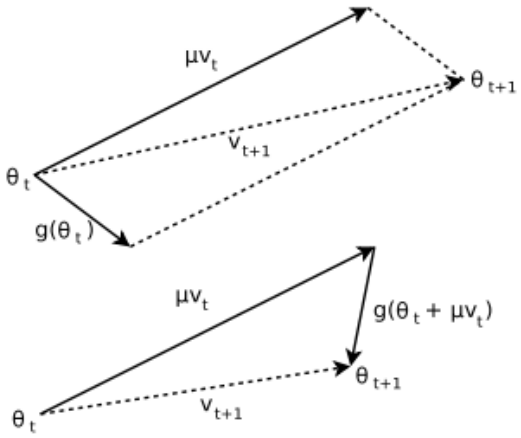
$$D_t = \beta D_{t-1} + (1-\beta)[\Delta w_t]^2 \quad (4.10)$$

$$v_t = \beta v_{t-1} + (1-\beta)\left[\frac{\partial L}{\partial w_t}\right]^2 \quad (4.11)$$

$$\Delta w_t = w_t - w_{t-1} \quad (4.12)$$

4.4.6. Nesterov's Accelerated Gradient – NAG

Nesterov'un dışbükey programlama problemi üzerine 1983 yılında yayınladığı ilk çalışma gradyan iniş algoritmalarının konusunu hedef alarak yapılmış bir çalışma olmamakla birlikte, Sutskever ve arkadaşları stokastik gradyan inişinde Nesterov'un algoritmasını uyarlamışlardır (Sutskever, Martens, Dahl & Hinton, 2013). Gradyan iniş algoritmalarında değişkenler, ilerideki noktalar hakkında öngörüde bulunmadan, gerideki ve o andaki noktalara dayalı olarak güncellenmektedir. Bu durum öğrenme sürecinin ve öğrenme performansının öngörülemez olmasına sebep olmaktadır. Şekil 4.7'de klasik momentum ve Nesterov hızlandırılmış gradyan vektör görüntüsü verilmiştir (Sutskever vd., 2013).



Şekil 4.7. Klasik momentum (üstte), Nesterov Hızlandırılmış Gradyan (altta)

NAG algoritması değişkenlerin güncellenmesinde yakın ileri bir noktayı sezgisel yaklaşımla ele alıp tahminde bulunarak, mevcut parametre değerlerini bu duruma göre ayarlamaktadır. Bu şekilde yakınsama hızlandırılmaktadır (Muehlebach & Jordan, 2019).

Değerlerin güncellenmesinde, denklemde verilen, öngörülen gradyanların üstel haraketli ortalaması olan m değeri kullanılmaktadır (Karim, 2018).

$$w_{t+1} = w_t - \alpha m_t \tag{4.13}$$

$$m_t = \beta m_{t-1} + (1-\beta)\frac{\partial L}{\partial w^*} \tag{4.14}$$

$$w^* = w_t - \alpha m_{t-1} \tag{4.15}$$

Denklemdeki *t* adımı için geri yayılım gerçekleştirilmeden önce bir ileri yayılım gerçekleştirilmesi gerekmektedir. Denklem (2.18)'de verilen w^* değeri, öngörülen gradyan değeridir. Denklem (4.15)'da ki biçimde bir adım ileri gidilerek elde edilmektedir. β için tercih edilen değer genellikle *0.9* olmaktadır (Karim, 2018)

4.4.7. Adaptive Moment Estimation– Adam

Bu model, Kingma ve Ba tarafından 2015 yılında önerilmiş olup öğrenme oranları her parametre için adaptif olarak hesaplanan bir diğer stokastik gradyan modelidir. Bu yöntem, seyrek gradyanlarda iyi çalışan AdaGrad ve sabit olmayan ortamlarda iyi sonuç veren RmsProp yöntemlerinin iyi yönlerini harmanlamıştır (Kingma & Ba, 2015).

Birinci derece gradyanları hesaplamak için ihtiyaç duyulan bellek miktarı küçüktür. Öğrenme sürecinde katlanarak azalacak olan önceki birinci derece gradyanların karesi ile birinci derece gradyanların kayıtlarını da tutar (Zhang, 2019). Önerilen yapı büyük veri setleri veya yüksek boyutlu parametre uzaylarını içeren makine öğrenmesi sorunlarını çözmeyi amaçlamaktadır. NAG yapısının momentum yapısına göre daha üstün olması sebebiyle, Adam yapısı içerisinde kullanılan vanilla momentumu yerine NAG tercih edilmesiyle iki yöntemin de avantajlarından faydalanılan modifiye bir yapı olan NADAM, Dozat (2016) tarafından önerilmiştir.

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{v_t}+\epsilon}}.\hat{m_t} \tag{4.16}$$

$$\hat{m_t} = \frac{m_t}{1-\beta_1^t} \tag{4.17}$$

$$\hat{v_t} = \frac{v_t}{1-\beta_2^t} \tag{4.18}$$

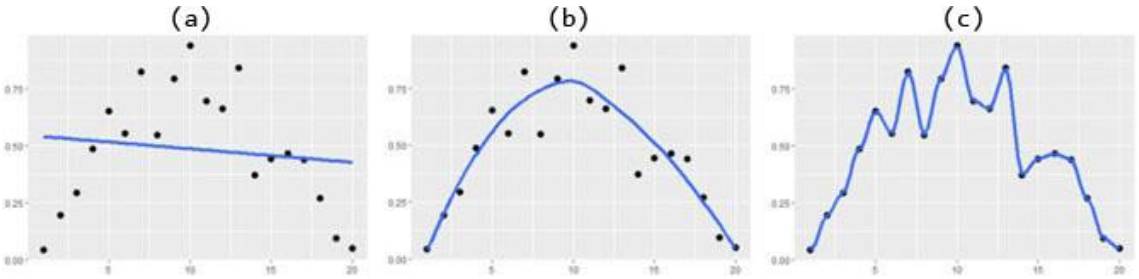
$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\frac{\partial L}{\partial w_t} \tag{4.19}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)\left[\frac{\partial L}{\partial w_t}\right]^2 \tag{4.20}$$

Denklem (4.16) için (4.17) ve (4.18) bias düzeltmeleridir. Önerilen varsayılan değerler ise şöyledir; $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$ (Ruder, 2017).

4.5. Aşırı Öğrenme – Overfitting

Aşırı öğrenme sorunu yapay zekâ alanında temel bir engel durum kabul edilmektedir. Model veri seti üzerinde gerekenden fazla eğitim gerçekleştirmişse bu durum ile karşılaşılmaktadır. Böyle bir durumda eğitim setinde yüksek bir skor elde edilmesine rağmen, test verileri ile çalışıldığında skor oranı düşük seviyelerde kalmaktadır. Laynman'dan aktaran Mutasa ve arkadaşları (2020), aşırı öğrenme durumunu, bir yapay zekâ modelinin yalnızca eğitim örneğine uygulanabilir şekilde öğrendiğini ve artık genel popülasyona genellenemeyeceği anlamına geldiğini belirtmiştir. Model eğitim setindeki verileri ezberlediğinden ve farklı bir veri üzerinde aynı deseni bulamayacağından ötürü çok kötü tahminler yapabilmektedir. Bunun sebebi ise değişkenler arasındaki gerçek ilişkiyi değil de, verilerdeki gürültüyü öğrenmesidir. Şekil 4.8'de yetersiz öğrenme, iyi öğrenme ve aşırı öğrenme durumları verilmiştir (Mutasa vd., 2020).

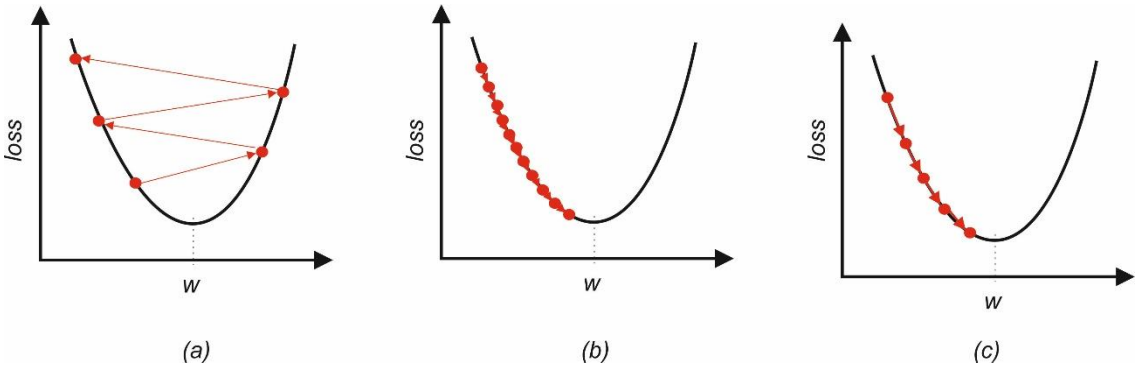


Şekil 4.8. Yetersiz Öğrenme (a), İyi Öğrenme (b), Aşırı Öğrenme (c)

4.6. Öğrenme Katsayısı – Learning Rate

Öğrenme oranı zaman ile ters orantılı ve zamanın lineer fonksiyonudur. Ağın parametrelerini ne kadar hızlı güncellediğini belirleyen bir hiper-parametredir. Öğrenme başarımı ile ters orantılı olduğundan daha yüksek bir öğrenme katsayısı ile daha zayıf bir öğrenme gerçekleşmektedir. Bu değer için, 0.01, 0.0001 gibi sabit bir değer belirlenebileceği gibi, adım adım artan bir değer de belirlenebilir. Öğrenme oranının yüksek seçilmesi yüksek salınıma sebep olacaktır. Bu durumda öğrenme durumu gerçekleşmeyebilir. Çok küçük seçilmesi de öğrenmenin süresini uzatmaktadır (Yu, Qi, Ma, He, Wang & Zhao, 2020). Bu nedenle öğrenme durumunun yüksek olabileceği mümkün olan en kısa sürede sonlanabilecek bir optimum değer seçilmesi gerekmektedir. En uygun değer için çeşitli eniyileyici algoritmaları kullanılmaktadır. Şekil 4.9'da gradyan iniş grafikleri farklı öğrenme oranlarına göre verilmiştir. Şekil 4.9'da (a) durumunda adım aralıkları çok büyük olduğundan hata değerinin azalması beklenirken artmaktadır. (b) durumunda ise adımlar çok küçük olduğundan optimum noktaya ulaşmak zaman almaktadır (Akca, 2020).

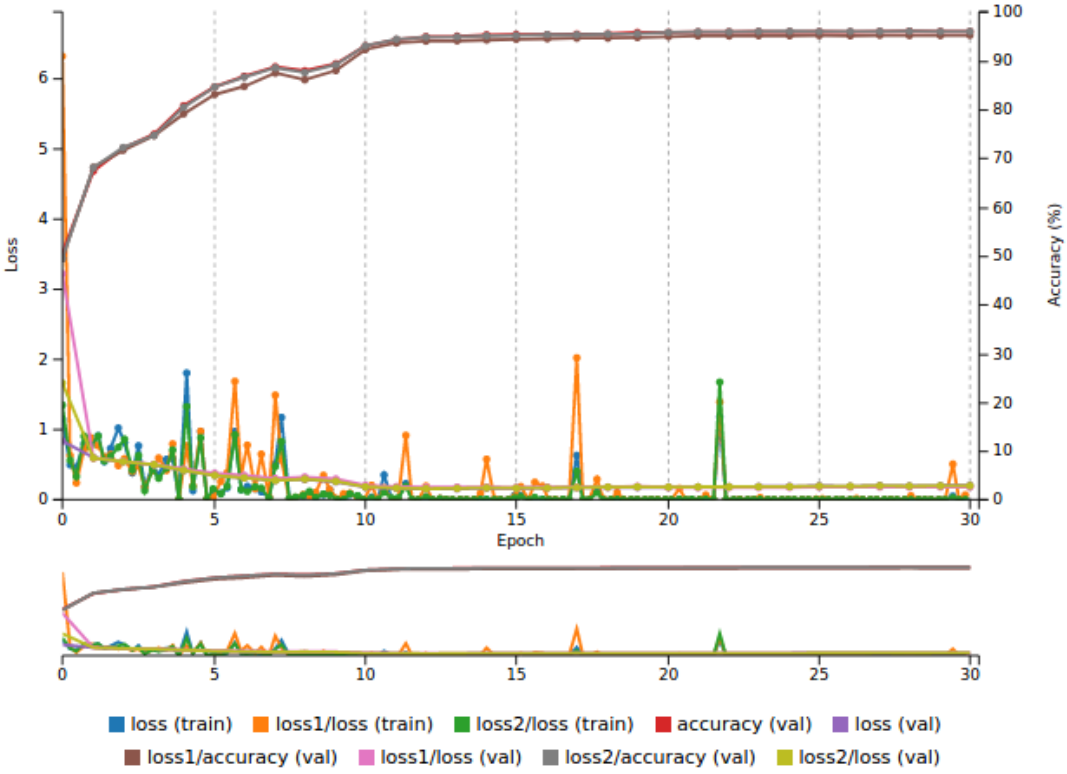
Geriye yayılım işleminde türev alınarak bulunan fark değeri, öğrenme katsayısı ile çarpılır. Buradan çıkan sonuç ağırlık değerinden çıkarılarak yeni ağırlık değerleri hesaplanmış olmaktadır. Öğrenme oranının belirlenebilmesi için, ağın büyüklüğü, problemin karmaşıklık düzeyi, ağ mimarisi gibi birçok değişken göz önünde bulundurulmaktadır.



Şekil 4.9. (a) Yüksek öğrenme oranı (b) Düşük öğrenme oranı (c) Optimum öğrenme oranı (Akca, 2020)

4.7. Tur Sayısı - Epoch

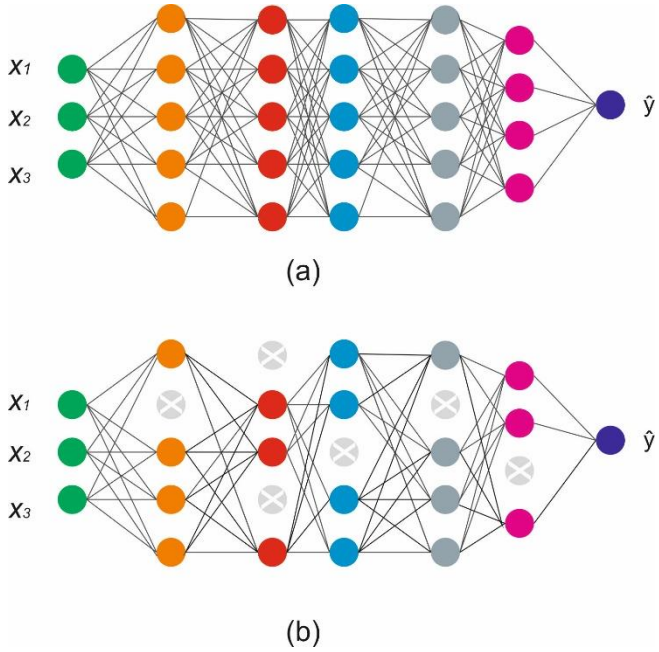
Model eğitilirken veri setinin tamamı eğitime sokulmamaktadır. Veri seti kümelere ayrılarak bir seferde belirli sayıdaki veri eğitime alınır ve model başarısı test edilir. Bu başarı durumuna göre geriye yayılım algoritması ile ağırlık yeniden hesaplanıp güncellenir. Daha sonrasında diğer veri kümesi işlenerek eğitim bu şekilde devam eder. Bu süreçte her bir döngü, tur (epoch) olarak isimlendirilir. Başarılı bir eğitim sürecinde, ilk turlarda öğrenim başarısı düşük olarak başlamakta sonraki turlarda başarı değeri artmaktadır. Bununla birlikte belirli bir tur sayısından sonra başarı değeri çok yavaş artmakta veya durmaktadır. Veri setinin boyutuna ve ağ mimarisinin yapısına bağlı olarak bazı modellerin eğitilmesi günler hatta aylar sürebilmektedir. Bu süreleri azaltabilmek için hiper-parametreler en uygun seviyelerde seçilmeye çalışılmaktadır. Şekil 4.10'da tur sayısı ve öğrenme başarısı arasındaki ilişki grafiği verilmiştir.



Şekil 4.10. Tur sayısı ve öğrenme başarısı arasındaki ilişki

4.8. Seyreltme - Dropout

Seyreltme işlemi sinir ağlarında aşırı uyum sorununu çözmek ve öğrenim başarısını arttırmak için kullanılan bir tekniktir (Maldonado-Cruz & Pyrcz, 2021). Seyreltme işlemi unutmak olarak da değerlendirilebilir. Ağdaki zayıf ağırlık değerine sahip olan nöronların ağdan atılması ile öğrenim başarısının arttığı gözlemlenmiştir (Nandini, Kumar & Chidananda, 2021). Şekil 4.11'de standart bir sinir ağının seyreltme işleminden sonraki durumu verilmiştir (Nandini vd, 2021).



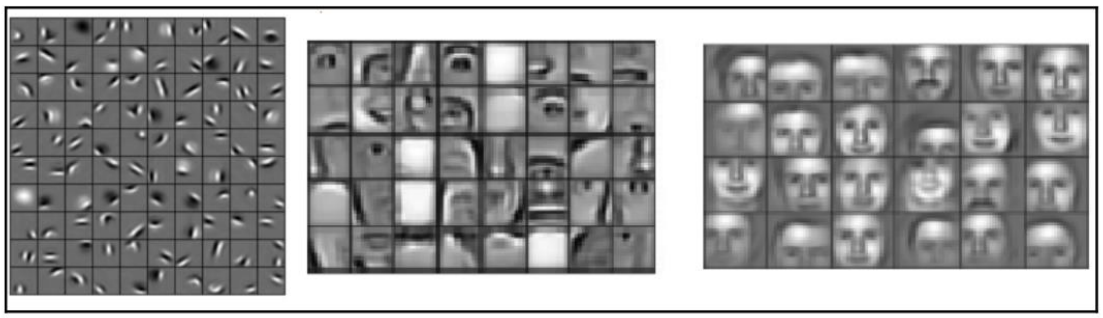
Şekil 4.11. Standart bir sinir ağı (a) Seyreltme işleminden sonra sinir ağı (b)

BÖLÜM 5

5. EVRİŞİMSEL SİNİR AĞLARI

Evrişimsel sinir ağları (ESA), insan gözünün görüntü işleme biçimini temel alarak geliştirilmiş bir YSA mimarisidir. Giriş olarak gönderilen görsellerde aranan şekil veya görüntüleri ayırt etmek, sınıflandırmak veya nesne tanımak gibi süreçler için kullanılmaktadır. Bir nesneyi tanımlayabilmek için o nesnenin düşük seviyedeki özelliklerini öğrenerek nesneyi diğer nesnelerden ayrıştırmayı öğrenir. Örneğin bir kediyi tanımlayabilmek için, kediyi kedi yapan alt özelliklerin kademe kademe çıkarımı yapılır. Klasik yapay sinir ağları mimarilerinde hesaplanan parametrelerin çok fazla olması, yoğun işlem gücü ihtiyacını doğurmaktadır. Veri setinin büyümesi, özellikle görsel tanıma gibi işlemlerde özellik çıkarımını zorunlu kılmaktadır. ESA yapısında giriş değeri olan görsel, evrişim katmanında çeşitli filtrelere tabi tutularak özellik çıkarımı gerçekleştirilir ve ağ bu özellikler üzerinde eğitilir. Bu filtreler, görselde kenar yakalamayı sağlayacak, çeşitli dikey ve yatay kenar filtreleri olabilir. LeCun ve arkadaşları (1998) yayınladıkları çalışmada ESA mimarisini kullanarak el yazısı rakamların tanınması konusunda başarı sağlamıştır. Bu çalışma ESA'da öncü kabul edilmektedir. Bundan sonra 2010 yılına kadar ESA için bir boşluk söz konusudur. Zaman içerisinde cep telefonlarında kameraların yaygınlaşması ile sosyal medya ve diğer web ortamlarında görsel materyal ve etiketlenmiş veri patlaması yaşanmış ve ihtiyaç duyulan olgunlaşma GPU gücü ile istenilen seviyeye ulaşmıştır. Lee ve arkadaşları 2011 yılında derin inanç ağları ile hiyerarşik temsillerin denetimsiz öğrenmesi üzerine yaptıkları çalışmada özellik çıkarım hiyerarşisini Şekil 5.1'deki gibi ifade etmişlerdir. Burada temel kenar parçalarından temel şekiller, temel şekillerin birleşiminden de yüzlerin oluşumu gösterilmektedir. 2012 yılında yapılan ImageNet görsel tanıma yarışmasını, ESA

mimarisi ile oluşturulan derin öğrenme modeli kazanmış ve ESA'nın etkili bir mimari olduğu kanıtlanmıştır (Krizhevsky, Sutskever & Hinton, 2012). Bundan sonraki süreçte görsel tanıma ve sınıflandırma konusunda ESA mimarisi popüler hale gelmiştir.



Şekil 5.1. Evrişim işlemi ile özellik hiyerarşisi (Lee, Grosse, Ranganath & Ng, 2011)

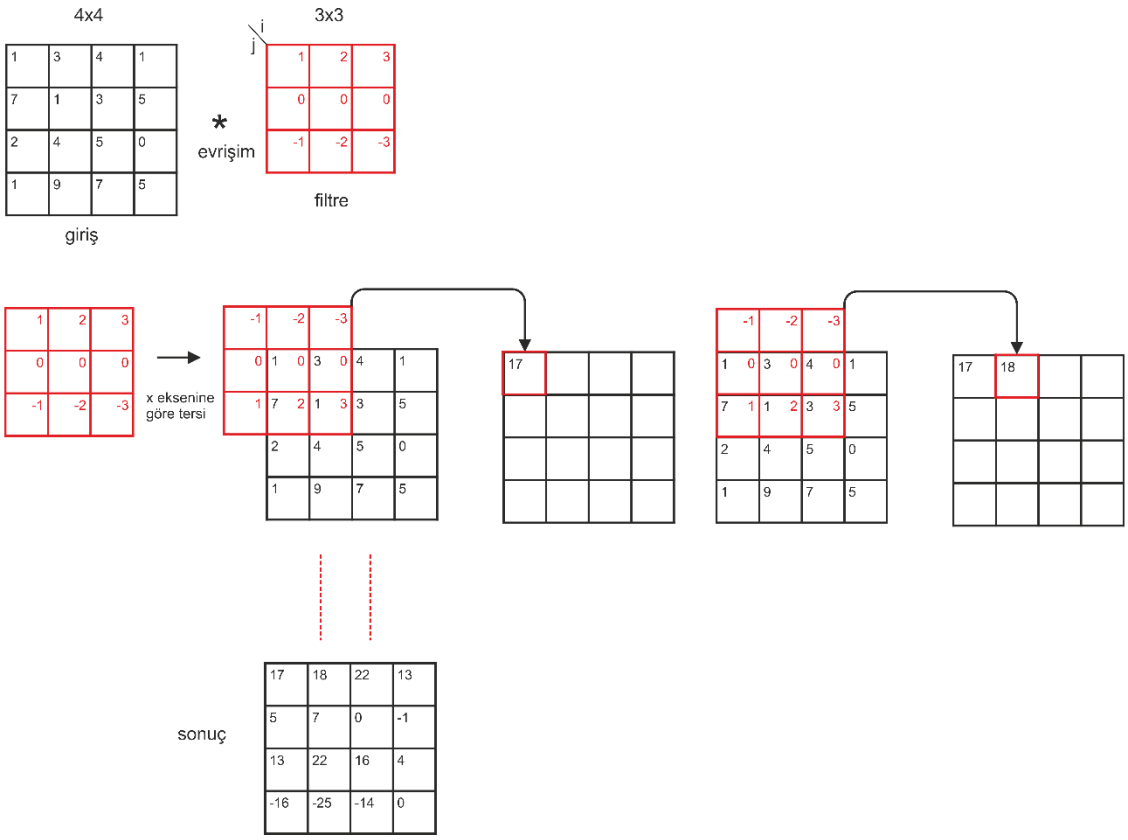
5.1. Evrişim İşlemi

Evrişim işlemi, matematikte bir h(t) sisteminin x(t) girişi için y(t) çıkış işaretini zaman düzleminde bulmaya yarayan bir işlem olarak tanımlanmaktadır (ocw.mit, 2011). Bu işlem, "*" işareti ile ifade edilir.

Giriş görselleri matris olarak ifade edilmektedir. Evrişim işleminde bu matrisler iki boyutlu olarak tanımlanmaktadır. Eğitim sürecinde GPU donanımının kullanılmasının sebebi de GPU'ların matris işlemleri üzerinde daha güçlü olmasındandır.

Evrişim işlemi Denklem (5.1)'te verilmiştir. Şekil 5.2'de iki boyutlu 4x4 bir matris üzerinde 3x3 bir filtre matrisi ile evrişim işlemi örneği verilmiştir. Evrişim işleminden önce 3x3 matrisin x eksenine göre simetriği alınmıştır. Evrişim işlemi sırasında filtre matrisinin merkezi giriş matrisinin (1,1) elemanından başlayarak işlem gerçekleştirilir. Bu işlem sırasında her giriş elemanına karşılık gelen filtre elemanı çarpılarak sonuçlar toplanır. Çıkış matrisinin (1,1) elemanından başlayarak karşılık gelen yere sonuç aktarılır. Bu işlem sırasında filtre matrisinin bazı elemanlar giriş matrisinden taşma gösterebilir. Taşan elemanların giriş matrisindeki karşılığı 0 kabul edilir.

$$y[a,b] = \sum_j \sum_i x[i,j].h[a-i,b-j] \quad (5.1)$$



Şekil 5.2. İki boyutlu matris üzerinde evrişim işlem örneği

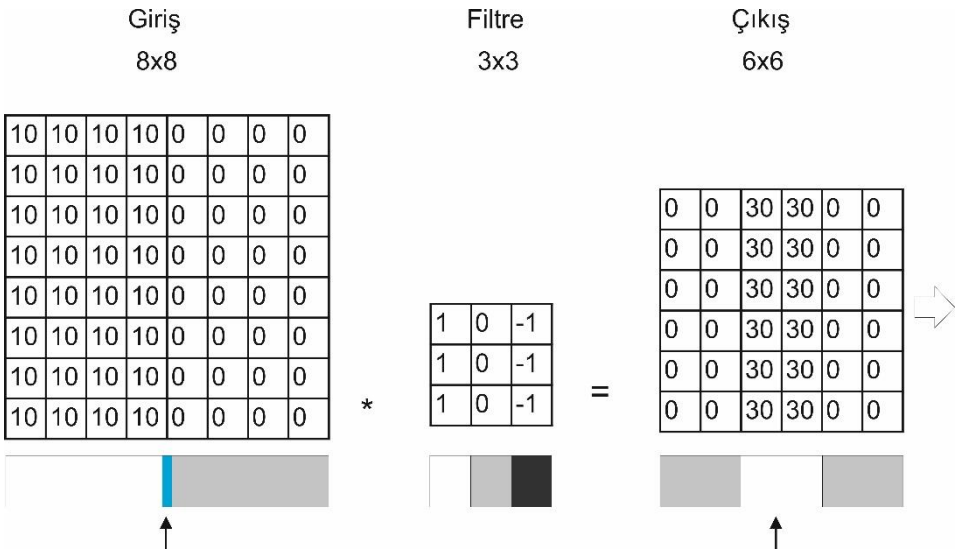
5.2. Kenar Tespiti

Kenar bulma işlemi, ESA'da özellik çıkarımı işlemi için fayda sağlamaktadır. Giriş bilgisi üzerindeki yüksek frekanslı kısımları ifade eder. Klasik makine öğrenmesinde özellik çıkarımı için Prewitt, Sobel, Robert gibi kenar bulma filtreleri uygulanmaktadır. ESA'da bu işlem giriş katmanında otomatik gerçekleştirilir.

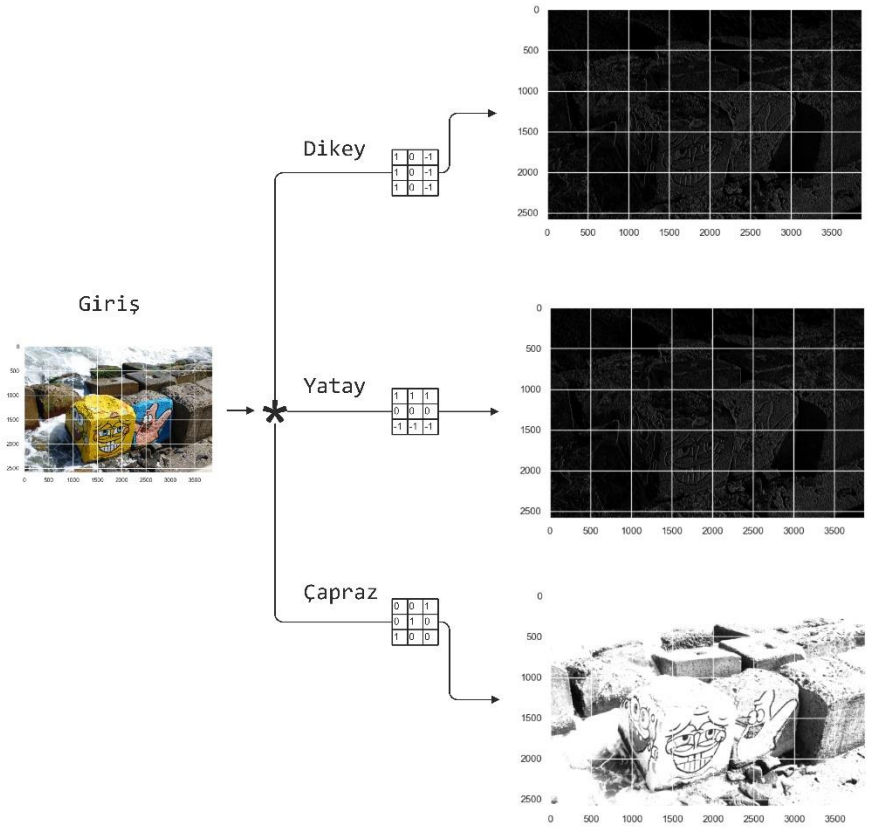
Kenar bulma işlemi, görseldeki bir nesnenin bitim çizgisi ile artalan arasındaki geçiş kısmında oluşan yüksek zıtlığın tespit edilmesi ile hesaplanmaktadır. Bu geçiş kısmında yüksek frekans tespit edilir. Şekil 5.3'teki girişteki görsel pikselleri 2 boyutlu bir matris olarak ele alınmaktadır. Matris üzerindeki yüksek değerli piksellerin açık renk, eksi değerli piksellerin de koyu renk olduğu durumda 8x8 giriş matrisine 3x3 dikey kenar

bulma filtresi uygulanmaktadır. Yakalanmak istenilen kısım, yüksek frekanslı alan olan 10'dan 0'a geçiş olan kısımdır. Bu kısım nesne ile artalan arasındaki sınır pikselleri olabilir. Sırasıyla 3x3 boyutundaki filtre matrisi, 8x8 boyutundaki giriş matrisinin üzerinde [1,1] elemanı [1,1]'e karşılık gelecek şekilde evrişim işlemi uygulanır. Bu şekilde birer kaydırılarak filtre işlemi uygulandığında çıkış matrisinin boyutu $ç_i=(çxç)$, çıkış matris boyutu, $g_i=(gxg)$, giriş matris boyutu ve $f_i=(fxf)$ filtre matrisi boyutu olmak üzere denklem (5.2)'dan 6x6 olacaktır. Şekil 5.4'te bir görsel üzerinde yatay, dikey ve çapraz kenar bulma işlem sonuçları verilmiştir.

$$ç_i = g_i - f_i + 1 \tag{5.2}$$



Şekil 5.3. Kenar bulma işlemi



Şekil 5.4. Yatay, dikey ve çapraz kenar bulma örnekleri

5.3. Padding İşlemi

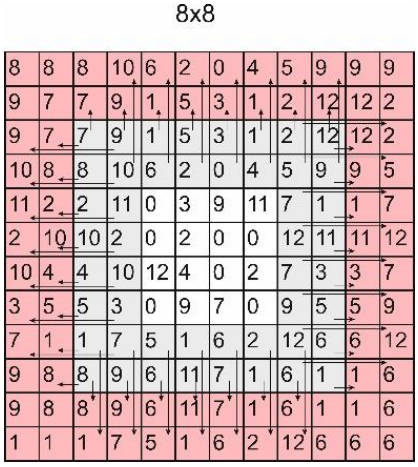
Padding işlemi, evrişim işlemi sonrasında çıkış matrisin boyutunun azalması sorununun önüne geçmek için giriş matrisinin etrafına ihtiyaç duyulan miktarda piksel eklenmesi işlemidir. Eklenecek piksel boyutu *n x n*, giriş matrisi boyutu, *f x f*, filtre boyutu, p eklenecek piksel boyutu olmak üzere $p = \frac{f-1}{2}$ şeklinde hesaplanır (Ayyüce Kızrak & Bolat, 2018).

Şekil 5.5'te 8x8 boyutunda bir matrise 3x3 boyutlu bir filtre matrisi uygulanması durumunda çıkış görüntüsünün boyutunun değişmemesi için *s* adım sayısı olmak üzere denklem (5.3) üzerinden hesaplama yapılır.

$$\left(\frac{n+2p-f}{s}+1\right) x \left(\frac{n+2p-f}{s}+1\right) \tag{5.3}$$

$$\left(\frac{8+2p-3}{1}+1\right)\ x\ \left(\frac{8+2p-3}{1}+1\right)=(8x8) \tag{5.4}$$

$p = 2$ olarak bulunur. Dolgu piksellerinin içeriği tamamen sıfır seçilebileceği gibi, en yakın piksel değerleri şekilde görüldüğü gibi kopyalanarak yerleştirilebilir. Ancak tamamının sıfır seçilmesi durumunda uygulanacak olan evrişim işleminde piksel değerlerini sıfıra düşürebileceğinden tercih edilmez.



Şekil 5.5. 8x8 boyutundaki bir matrise 3x3 boyutunda bir filtre uygulanması durumunda uygulanacak padding işlemi

5.4. Adım Aralığı

Adım aralığı evrişime tabi tutulacak olan filtre matrisinin giriş matrisi üzerinde kaç adım kaydırılacağını belirleyen parametredir. Adım boyutu arttırıldığında çıkış matrisinin boyutu küçülür. Adım sayısına göre çıkış matrisinin boyutu *s*, adım kaydırma boyutu olmak üzere, Denklem (5.5)'deki gibi hesaplanır (Ayyüce Kızrak vd., 2018).

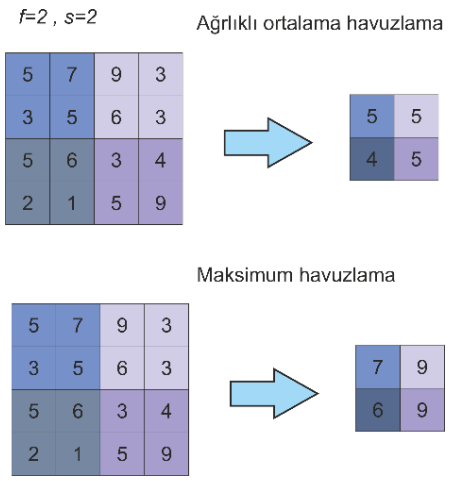
$$(h'\,x\,w') = \left(\frac{(n+2p-f)}{s}+1\right)x\left(\frac{(n+2p-f)}{s}+1\right) \tag{5.5}$$

Örneğin 7x7 boyutundaki matrisin 3x3 boyutundaki bir filtre için, Padding değeri 2, filtre adım kaydırma değeri 2 olarak belirlenirse çıkış boyutu Denklem (5.5)'den; (5x5) olarak hesaplanmaktadır.

$$(h'x\ w') = \left(\frac{(7+2.2-3)}{2} + 1\right) x \left(\frac{(7+2.2-3)}{2} + 1\right) = (5x5) \tag{5.6}$$

5.5. Havuzlama İşlemi

Havuzlama işlemi görüntü boyutunu azaltma amacı ile yapılan bir işlemdir. Filtrenin uygulandığı alandaki matris değerlerinin ağırlıklı ortalaması veya maksimum değeri alınarak çıkış matrisi oluşturulur. Şekil 5.6'da ağırlıklı ortalama havuzlama ve maksimum havuzlama örnekleri verilmiştir.



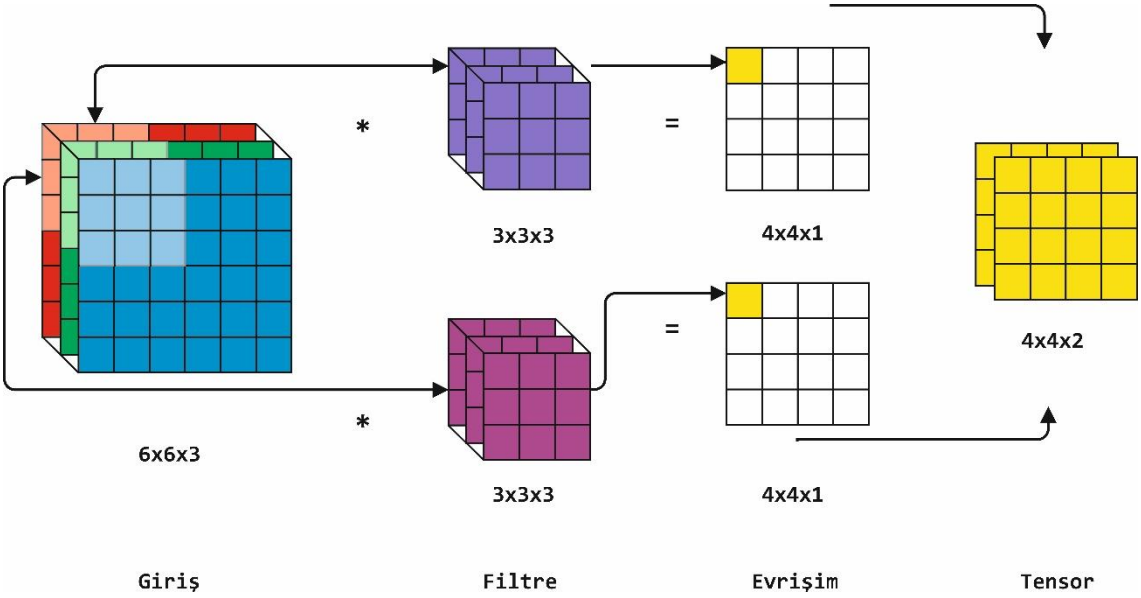
Şekil 5.6. Ortalama havuzlama ve maksimum havuzlama

5.6. Tensor Yapısı

Giriş olarak kullanılan görseller gerçek hayatta RGB renk biçimine sahip olduğundan görselin her kırmızı (R), yeşil (G) ve mavi (B) değerleri için bir matris yapısı olacaktır. Bu durumda, birbirine eşit boyuttaki üç matris için uygulanacak filtre de yine üç katmanlı bir yapıya sahip olacaktır. Filtre ile giriş matrisi üzerinde adım sayısına göre aynı sırada evrişim işlemi uygulanır. Her evrişim işlemi sonucu *n* giriş matris uzunluğu,

f filtre matrisi uzunluğu olmak üzere $n - f + 1$ boyutunda çıkış matrisine aktarılır. Burada üç katmandan elde edilen çarpım sonuçları toplanarak tek katmanda bir çıkış bir matrisinin ilgili kısmına aktarılır.

Özellik çıkarımı için evrişim işleminde çok sayıda filtre kullanılmaktadır. Ne kadar çok filtre kullanılırsa o kadar fazla özellik çıkarımı sağlanır. Giriş üzerinde her filtre ayrı ayrı evrişim işlemine tabi tutulur ve çıkış matrisleri bir sonraki aşamada biriktirilerek birleştirilir. Bu şekilde tensor yapısı elde edilmiş olur. Şekil 5.7'de çok kanallı evrişim tensor yapısı verilmiştir.



Şekil 5.7. Çok kanallı evrişim ve tensor yapısı

5.7. Kayıp Fonksiyonu

Kayıp fonksiyonu, modelin hata oranının hesaplanmasında ESA'nın son katmanında kullanılan bir fonksiyonudur. Aynı zamanda modelin başarımını ölçmek için de kullanılmaktadır (Zao, Gallo, Frosio & Kautz, 2018). Hata hesaplama işi optimizasyon problemi olarak ele alınmaktadır. Kayıp fonksiyonu ile modelin yaptığı tahminin gerçek değerden ne kadar uzakta olduğu hesaplanmaktadır. Bu değerin sıfıra yakın olması

beklenir. Kayıp fonksiyonunun iki parametresi vardır, biri ESA'nda çıkış olan tahmin değeridir, diğeri de verinin etiket değeridir.

Farklı problemler için farklı türde kayıp fonksiyonları önerilmektedir. Sınıflandırma problemleri için Softmax, regresyon problemleri için Öklid, ikili sınıflandırma problemleri için Hinge kayıp fonksiyonu yaygın olarak kullanılmaktadır. Fonksiyon denklemleri Çizelge 5.1.'de verilmiştir (Gosh, Sufian, Sultana, Chackrabarti & De, 2020). Burada "N" nöron sayısını, "p" olasılık değerini, "y" gerçek sınıf etiketini ifade eder.

Çizelge 5.1. Kayıp Fonksiyonları

Fonksiyon	***Denklem***
Softmax	$\boldsymbol{p_i = \frac{e^{a_i}}{\sum_{k=1}^{N} e^{a_k}}}$ *(N = nöron sayısı, p=olasılık)* $\boldsymbol{H(p,y) = -\sum_i y_i \log(p_i)}$
Öklid	$\boldsymbol{H(p,y) = \frac{1}{2N}\sum_{i=1}^{N}(p_i - y_i)^2}$
Hinge	$\boldsymbol{H(p,y) = \sum_{i=1}^{N} \max(0, m - (2y_i - 1)p_i)}$

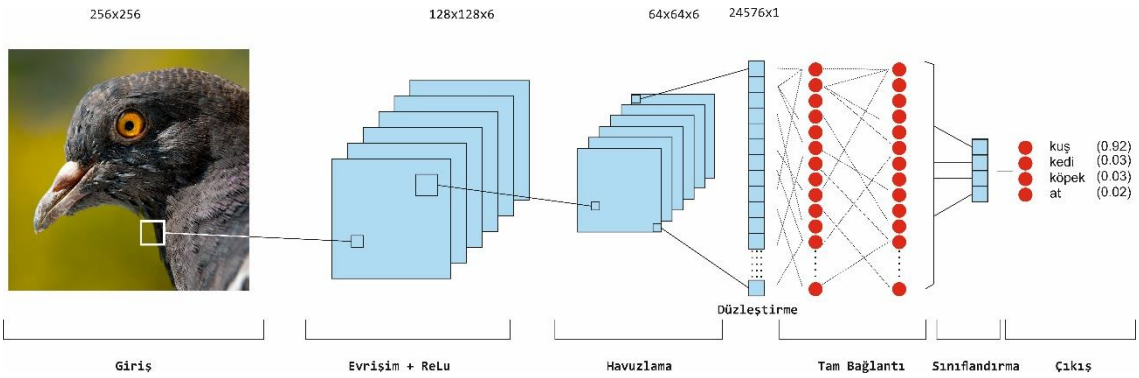
5.8. ESA Katman Yapısı

ESA'nda eğitim süreci çok katmanlı bir yapıda gerçekleşmektedir. Eğitim süresince çeşitli filtreler ile uygulanan evrişim işlemleri sonucunda elde edilen değerler aktivasyon fonksiyonlarına tabi tutulur. Sinir ağlarında gerçekleştirilen bir eğitim turu sonucunda elde edilen kayıp değeri ile geri yayılım gerçekleşir ve ağırlıklar güncellenir. Mümkün olan en iyi sonuca ulaşana kadar eğitim süreci devam eder.

ESA'nda yer alan katmanlar, giriş katmanı, evrişim katmanı (evrişim + ReLU), havuzlama katmanı, tam bağlantı katmanı, sınıflandırma katmanı ve çıkış katmanı olarak sıralanabilir. Bu katmanların sıralamaları Şekil 5.8'de verilmiştir.

Giriş katmanı, ağın ilk katmanı olup bu katmanda veri ağa ham halde verilir. Burada giriş verisinin boyutu yüksek olursa eğitim süresi ve bellek ihtiyacı artacaktır. Tam tersi, küçük olarak verilirse, detaylar kaybolacağından ağ üzerinde derinlik azalacak

ve çıkarılabilecek öznitelikler yetersiz kalacaktır. Giriş verisinin boyutu ağın derinliğini, uygulanacak filtre sayısı ise ağın genişliğini belirlemektedir.



Şekil 5.8. ESA Katman yapısı

Evrişim katmanı ESA'nın temel katmanlardan biridir. Evrişim katmanında giriş verisi üzerinden uygulanan filtreler ile öznitelikler belirlenir. Giriş verisi ile çok sayıda filtre evrişim işlemine tabi tutularak, veri üzerinde tespit edilen öznitelik miktarı arttırılır. Örnek olarak yatay kenarların tespiti için bir filtre, dikey kenarların tespiti için bir filtre çapraz kenarların tespiti için başka bir filtre kullanılabilmektedir. 3x3, 5x5, 7x7 gibi farklı boyutlarda filtreler uygulanabilmektedir. Evrişim işlemine tabi tutulan bu filtrelerin boyutları ve adım sayılarına göre çıkış verisinin boyutu azalmaktadır. Ağ derinliğini arttırmak için çok sayıda evrişim katmanı kullanılabilir. Ancak giriş verisinin boyutları küçük olduğunda her evrişim işlemi sonunda veri boyutu azaldığından, birkaç katman sonrasında veri boyutu öz nitelik çıkarılamayacak kadar küçülebilmektedir. Giriş verisi RGB olmak üzere üç farklı renk kanalına sahip olduğundan, evrişime tabi tutulacak olan filtreler de üç renk kanalından oluşmaktadır. Filtrelerin her renk kanalı, giriş verisindeki kendi renk kanalına karşılık gelecek şekilde evrişim işlemi uygulanmaktadır. Bu evrişim işlemi sonucunda her kanaldan çıkan sonuç toplanarak aktivasyon haritasına aktarılmaktadır. Her kanal için ağın eğitim durumuna göre farklı ağırlıklar kullanılabilir.

Evrişim işlemi sonucunda oluşturulan aktivasyon haritasına bir aktivasyon fonksiyonu uygulanmaktadır. Burada uygulanabilecek olan aktivasyon fonksiyonlarından Bölüm 4.5'te bahsedilmiştir. Aktivasyon fonksiyonu ile derin ağın, doğrusal olmayan bir yapıya dönüştürülmesi amaçlanmaktadır. Bu şekilde ağ öğrenimi daha hızlı gerçekleşmektedir.

Havuzlama katmanı ile, ağ üzerinde birden fazla evrişim katmanı kullanılması durumunda, bir sonraki evrişim katmanı için veri matrisinin yükseklik ve genişlik boyutlarını azaltma işlemi gerçekleştirilmektedir. Buradaki boyut azaltma işlemi bir sonraki evrişim katmanın işlem yükünü azaltmakla birlikte, ayrıca aşırı öğrenmenin de önüne geçmektedir. Veri boyutu özellik çıkarımı için bu kadar önemliyken, yine de havuzlama işlemi ile boyut düşürülerek verinin bir sonraki evrişim katmanına gönderilmesi öğrenme için iyi sonuç vermektedir. Maksimum havuzlama ve ortalama havuzlama gibi çeşitli filtre biçimlerde uygulanabilmektedir. Veri boyutunun azalmasının istenmediği durumlar için Padding işlemi uygulanarak matrisin dış kenarlarına dolgu pikselleri eklenebilir (Gosh vd, 2020).

Tam bağlantılı katmanı, evrişim işleminin sonunda üretilmiş olan tüm matris değerleri ile bağlıdır. Çıkışları ile sınıflandırma katmanını beslemektedir. Bu sebeple tam bağlantılı katman olarak isimlendirilir. Çıkış değerlerini hesaplamak için matris çarpımlarına bias değeri eklenir. Dropout tekniği aşırı öğrenme durumunun önüne geçmek için tam bağlantılı katmanda kullanılabilir. Zayıf olan nöronlar bu aşamada ağdan atılarak, güçlü değerler ile işleme gidilir (Bezdan & Bacanin, 2019). Şekil 5.8'te verilen örnek değerlerde havuzlama katmanı sonucunda *64x64x6* elde edilen çıkış 24576 adet çıkış nöronuna sahiptir. Tam bağlantılı çıkış katmanı *2048x1* boyutunda seçilirse, 24576 adet çıkış nöronun her biri 2048 tam bağlantı katmanı nöronları ile bağlanmaktadır.

Sınıflandırma katmanında, yapılacak olan sınıflandırma sayısına göre çıkış bulunmaktadır. Örnek olarak verilen Şekil 5.8'te giriş verisinin kuş, kedi, köpek ve at etiketlerine göre sınıflandırılması isteniyorsa, 4 tane çıkış değeri olması gerekmektedir. Bir önceki tam bağlantı katmanının 2048 olduğu varsayıldığında, *2048x4* ağırlık matrisi elde edilmektedir. Sınıflandırma işleminde kullanılan aktivasyon fonksiyonu genellikle softmax olarak tercih edilmektedir. Bu fonksiyon, çıkış değeri olarak her çıkış katmanına 0 ile 1 arasında bir değer döndürerek sınıflandırma işlemini gerçekleştirir. Örneğin Şekil 5.8'te verilen görselde softmax kuş için çıkış değeri 1'e en yakın değer olarak 0,92 olarak hesaplandığından giriş görselinin kuş olduğu anlaşılmaktadır.

5.9. ESA Ağ Modelleri

Evrişimli sinir ağ modelleri, temel yapı olarak ortak matematiksel bileşenleri kullanmalarına rağmen yapı olarak çeşitlilik gösterebilmektedir. Katman konumlarının

farklı biçimlerde ele alınması, yoğunluk, derinlik, genişlik ve hiperparametreler açısından gösterilen çeşitlilik problem çözümüne olumlu yönde etki etmektedir.

ImageNet görsel nesneleri tanıma için algoritmaların eğitilmesi amacıyla oluşturulmuş büyük bir görsel veri tabanıdır. 2017 yılına kadar her yıl bu veri tabanı üzerinden büyük veri tanıma zorluğu (ILSVRC – ImageNet Large Scale Visual Recognition Challenge) üzerine yarışma düzenlenmiştir (Image-net, 2021). Bu proje ile YSA'nda bir ilerleme motivasyonu oluşturmak hedeflenmiştir. 2012 yılında ilk kez AlexNet ile bir ESA mimarisinin bu yarışmayı kazanmasıyla, ESA üzerinde yapılan çalışmalar artmış ve popülerlik kazanmıştır. Çizelge 5.2'de 2010 – 2017 yılları arasında kazanan modeller ve hata oranları verilmiştir. Derin öğrenme ile geleneksel makine öğrenmesi arasındaki başarı farkı net olarak görülmektedir.

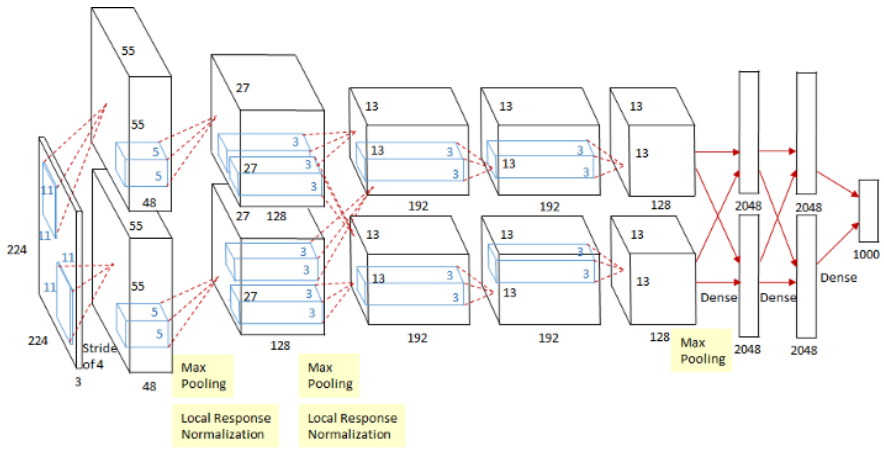
Çizelge 5.2. ILSVRC kazananları ve hata oranları (Image-net, 2021)

Yıl	***Kazanan***	***Hata Oranı (%)***	***Mimari***
2010	NEC-UIUC	28.19	Makine Öğrenmesi
2011	XRCE	25.77	
2012	AlexNet	15.30	Derin Öğrenme - ESA
2013	ZFNet	11.20	
2014	GoogLeNet	6.70	
2015	ResNet	3.57	
2016	ResNetXt	3.03	
2017	SENet	2.25	

5.9.1. AlexNet

AlexNet, ImageNet görüntü tanıma yarışmasını kazanarak ESA'na popülerlik kazandırmıştır. 2012'den önce yarışmayı kazanan klasik makine öğrenmesi tekniklerindeki %25 hata oranını %15'e kadar düşürmüştür. 60 milyon parametreyi 650.000 nöron ile paralel bağlı çift GTX 580 3GB GPU ile 6 günde işlemiştir. Paralel GPU ile işlem yapılmasını öneren ve destekleyen ilk ESA modelidir.

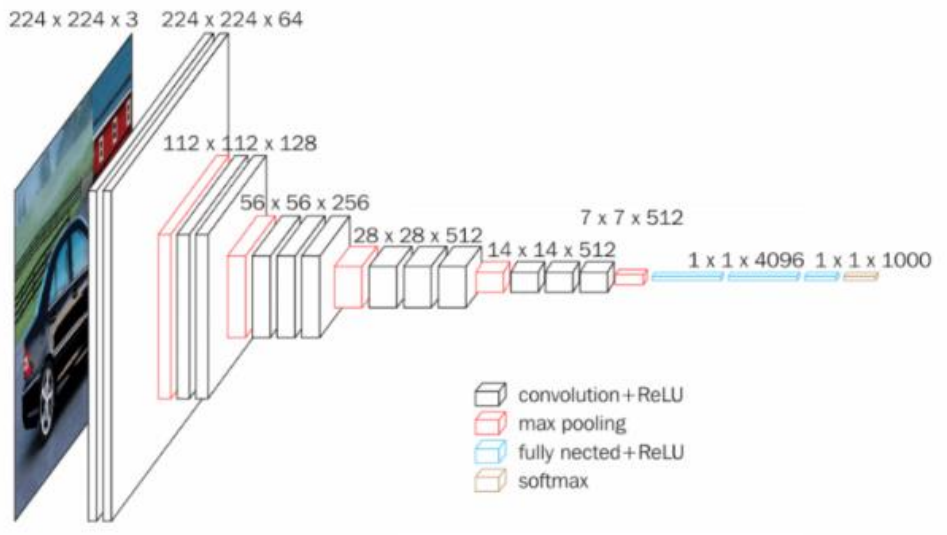
Şekil 5.9'da AlexNet mimarisi verilmiştir (Krizhevsky vd., 2012). Ağın girişinde 11x11x3 boyutunda 48+48 olmak üzere paralel olarak iki evrişim işlemi uygulanmaktadır. Devamında çıkan öznitelik haritalarına da evrişim işlemleri paralel olarak uygulanmakta ve son katmanda birleştirilmektedir. Aktivasyon fonksiyonu olarak ReLu fonksiyonu kullanılmıştır.



Şekil 5.9. AlexNet Mimarisi

5.9.2. VGG16

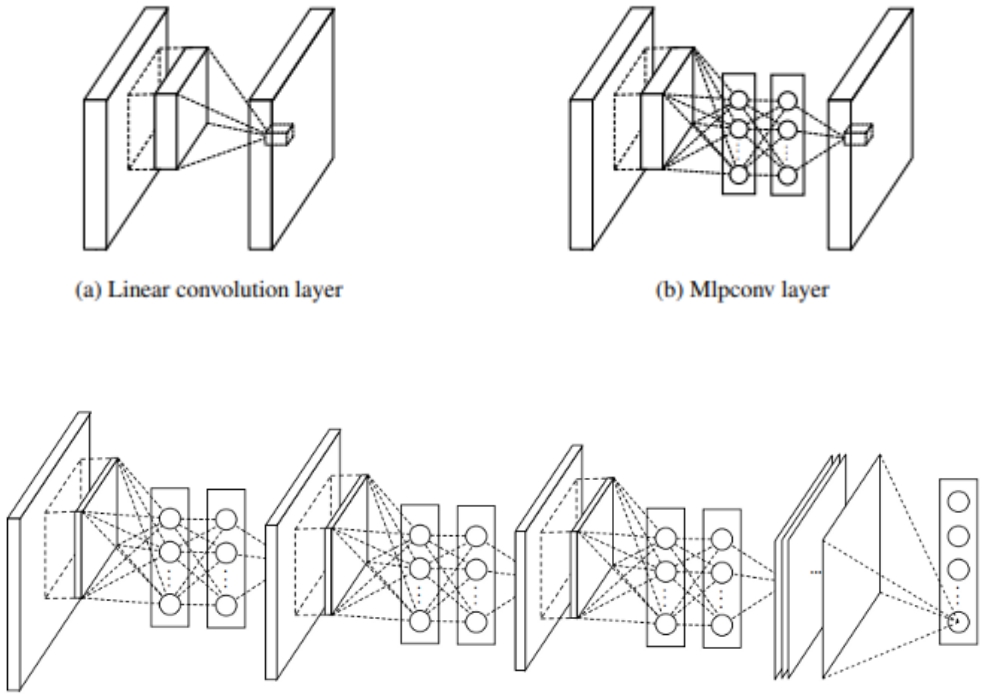
VGG16, ImageNet yarışmasında birinci olamamasına rağmen geçerli ve temel bir mimari olarak kabul edilmektedir. 2014 yılındaki ILSVRC yarışmasını %7,3'lük bir hata ile tamamlamıştır. AlexNet mimarisinde kullanılan 11x11 ve 5x5 boyutundaki filtre boyutları 3x3 boyutuna küçültüp sabitlenerek iyileştirme yapılmıştır (Simonyan & Zisserman, 2014). Evrişim ve tam bağlantılı katmanlar toplamda 16 olduğu için ismi VGG16'dır. VGG19 adında bir türevi de mevcut olup sadece katman sayıları farklıdır. Şekil 5.10'da VGG16 mimarisi verilmiştir (Hassan, 2018).



Şekil 5.10. VGG16 Mimarisi

5.9.3. GoogLeNet

GoogLeNet, 2014 yılı ILSVRC yarışmasını %6,7 top-5 error rate ile kazanan modeldir. Bu, insan hata oranına oldukça yakın bir orandır. AlexNet, VGG ve LeNet bir dizi evrişim ve havuzlama katmanından doğrusal bir yapıda yararlanarak özellik çıkarımı yapan ve son katmanda tam bağlantılı katmanda sınıflandırma yapan ortak bir mimari biçimine sahiptirler. 2014 yılında Lin ve arkadaşları "Ağ İçinde Ağ" kavramıyla derin öğrenme modelinin içerisine mikro katmanlar ekleyerek ağ içerisindeki her evrişim sürecinden sonra doğrusal olmayan bir aktivasyon fonksiyonu eklemeyi önermişlerdir. Bu şekilde doğrusal olan evrişim katmanlarının içerisine çok katmanlı algılayıcı ekleyerek öğrenme becerisini arttırmışlardır. Bu yapıya çok katmanlı algılayıcı (Multi-Layer Perception) MLP adını vermişlerdir (Lin, Chen &Yan, 2014). Yaptıkları bu çalışmada 3 adet MLP katmanı bulunmaktadır. Şekil 5.11'de doğrudan evrişim katmanı, MLP katmanı ve ağ içinde ağ yapısı verilmiştir (Lin vd, 2014).

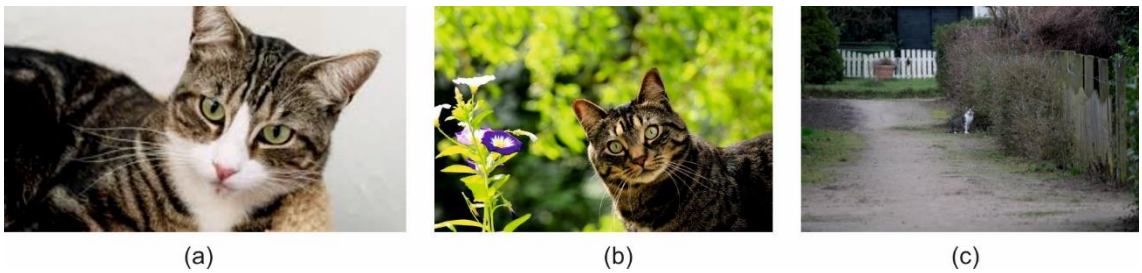


Şekil 5.11. (a) Doğrusal evrişim katmanı (b) MLP katmanı (c) Ağ içinde ağ yapısı

Google çalışanı olan Szegedy vd. 2014 yılında bu makaleden yola çıkarak LeNet mimarisini temel alan GoogLeNet mimarisini önermişlerdir (Szegedy vd., 2014). LeNet mimarisinden daha derin ve daha geniş yapıda bir model oluşturulmuştur.

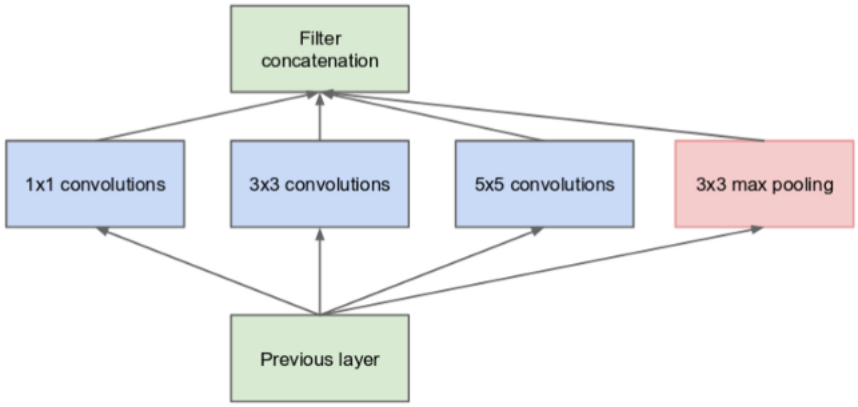
Şekil 5.12'de verilen görsellerin, sınıflandırma işlemi yapacak bir ağ için giriş verisi olduğunu düşünürsek, burada aranan özellikler için temel nesnenin görsel üzerindeki boyutu çok farklı biçimlerde olabilmektedir. Özellik çıkarımı için küçük bir filtre kullanımı, büyük görselde gereksiz ve fazla miktarda çıkarıma sebep olabilecekken, büyük filtre kullanımı görselin küçük bir kısmında yer alan aranan nesnenin özelliklerinin kaçırılmasına yol açabilmektedir. Bunun çözümü için derinliğin artması durumunda da ağ ezberlemeye yatkın bir hal alacak, ayrıca gradyan güncellemelerinin bütün bir ağ boyunca yayılımı da zorlaşacaktır. Bu durumun çözümü için GoogLeNET mimarisi içerisinde bulunan InceptionV1 ağı derinleştirmektense, aynı giriş verisine paralel olarak farklı boyutlarda filtre ile yine paralel bir havuzlama işlemi uygulayarak, genişletme

yoluna gidilmiştir (Szegedy vd, 2014). InceptionV1 daha sonrasında işlem yükünü azaltma ve performans artırımı sağlamak için InceptionV2, InceptionV3 ve InceptionV4 olarak geliştirilmiştir.

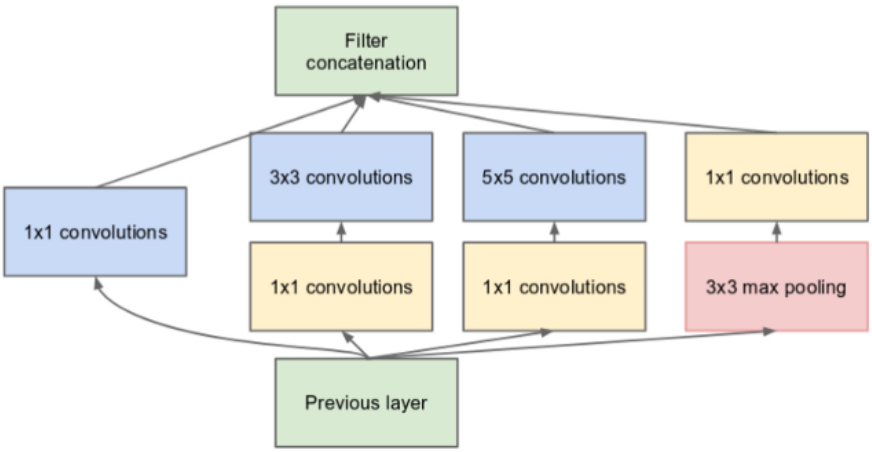


Şekil 5.12. (a) Görselin çoğunu kaplayan bir kedi (b) görselin bir kısmını kaplayan bir kedi (c) görselin küçük bir kısmını kaplayan kedi

Şekil 5.13'te naif başlangıç modülü katman yapısı verilmiştir. İşlem sayısının artmasının önüne geçmek için Şekil 5.14.'de önceki katmandan sonra 3x3 ile 5x5 evrişimlerin önüne eklenen fazladan bir 1x1 evrişim işlemi ile giriş kanalı azaltılmış olur. Ayrıca her 1x1 evrişimden sonra aktivasyon fonksiyonu olarak ReLu kullanılmaktadır. Örnek olarak giriş boyutunun 80x80x60 olduğunu varsayarsak uygulanacak olan 1x1x20 boyutundaki bir filtre ile evrişim işleminin sonucu 80x80x20 olacaktır. Bu işlem özniteliklerin havuzlaması olarak kabul edilmektedir. Bu aşama InceptionV1 olarak isimlendirilir (Szegedy vd., 2014).



Şekil 5.13. Naif başlangıç modülü (Szegedy vd., 2014)



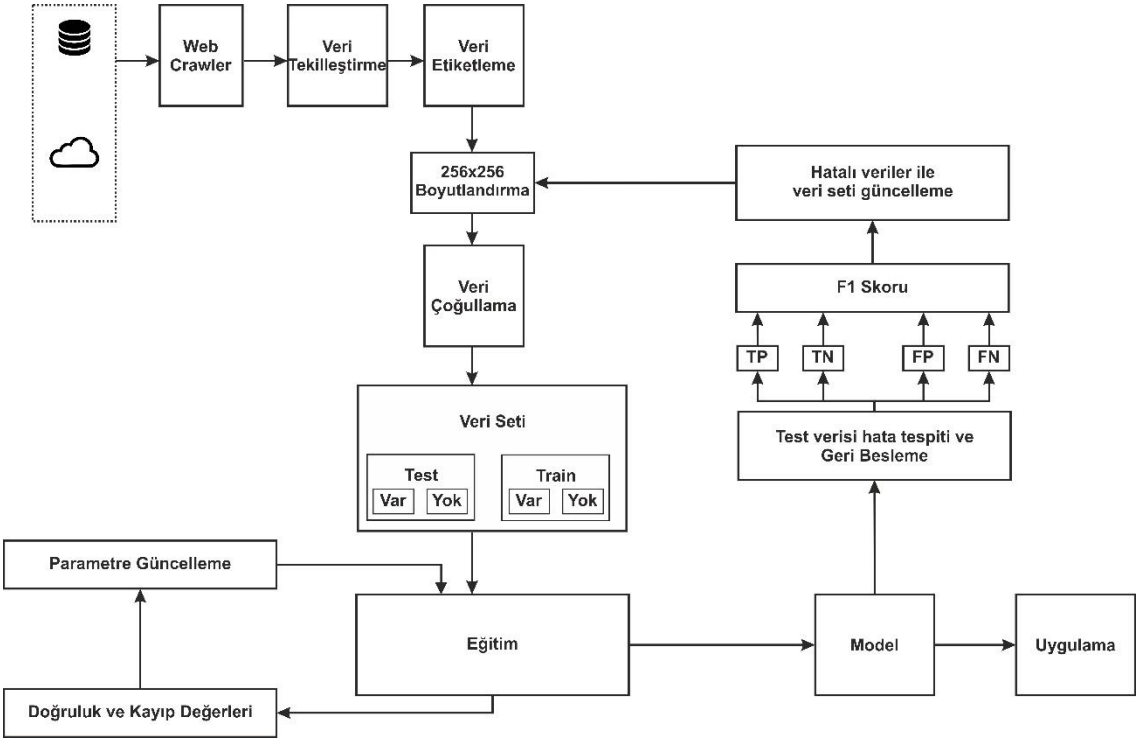
Şekil 5.14. Boyut küçültme ile başlangıç modülü

BÖLÜM 6

6. DERİN ÖĞRENME KULLANILARAK GELİŞTİRİLEN YENİ NESİL GÜVENLİK DUVARI

6.1. Materyal ve Yöntem

Yapılan bu çalışmada zararlı olabilecek görsellerin yer alabileceği bir web sitesinin adresi üzerinden engellenmesi yerine sadece görsellerin filtrelenebilmesi amaçlanmıştır. Bir web sitesinin içeriği tamamen zararlı görsellerden oluşmayabilir. Örneğin bir blog sitesi sadece birkaç gönderisinde bu zararlı görsellerden oluşan bir içerik hazırlayabilir. Geri kalan kısmı zararlı görseller içermekten ziyade faydalı olabilecek görseller de içerebilir. Sosyal medya da bu duruma gösterilebilecek diğer bir örnektir. Instagram, Facebook, Twitter gibi sosyal medya siteleri faydalı gruplar, listeler, aile ve arkadaş gönderilerinden oluşabileceği gibi, bu sitelerin kendi filtrelerinden sıyrılabilen görsellerden gönderiler de içerebilir. Haberleşme ve iletişim açısından sosyal hayatın bir parçası olan bu web sitelerini tamamen engellemek ise problemi daha farklı bir boyuta taşıyabilir. Ayrıca adres bazlı engelleme kendi içerisinde farklı zorluklar içermektedir. Engellenen web sitesi farklı bir adres ile bu engelden sıyrılabilmektedir. Bu değişikliklerin sürekli takip edilerek engel listesinin güncellenmesi ayrıca bir iş yükü sayılabilir. Şiddet, cinsellik ve madde kullanımı özendiriciliği açısından bu sitelerin filtreleri esnek olabilmektedir. Yapılan bu çalışmada, alkol tüketiminin özendiriciliği ele alınmış, oluşturulan yapıda alkollü içecekler ve alkollü içeceklerin bulunduğu ortam görsellerini tanımlayabilen bir derin öğrenme modeli eğitilmiştir.



Şekil 6.1. İşlem basamakları

Şekil 6.1'de çalışmanın işlem basamakları verilmiştir. Bu çalışma, veri seti için görsel temini, veri setinin düzenlenmesi ve etiketlenmesi, hiper-parametrelerinin seçimi, eğitim, model performans değerlendirmesi ve güvenlik duvarı uygulaması olmak üzere altı temel bölümden oluşmaktadır.

6.1.1. Uygulama Zorlukları

Filtreleme işlemi için seçilen temel zararlı içerik alkollü içecekler olarak belirlenmiştir. Ancak bunun için faydalanılabilecek hazır bir veri seti örneği bulunmamaktadır. Bunun için bir veri seti oluşturma ihtiyacı duyulmuştur. Alkollü içeceklerin çeşitleri ve marka çeşitliliği dikkate alındığında, veri setinin genişliği mevcut donanımın yeterli olabileceği seviyeleri aşmaktadır. Wikipedia web sitesinde alkollü içecekler başlığında, sadece hammaddesine göre alkollü içecekler 43 ayrı başlıkta toplanmaktadır. Yine aynı sitede fermente içkiler 64, damıtılmış içkiler 78 ayrı başlıkta ele alınmaktadır (Wikipedia, 2021). Her kategorinin farklı marka ve ambalaj yapısı içinde bulunduğu düşünülürse veri seti boyutunun oldukça yüksek olması gerekmektedir. Ayrıca bu içkilerin içerik rengi ve ambalajlarının, alkolsüz içeceklerin ambalaj ve renklerine

benzemesi durumunda, veri setinde alkolsüz içeceklere de karşılaştırma verisi olarak yüksek miktarda yer verilmesi gerekmektedir. Bütün bu çeşitliliğe ilave olarak, bazı alkollü içecekler ile gerçek hayat görselleri arasında bir örüntü benzerliği oluşması sebebiyle, gerçek hayat görsellerine de veri setinde yer verilmesine ihtiyaç duyulmuştur. Görsellerin değerlendirilmesi aşamasında sadece alkollü içerik görsellerinin ambalajlarının engellenmesinden ziyade, görselin içerisinde alkollü içecek bulunma durumu da ele alınmıştır.

İhtiyaç duyulan veri setinin büyüklüğü göz önünde bulundurulduğunda bu veri seti ile eğitilecek olan modelin eğitim süresi uzamaktadır. İdeal sonucun alınabilmesi için belirlenmesi gereken hiperparametrelerin test edilmesi de bu sürelere eklenmektedir.

6.2. Veri Seti

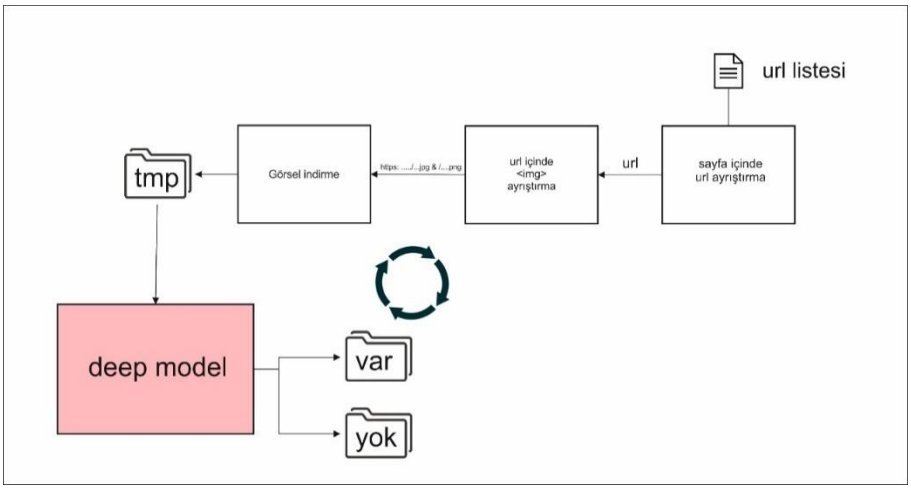
Alkollü içecekler için amaca uygun büyüklükte genel kullanıma açık bir veri seti olmadığından, model eğitiminde kullanılan görseller python ortamında tasarlanan bir web crawler aracılığı ile internet üzerinden temin edilmiştir. Tasarlanan crawler için oluşturulan arayüz görüntüleri EK-1'de verilmiştir.

Web crawler, web sayfalarını dolaşarak hyper text içerisinden etiketleri ayıklayıp indeksleyen, sayfalar içerisinden ayıkladığı bağlantı linklerini yinelenen bir yapı ile döndürerek aynı işlemleri tekrar eden yazılımlardır (Dikaiakos, Stassopoulou & Papageorgiou, 2005). Robot olarak da isimlendirilebilen bu yapı, tarama işlemini yavaş gerçekleştirdiğinden ötürü, yavaş veya emeklemek anlamına gelen "craw" kelimesi ile isimlendirilmiştir. Web sayfalarını indekslemek amacı ile kullanılmaktadır. Sürecinin yavaş ilerlemesinin sebebi, sayfalar indekslenirken sunucuya insan tepki süresinden çok daha kısa aralıklarla istek gönderilmesinin, sunucu tarafından saldırı olarak algılanabilmesidir. Dolayısıyla işlem yavaş gerçekleştirilmediğinde sunucu tarafından savunma amacıyla istekler engellenebilmektedir.

Tasarlanan crawler yapısı 7 aylık bir süre içerisinde, çeşitli web adreslerinden oluşan adres veri tabanı üzerinde kesintisiz olarak çalışmış, yaklaşık 3 milyon etiketsiz görsel elde edilmiştir.

Veri setinin ilk oluşturma aşamasında ayrıştırılan 80.000 adet görsel ile eğitilen ilk prototip, crawler yapısı ile uygulanmıştır. Bu sayede taranan web sitelerinden elde

edilen etiketsiz görseller ayrıştırılarak dosyalanmıştır. Eğitilen ilk V1 modeli veri setinin istenen genişlikte olmamasından dolayı bu işlem düşük doğrulukta gerçekleşmiş, sonrasında el ile ayrıştırılarak etiketli veri tabanı genişletilmiştir. Bu işlem etiketsiz veri tabanını düşük doğrulukta da olsa ayrıştırarak el ile yapılan ayrıştırma işlemini kısmen kolaylaştırmıştır. Şekil 6.2'de derin öğrenme modelinin uygulandığı crawler yapısının blok şeması verilmiştir.



Şekil 6.2. Web crawler blok şeması

Veri seti etiketlemesi iki ana kategoride yapılmıştır. Etiketleme, filtreleme için görselde sadece alkollü içeceklerin olduğu veya olmadığı bilgisine göre "*alkol_var*" ve "*alkol_yok*" şeklindedir. Şekil 6.3'te *alkol_var* etiketli, Şekil 6.4'te ise *alkol_yok* etiketli görsellerden oluşan veri setinden küçük bir örnek gösterilmiştir.



Şekil 6.3. alkol_var etiketli veri seti örnekleri



Şekil 6.4. alkol_yok etiketli veri seti örnekleri

Veri seti hazırlanırken, sadece alkollü içeceklerin ön planda olduğu görseller tercih edilmemiştir. Kadeh kaldırma, kalabalık alkollü bir eğlence ortamı, alkollü yemek ile birlikte olacak şekilde birçok farklı ortam ve sahne durumu *alkol_var* etiketi olarak kabul edilmiştir.

Öğrenme sürecinde konvolüsyonel ağ kullanılmıştır. Bu ağın çekirdek ve basamak büyüklükleri bir kez tanımlandığında, her bir kattaki konvolüsyon karşılık gelen girişlere göre uygun boyut çıktıları üretebildiği için girdi boyutunda hiçbir sınırlaması yoktur. Fakat temin edilen fotoğraflar 1000 x 1000 piksel ve üzerinde olduğundan disk üzerinde yer kazanımı açısından sıkıştırma işlemi uygulanarak 256 x 256 piksel boyutuna düşürülmüştür.

6.2.1. Veri Tekilleştirme

Veri setinde bir görselin birden fazla bulunması durumunda, eğitilen model bir yöne doğru hatalı öğrenme gerçekleştirebilir veya ezberlemeye başlayabilir. Bu modelin eğitiminde istenmeyen bir durumdur. Veri setinde yer alan görsellerin tekil olması önem arz etmektedir. Bu sebeple temin edilen bütün görseller Ortalama Kare Hata (Mean Squared Error – MSE) fonksiyonu ile karşılaştırılmıştır. MSE bir risk fonksiyonudur ve makine öğrenmesi modellerinin tahmin performansını ölçmek için de kullanılmaktadır. İstenen değer ile elde edilen değeri karşılaştırarak bir sonuç ortaya koymaktadır. Her zaman pozitif bir değer üretmektedir (Sarkar, 1996). Dolayısıyla görseller arasındaki benzerlik durumlarını karşılaştırmak için uygundur. Makine öğrenmesi modelinin performans ölçümünde, istenen değer ile elde edilen değer arasındaki farkın sıfıra yakın olması istenmektedir. Veri setinin oluşturulması aşamasında tekil olmayan görsellerin diğer kopyalarını bulmak için bu işlem uygulanmıştır. %5'ten daha küçük fark olan görseller veri setinden ayıklanmıştır. Şekil 6.5'te verilen örnek iki görsel arasındaki fark değeri MSE ile hesaplanmıştır.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 \quad (6.1)$$

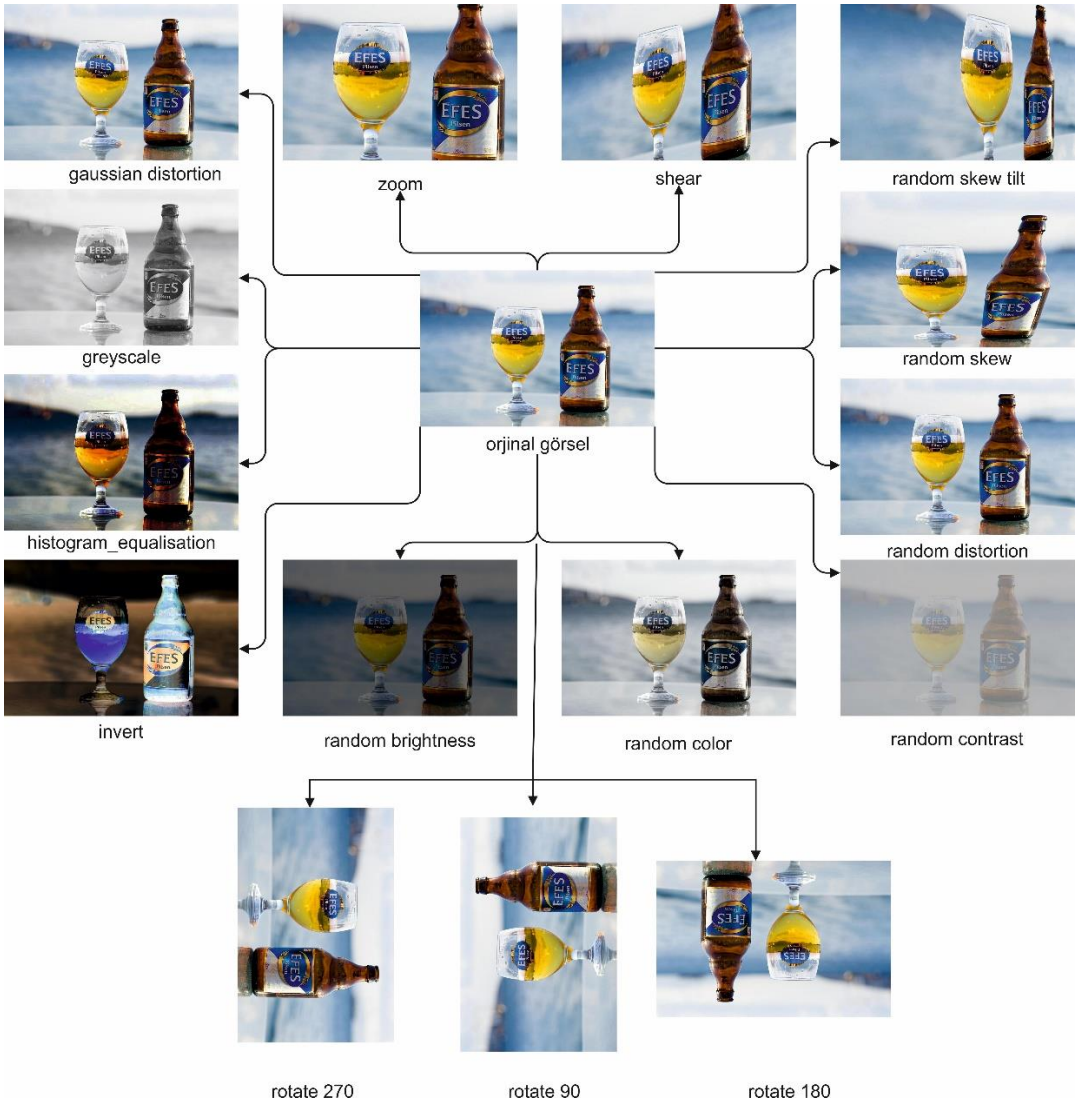


Şekil 6.5. MSE ile yapılan karşılaştırma

6.2.2. Veri Çoğaltma

Derin yapay sinir ağları, öğrenme süreci için devasa boyutlarda veri setine ihtiyaç duymaktadır. Veri setinin yetersiz büyüklükte olması öğrenme işleminde ezberlemeye (overfitting) veya öğrenmemeye (underfitting) yol açabilir. Sınırlı miktarda görsel veriyi, çeşitli algoritmalar ile çoğaltarak büyütmek mümkündür. Görsel veriyi çoğaltmak için geometrik ve fotometrik yöntemler uygulanır (Taylor & Nitschke, 2017). Geometrik çoğaltma, bir görselin ilk halinin yönünü ve pozisyonunu değiştirerek CNN öğrenme sürecinde kullanılacak farklı bir görsel oluşturmadır. Bu işlemler ters çevirme, kesme, ölçeklendirme ve döndürme olabilir. Fotometrik çoğaltma ise görsel üzerinde renk kanalları üzerinde oynamalar yapılarak uygulanan bir çoğaltma işlemidir (Shorten & Khoshgoftaar, 2019).

Veri çoğaltma işlemi için python ortamında augmentor kütüphanesinden faydalanılmıştır. Şekil 6.6'da örnek bir görselden veri çoğaltma ile elde edilen görseller verilmiştir.



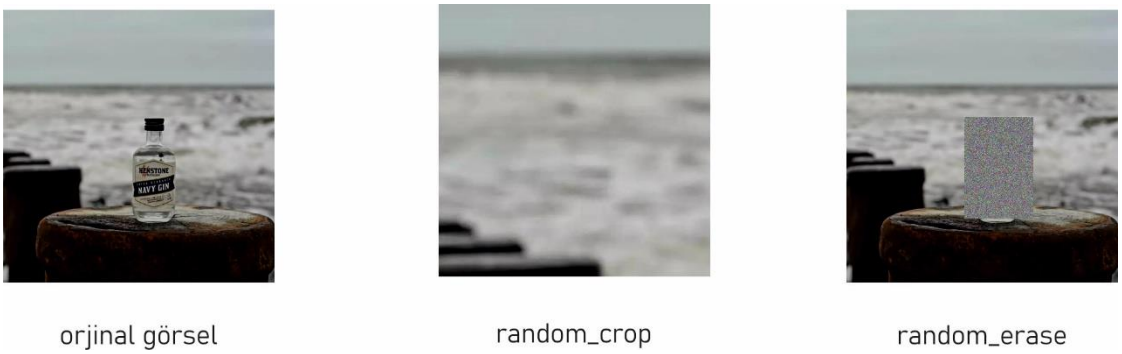
Şekil 6.6. Veri çoğaltma örnekleri

Augmentor, makine öğrenimi problemleri için veri oluşturulmasında kullanılan bir python yazılım paketidir (Augmentor, 2021). Augmentor, çok aşamalı bir işlem süreci olan veri büyütme işlemini, bir işlem sırası oluşturma yaklaşımı ile çözmektedir. Bu işlem sırası, işlemlerin sıraya eklendiği bir işlem hattı (pipeline) mantığına sahiptir. Görseller bu işlem hattından geçirilerek her işlem sırasıyla görsellere uygulanır. Burada uygulanacak her işlem için olasılık parametrelerini seçmek mümkündür. Çizelge 6.1'de uygulanan işlemler ve parametreler belirtilmiştir.

Çizelge 6.1. Augmentor kütüphanesinden uygulanan işlemler ve parametreler

İşlem	***Parametreler***
gaussian_distortion	probability=0.9, grid_width=8, grid_height=5, magnitude=5, corner="bell", method="in", mex=0.5, mey=0.5, sdx=0.05, sdy=0.05)
greyscale	probability=0.9
histogram_equalisation	probability=0.9
invert	probability=0.9
random_brightness	probability=0.9, min_factor=0.2, max_factor=0.6
random_color	probability=0.9, min_factor=0.1, max_factor=0.8
random_contrast	probability=0.9, min_factor=0.1, max_factor=0.8
random_distortion	probability=0.9, grid_width=5, grid_height=6, magnitude=4
shear	probability=0.9, max_shear_left=15, max_shear_right=15
skew	probability=0.9, magnitude=1
skew_tilt	probability=0.9, magnitude=1
zoom	probability=0.9, min_factor=1.1, max_factor=1.8

Kütüphanede, görsel üzerinde rastgele bir alanı silme (random_erase) ve bir bölümü kesme (random_crop) gibi işlemler de olmasına rağmen çoğaltma işleminde tercih edilmemiştir. Tercih edilmemesinin sebebi, veri setindeki görsellerin bir kısmında küçük bir alanda alkollü içecek görüntüsü bulunabilmesindendir. Rastgele kesme veya silme işlemi ile bu alanda bulunan görüntünün kesilmesi durumunda çoğaltılmış olan ilgili görsel üzerinde alkollü içecek bulunmayacağından belirlenen etiket ile kullanımı uygun olmayacaktır. Şekil 6.7'de bu duruma örnek olabilecek bir görselin *random_crop* ve *random_erase* durumlarında yaşanabilecek kayıpları gösterilmiştir. İlgili verilerin oranı, tek tek incelenmeden kesin bilinemeyeceğinden ötürü bu tür bir uygulamanın veri seti etiketleme doğruluğuna zarar verebileceği kararına varılmıştır.



Şekil 6.7. Rastgele kesme ve rastgele silme işlemlerinde yaşanabilecek veri kayıpları

Augmentor kütüphanesinde veri çoğaltma süreci ve üretilen veri miktarı açısından değerlendirilmesi Çizelge 6.2'de verilmiştir. Test, eğitim ve geri besleme süreçleri için veri setinde, veri çoğaltma işleminden önce, bulunan tekil veri miktarı db_01, db_02, db_03 olarak Çizelge 6.2'de gösterilmiştir. Parametre seçim süreci için db_00 veri seti, veri çoğaltma işleminin dışında tutulmuştur.

Denklem (6.2)'de, db_t toplam tekil veri sayısını; i_s, uygulanan işlem sayısını; s_v, işlem başına üretilen sentetik veri miktarını; r_n, veri başına döndürme sayısını ifade etmektedir.

$$Toplam\ Veri = (db_t + i_s . s_v)(1 + r_n) \tag{6.2}$$

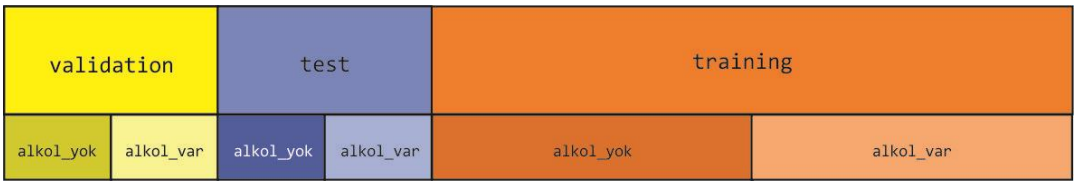
Çizelge 6.2. db_01, db_02 ve db_03 için veri setleri için veri çoğaltma işlem sayısı ve toplam veri miktarı

	db_t	i_s	S_v	r_n	*Toplam Veri*
db_01	279643	12	10000	3	1598572
db_02	305254	12	70000	3	4581016
db_03	311370	12	70000	3	4605480

Belirtilen s_v miktarı, her işlem başına tekil veri setinden rastgele işleme aldığı veri sayısını vermektedir. Her işlem sonucunda s_v kadar veri üretilmektedir. Gereğinden fazla

veri tekrarına düşülmesinden çekinildiği için bu sayı kısıtlı tutulmaya çalışılmıştır. 90, 180 ve 270 derece döndürme işlemleri, toplam işlem sonucunda elde edilen verilere ayrı ayrı uygulanmıştır.

Eğitimde evrişimsel sinir ağları kullanıldığından veri seti buna göre düzenlenmiştir. Her veri seti train, test ve validation olarak bölümlere ayrılmıştır. Her bölüm, kendi içerisinde alkol var-yok durumuna göre iki kısma ayrılmıştır. Şekil 6.8'de gösterildiği gibi, veri seti, test %20, train %60 ve validation %20 olarak düzenlenmiştir.



Şekil 6.8. Veri seti training, test ve validation bölüm oranları

Training kısmında veriler, aranan özellikteki her sınıfı temsil edecek şekilde düzenlenmiştir. Bu bölümde yer alan veriler, doğrudan sinir ağının eğitim işleminde kullanılan verilerden oluşmaktadır. Sinir ağı eğitim sürecinde her döngüde validation kısmındaki veriler ile ağırlık değerleri kıyaslanır. Elde edilen sonuca göre eğitim sürecindeki performansı iyileştirmek amacıyla parametreler ağ tarafından güncellenir. Test verileri ise ağın başarım durumunu tarafsız bir şekilde değerlendirmek için kullanılır.

Test ve validation kısımlarında yer alacak veri miktarını belirlerken örneklem çeşidi dikkate alınmıştır. Sınıflandırma işlemi için örnek sınıf çeşitleri çok fazla olduğundan, veri seti, eğitim başarımını arttırma bakımından geniş tutulmuştur. Validation ve test için belirlenecek oran, veri seti boyutunun küçük olması durumunda yeteri kadar örneklem içeremeyebileceğinden eğitim işleminde istenen başarım seviyesine ulaşılamamaktadır. Bu nedenle *db_00, db_01, db_02* ve *db_03* olmak üzere sırasıyla genişletilerek dört adet veri seti oluşturulmuş, eğitim işleminin test sürecinde genişletilerek arttırılmıştır. Çizelge 6.3'te bu veri setlerinin toplam verisi ile test, train, validation veri miktarları ve bu verilerin disk üzerindeki boyutları verilmiştir.

Çizelge 6.3. Veri setleri ve test, train, validation veri miktarları

Veri seti	*Toplam veri*	*train*	*test*	*validation*	*Boyutu*
db_00	80000	48000	16000	16000	8,65GB
db_01	1598572	959144	319714	319714	156GB
db_02	4581016	2748610	916203	916203	431GB
db_03	4605480	2763288	921096	921096	446GB

Veri Tabanı oluşturulurken lmdb yapısı kullanılmıştır. Lmdb (Lightning Memory-Mapped Database) çok basitleştirilmiş bir B-tree (binary tree) tabanlı veri tabanı yönetim kütüphanesidir. Veri tabanının tamamı bir bellek haritasında gösterilir ve tüm veri alma işlemleri doğrudan haritalanan bellekten veri döndürür, bu nedenle veri alma sırasında herhangi bir malloc (memory allocation) veya memcpy (memory copy) oluşmaz (Bi, Zhao & Jia, 2016)

Oluşturulan veri seti veri tabanlarının ortak özellikleri ise şöyledir;

- Görsel boyutu: 256x256
- Görsel Türü: Renkli
- Veri tabanı türü: lmdb
- Görsel kodlaması: PNG
- Yeniden boyutlandırma dönüşümü: Dolgu

6.3. Model Eğitim Süreci

6.3.1. Donanım

Eğitim ve uygulama için üç farklı sistem kullanılmıştır. Kullanılan sistemlerin özellikleri Çizelge 6.4'te verilmiştir. Eğitim sürecinin başında ve Hiper parametrelerin belirlenmesi aşamasında, NVIDIA GeForce 1050Ti ekran kartı bulunan birinci sistem kullanılmıştır. Daha sonraki süreçte ise veri seti boyutunun artması sebebiyle bu donanım yetersiz kaldığından Nvidia GeForce 1080Ti ekran kartı bulunan bir sisteme geçilmiştir.

Çizelge 6.4. Eğitim ve Uygulama için kullanılan donanım

Eğitim İşlemleri için		*Test ve Firewall uygulaması için*
NVIDIA GeForce 1050Ti 4GB	NVIDIA GeForce 1080Ti 11 GB	Intel Core i3 5005U
Intel Core i3 4130	Intel Core i7 8700K	Intel HD Graphics 5500
8GB DDR3 Bellek	32 Gb DDR4 Bellek	12 GB DDR3 Bellek
Ubuntu 16.04	Ubuntu 16.04	Ubuntu 20.04

6.3.2. Eğitim Ortamı

Nvidia, eğitim modellerinin sonuçlarının grafik üzerinde daha net olarak gözlemlenmesini sağlayan Digits yazılımını geliştirmiştir. Bu yazılımda, CAFFE ve Tensorflow gibi derin öğrenme araçları bir web arayüz desteği sağlanarak kullanılmaktadır. Digits ile eğitim durumu görselleştirildiğinden öğrenme durumu anlık olarak düzenli bir şekilde görsel grafiklerden takip edilebilmekte, istenmeyen sonuç gözlemlendiğinde ise eğitim istenilen seviyede kesilebilmektedir.

Ubuntu 16.04 üzerinde kararlı olarak çalışan Digits 6.1 yazılımının kurulumunu gerçekleştirmek için aşağıda verilen bağımlı kütüphanelerinden faydalanılmıştır. Eğitim ortamı için Python 2.7 kullanılmıştır.

- graphviz
- python-dev
- python-flask
- python-flaskext.wtf
- python-gevent
- python-h5py
- python-numpy
- python-pil
- python-pip
- python-scipy
- python-tk
- gfortran
- libatlas-base-dev

- libboost-filesystem-dev
- libboost-python-dev
- libboost-system-dev
- libboost-thread-dev
- libgflags-dev
- libgoogle-glog-dev
- libhdf5-serial-dev
- libleveldb-dev
- liblmdb-dev
- libopencv-dev
- libsnappy-dev
- python-all-dev
- python-matplotlib
- python-numpy
- python-opencv
- python-pydot
- python-skimage
- python-sklearn
- python-setuptools
- libprotobuf-dev
- protobuf-compiler
- doxygen
- libnccl2
- libnccl-dev

Uygulama yapılacak olan bilgisayar üzerinde Nvidia GeForce 1080ti ekran kartı bulunduğu için, ona uygun olan ekran kartı sürücüsü seçilip kurulmuştur (Nvidia, 2021).

Grafik işlemcinin paralel veri işleme kabiliyetinden faydalanmak için grafik hızlandırıcı CUDA (Compute Unified Device Architecture) 9.1 yazılımından faydalanılmıştır. CUDA, Nvidia tarafından bilgisayarın işlem performansına katkı yapması için geliştirilen bir paralel programlama platformudur. C/C++, Java ve Python

gibi dillerde yazılmış olan algoritmaların grafik işlemci üzerinde çalışmasını sağlamaktadır (Sanders & Kandrot, 2010).

Çalışmada, Cuda yazılımının derin öğrenmeye göre optimize edilmiş versiyonu olan cuDNN (CUDA Deep Neural Network Library)'ten faydalanılmıştır. cuDNN 7.5 versiyonu kullanılmıştır.

Berkeley Üniversitesi tarafından geliştirilen açık kaynak derin öğrenme uygulama platformu olan CAFFE (Convolutional Architecture for Fast Feature Embedding) eğitim platformu olarak tercih edilmiştir. CAFFE, görsel sınıflandırma ve segmentasyonu gibi farklı yapıdaki derin öğrenme mimarilerini de desteklemektedir (Jia vd., 2014). NVIDIA K40 GPU ile CAFFE günde 60 milyonun üzerinde görüntü işleyebildiğinden araştırma deneyleri için uygun bir tercih olmaktadır (Caffe, 2021).

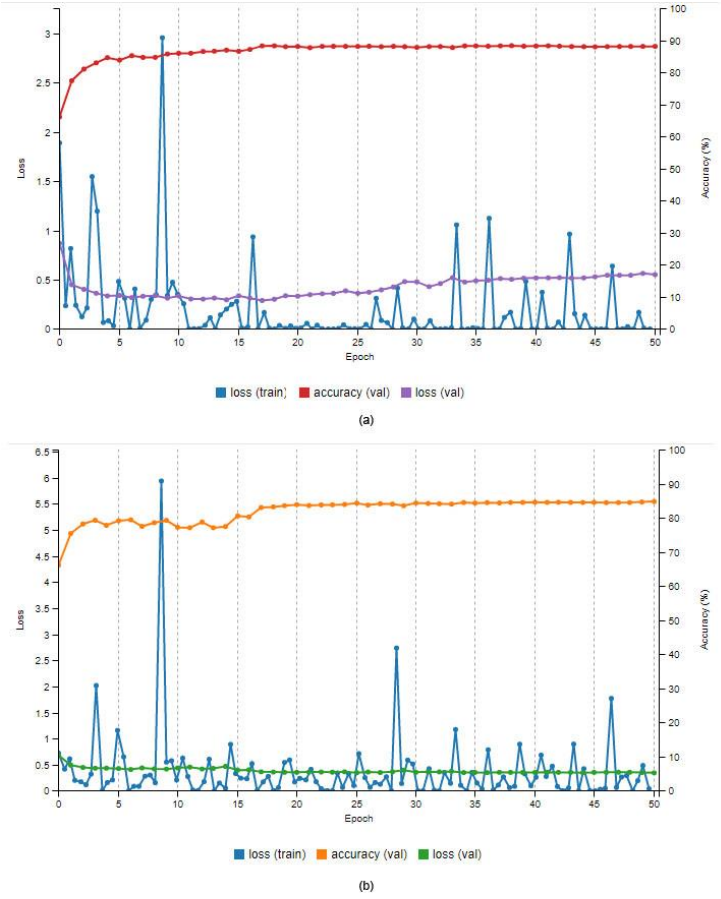
6.3.3. Hiper-Parametrelerin Belirlenmesi

Oluşturulan db_00 veri seti ile kısıtlı örneklem çeşitleri kullanılarak yapılan test eğitimlerinde uygun Hiper-parametrelerin belirlenmesi amaçlanmıştır. Büyük görsel veri ile eğitim gerçekleştirmek, günler hatta haftalar sürebilmektedir. İstenen en iyi değerler elde edilene kadar tekrar tekrar test yapılması, zaman kullanımı bakımından sorun yaratacağından, başlangıçta Hiper-parametrelerin belirlenmesi için yapılacak testlerde 80000 adet görsel veri kullanılmıştır. Ancak ele alınan problemin çözümü için çok sayıda çeşitli örnekleme ihtiyaç duyulmaktadır. Db_00, veri miktarı görsel çeşit açısından yetersiz olduğundan dolayı test eğitimlerinde ezber ve öğrenmeme durumları ortaya çıkmıştır. Test sürecinde kısıtlı veri seti ile en iyi sonucu verebilen parametreler gözlemlenmiş ve diğer geniş veri setleri ile bu parametreler kullanılarak eğitim gerçekleştirilmiştir.

Test sürecinde farklı yığın boyutları (batch size), öğrenme oranı hareket tarzları, öğrenme oranları, eniyileyici türleri ve ağ mimarileri kullanılmıştır. Bu test işlemleri sonucunda loss (val), loss (train) ve accuracy (val) değerleri dikkate alınarak değerlendirme yapılmıştır.

Öncelikle mevcut ağ mimarileri karşılaştırmaları yapılmıştır. Bunun için problemin çözümünde GoogLeNet ve AlexNet ağları test veri seti ile kıyaslanmıştır. Şekil 6.9'da loss (train), accuracy (val) ve loss (val) değerlerini gösteren grafik görüntüsü

verilmiştir. Çizelge 6.5.'te ise 20, 30, 40 ve 50. döngülerde çıktıların rakamsal değerleri gösterilmektedir.



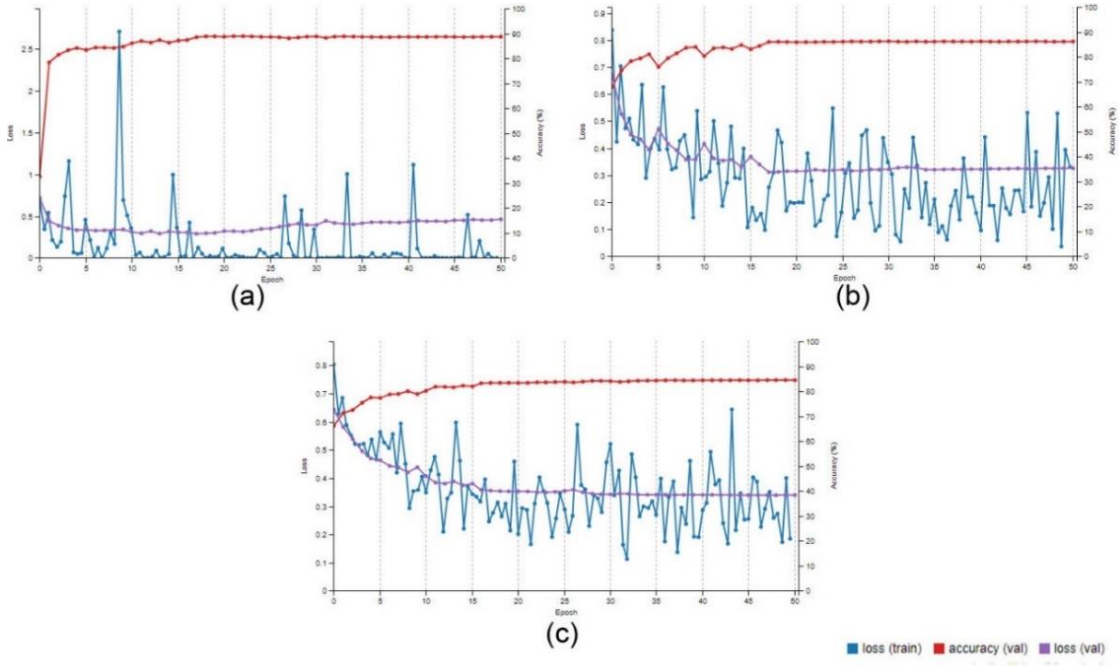
Şekil 6.9. db_00 veri seti kullanılarak GoogLeNet (a) ve AlexNet (b) karşılaştırması

Çizelge 6.5. GoogLeNet ve AlexNet mimarilerinin 20-50 arası döngülerde accuracy (val) loss (val) ve loss (train) değerlerinin karşılaştırmaları

		Accuracy (val)		***loss (val)***		***loss (train)***	
Network Türü		*GoogLeNet*	*AlexNet*	*GoogLeNet*	*AlexNet*	*GoogLeNet*	*AlexNet*
Döngü	20	88,17	83,97	0,33	0,35	0,01004	0,1697
	30	87,89	84,44	0,47	0,35	1,1 e-11	0,3545
	40	88,32	84,69	0,51	0,35	1,1 e-11	0,09748
	50	88,2	84,89	0,55	0,34	1,1 e-11	0,03857

GoogLeNet ve AlexNet 50 döngü boyunca eğitilmiş, iki mimaride de 20. döngüden sonra accuracy (val) değerlerinin küçük değişimler gösterdiği görülmüştür. Accuracy (val) değerinin başarılı bir öğrenme için yüksek olması beklenir. 20. döngüden sonra yüksek öğrenme değişimlerinin gerçekleşmediği gözlemlenmiştir. Başarılı bir öğrenme için loss (val) ve loss (train) değerlerinin zaman içerisinde azalarak birbirine yaklaşması beklenmektedir. Bu değerlerin zaman içerisinde birbirinden uzaklaşması eğitim sürecinin başarılı ilerlemediğini göstermektedir. Loss (val) değerinin loss (train) değerinin üzerinde olması aşırı uyum, iki değerin de zaman içerisinde artması öğrenmeme durumunun göstergesidir. Şekil 6.9'da GoogLeNet için zaman içerisinde aşırı uyum durumunun gerçekleştiği söylenebilir. Ancak bu, veri setinin örneklem çeşidinin az olması durumundan da kaynaklanabilmektedir. En yüksek accuracy değeri 88,2 ile GoogLeNet üzerinde gözlemlenmiştir. Ayrıca 20 döngünün eğitim için yeterli olduğu öngörülmektedir.

Yığın boyutu (batch size) 2^n şeklinde değerlerden seçilmektedir. Bu bir döngüde işlenecek veri miktarıdır. GPU bellek kapasitesinin yeterli olması durumunda yüksek değerler seçilebilmektedir. Çok yüksek seçilmesi halinde her döngüde büyük kümeler halinde veriler işleneceğinden ezber durumunun gözlemlenmesi olasıdır. Düşük seçildiğinde ise eğitim süresi uzamaktadır. Yığın boyutu belirlenirken 2, 16 ve 32 yığın boyutları ile deneme yapılmıştır. Şekil 6.10.'da eğitim sonuçlarının karşılaştırmaları verilmiştir. Yığın boyutu 2 seçildiğinde en yüksek accuracy (val) ile en uygun loss (train) ve loss (val) değerlerine 16. döngüde ulaşılmaktadır. Eğitim süresi uzadıkça model aşırı öğrenmeye eğilimli olmaya başlamaktadır. Yığın boyutunun 16 ve 32 seçildiği durumlarda ise yığın boyutunun 2 olduğu duruma kıyasla en yüksek accuracy (val) seviyesi eğitim süresi 50. döngüye ulaşmasına rağmen düşük kalmaktadır. Bu test sonucu doğrultusunda yığın boyutunun 2 seçildiği durumda öğrenmenin durumunun daha erken gerçekleşebileceği öngörülmektedir. Yığın boyutu asıl eğitim işlemlerinde ise model_1 ve model_2 eğitimlerinde 2, model_3 eğitiminde 16 olarak seçilmiştir. Çizelge 6.6'da farklı yığın boyutları ile yapılan test eğitim sonuçları verilmiştir.



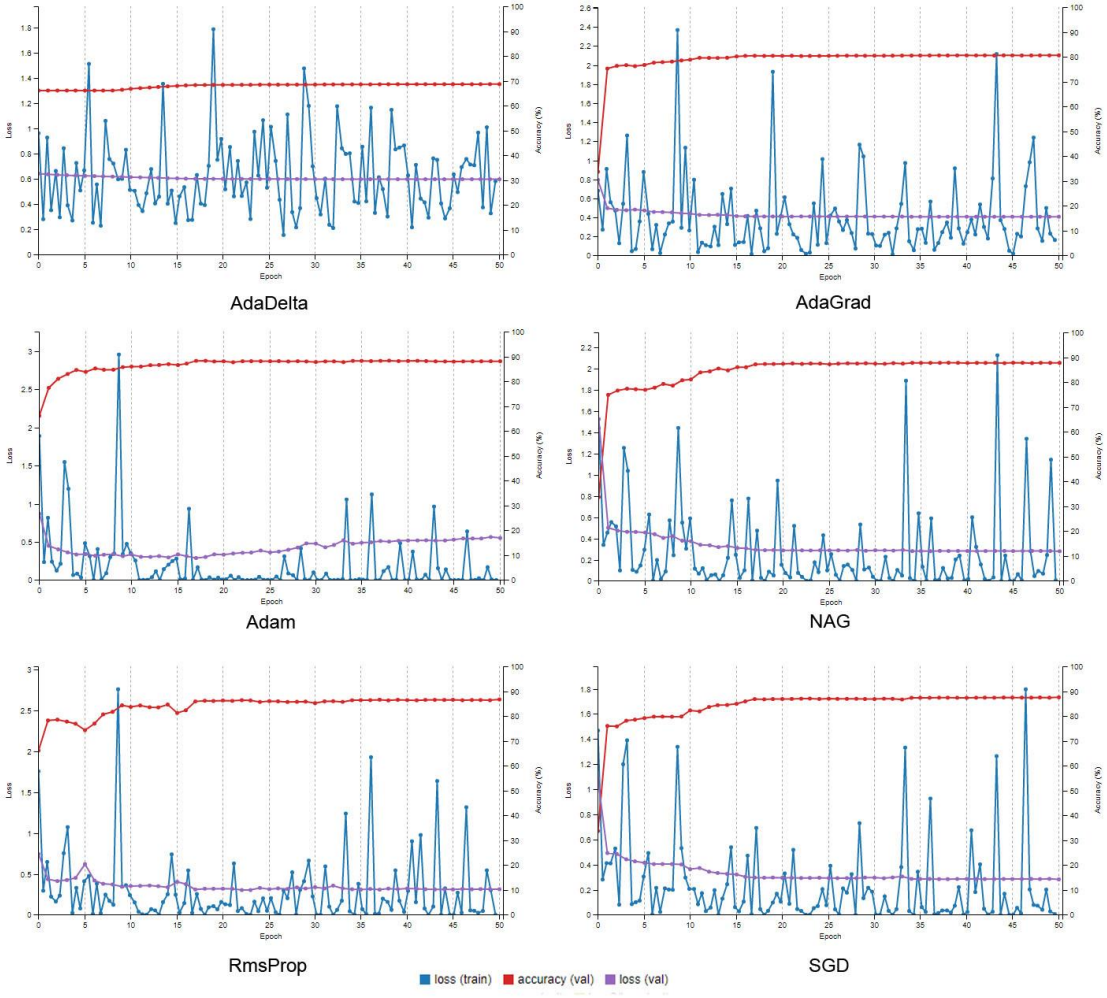
Şekil 6.10. 2 (a), 16 (b) ve 32 (c) yığın boyutlarının db_00 veri seti ve GoogLeNet kullanılarak kıyaslanması

Çizelge 6.6. 2, 16 ve 32 yığın boyutlarının 20 – 50 arası döngülerde accuracy (val) loss (val) ve loss (train) değerlerinin karşılaştırmaları

		Accuracy (val)			***loss (train)***			***loss (val)***		
Yığın boyutu		*2*	*16*	*32*	*2*	*16*	*32*	*2*	*16*	*32*
Döngü	20	88,828	86,0135	83,4637	0,005451	0,2	0,201358	0,32	0,31	0,35
	30	88,9362	86,3784	84,186	0,004192	0,3	0,520905	0,39	0,32	0,34
	40	88,7605	86,3176	84,5099	5,96 e-8	0,09	0,286525	0,43	0,32	0,34
	50	88,8348	86,2905	84,5842	0,000105	0,33	0,184655	0,46	0,32	0,339

Eniyileyici türlerinin karşılaştırmasında, en kötü sonucu AdaDelta vermiş, yaklaşık 20. döngüden sonra accuracy (val) %68,47 değerine ulaşmıştır. Sonrasında öğrenme durumu geri kalan 30 döngü boyunca sabit kalmıştır. SGD, NAG, AdaDelta ve AdaGrad eniyileyicileri loss (train) değerlerinde yüksek gürültü oranları gözlenmiştir. RmsProp en yüksek accuracy (val) değerine 20. döngüde ulaşmış, geri kalan döngülerde çok yüksek bir değişim gözlemlenmemiştir. Adam eniyileyicisi ile 50. döngü sonunda

en yüksek accuracy (val) değeri olan %88,2 gözlemlenmiş ancak 20. döngüden sonra veri setinden kaynaklı olarak ezber durumu oluşmuştur. Buna rağmen 20. döngüde diğer eniyileyicilere kıyasla en yüksek accuracy (val) değeri olan %88,17'ye ulaşılmıştır. 20. döngüde en yüksek değere ulaşılması sebebiyle eğitim için Adam eniyileyicisi tercih edilmiştir. Şekil 6.11'de eniyileyici türlerinin eğitim karşılaştırma grafikleri verilmiştir. Çizelge 6.7'de adım sönümleme ilkesi ile 20. döngüde en yüksek accuracy (val) değeri olan %88,95'e ulaşılmıştır.

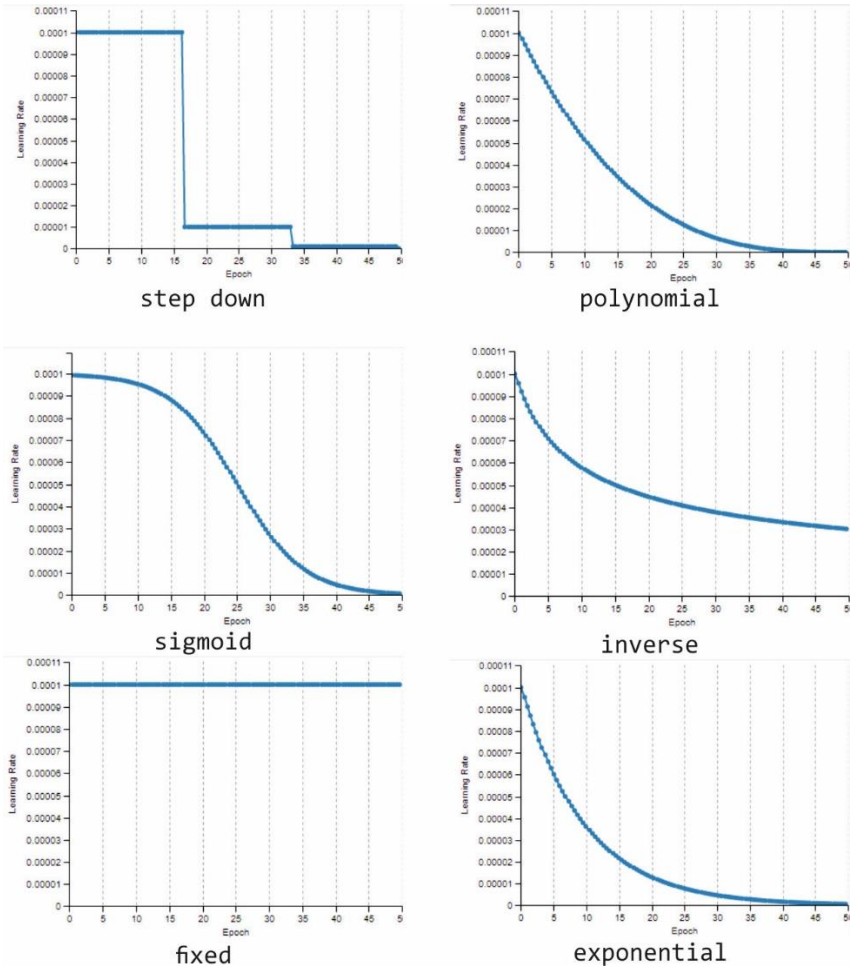


Şekil 6.11. db_00 veri seti ile AdaDelta, AdaGrad, Adam, NAG, RmsProp ve SGD Eniyileyici türlerinin eğitim karşılaştırmaları

Çizelge 6.7. Eniyileyici türleri 20 – 50 arası döngülerde accuracy (val) loss (val) ve loss (train) değerlerinin karşılaştırmaları

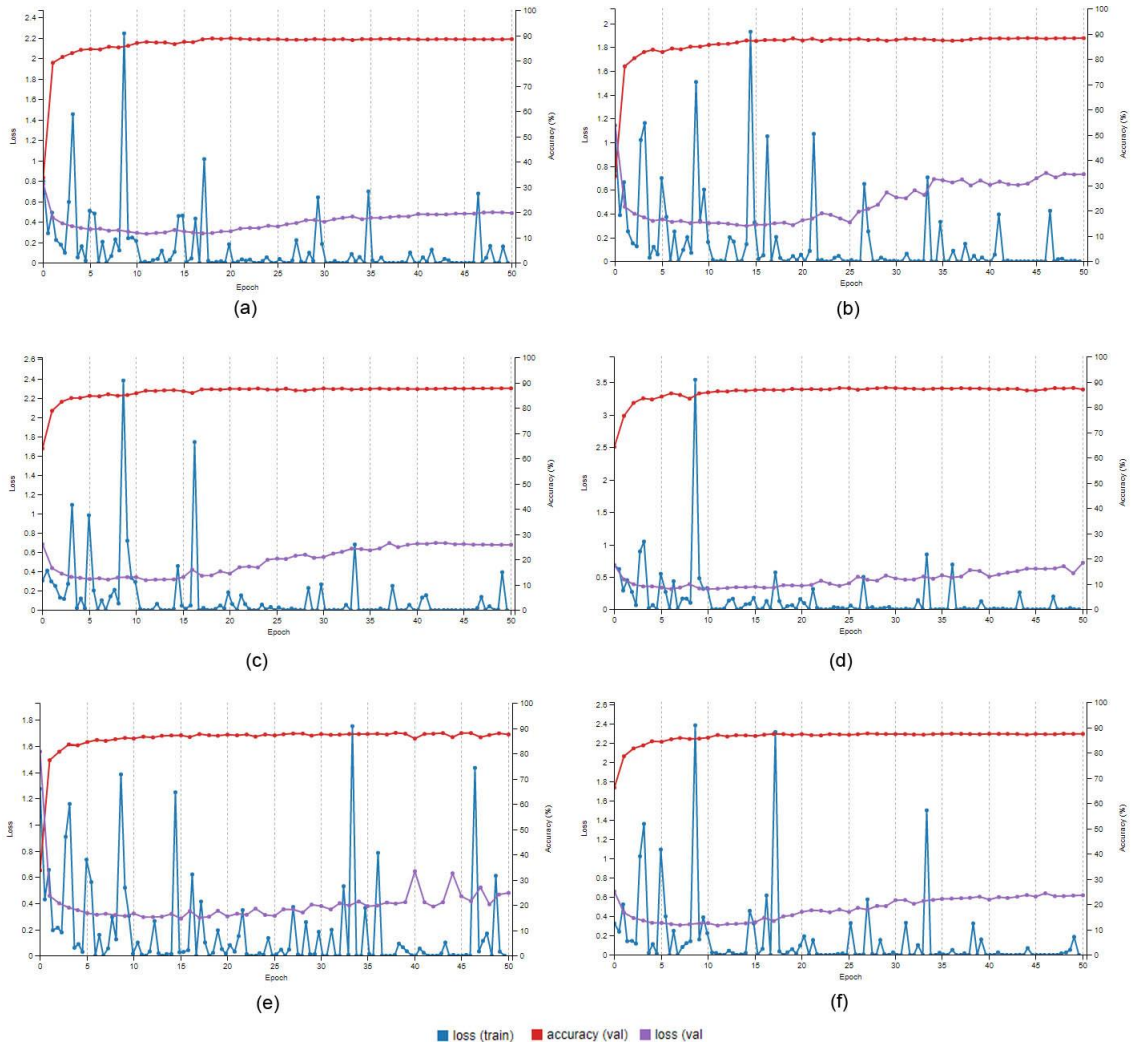
Accuracy (val)							
Eniyileyici Türü		*AdaDelta*	*AdaGrad*	*Adam*	*NAG*	*RmsProp*	*SGD*
Döngü	20	68,47	80,49	88,17	87,48	86,37	86,98
	30	68,59	80,58	87,89	87,55	85,31	86,96
	40	68,73	80,67	88,32	87,81	86,53	87,45
	50	68,77	80,71	88,2	87,86	86,79	87,66
loss (train)							
Döngü	20	0,5184	0,4134	0,01	0,15	0,1571	0,32
	30	0,4484	0,1057	1,7 e-11	0,00019	4,1 e-6	0,00031
	40	0,6279	0,1246	1,6 e-11	0,0017	0,035	0,0021
	50	0,5979	0,1635	1,1 e-11	0,0045	8,3 e-4	0,0057
loss (val)							
Döngü	20	0,6	0,41	0,33	0,28	0,31	0,29
	30	0,59	0,4	0,47	0,28	0,33	0,29
	40	0,59	0,4	0,51	0,28	0,32	0,28
	50	0,59	0,4	0,55	0,27	0,31	0,28

Evrişimli sinir ağları öğrenme oranlarına ve sönümleme şekillerine yüksek duyarlılıktadır. Öğrenme oranları ve sönümleme biçimleri üzerinde yapılacak küçük değişiklikler, tahmin doğruluğunu olumlu veya olumsuz yönde doğrudan etkilemektedir (Li, Zhang & Yoon, 2021). Öğrenme hızı ilkeleri, öğrenme oranını önceden tanımlanmış bir düzende azaltarak eğitim sırasında öğrenme oranını ayarlamayı amaçlamaktadır. Şekil 6.12.'de Digits yazılımında mevcut olan sabit (fixed), adım sönümleme (step down decay), polinom sönümleme (polynomial decay), devrik sönümleme (inverse decay), üstel sönümleme (exponential decay) ilkeleri gösterilmiştir.



Şekil 6.12. Test edilen öğrenme oranı sönümleme ilkeleri

Şekil 6.13'te farklı sönümleme ilkeleri kullanılarak eğitilen modellerin eğitim grafikleri verilmiştir. Çizelge 6.8'de ise bu eğitimlerin accuracy (val), loss (val) ve loss (train) çıktılarının rakamsal değerleri verilmiştir.

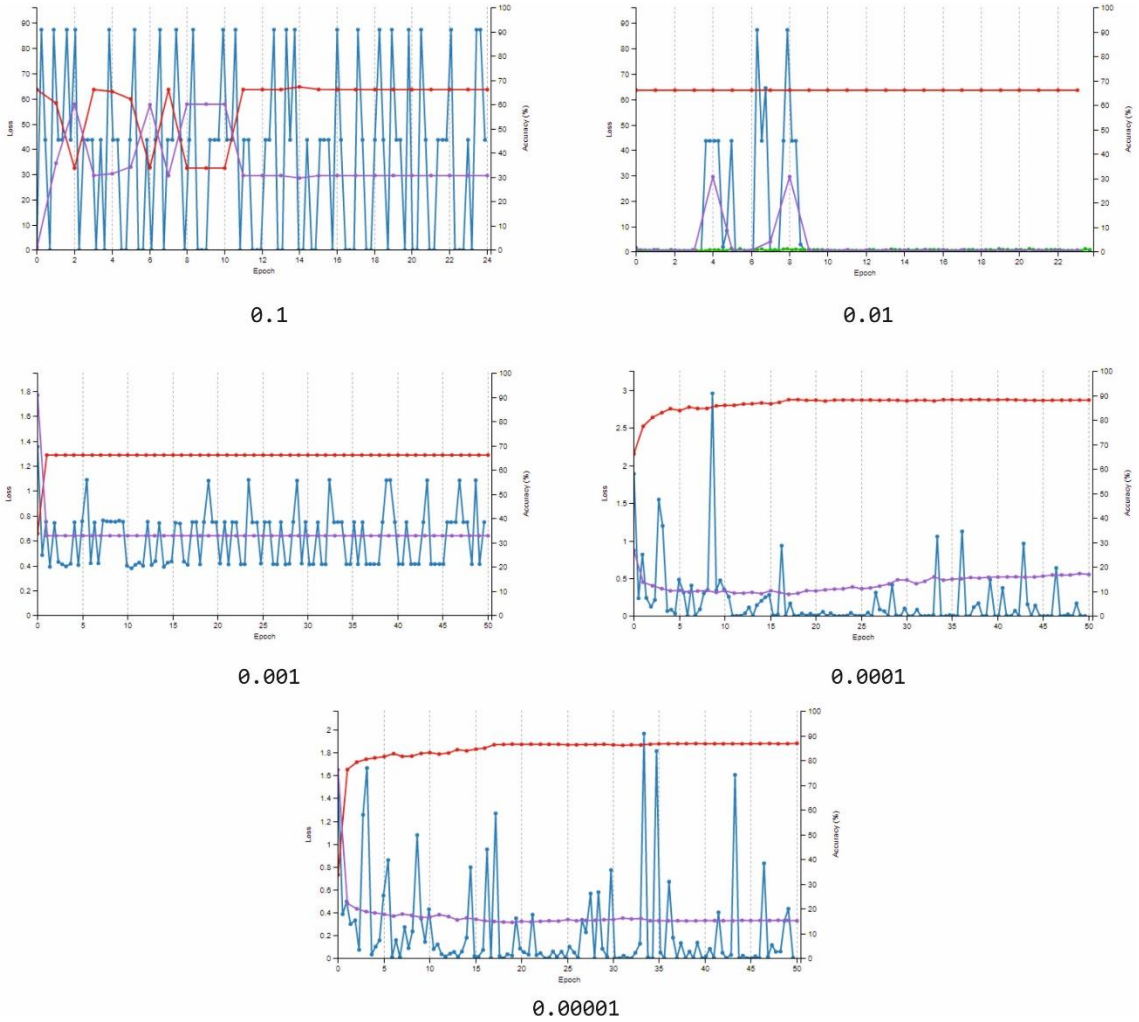


Şekil 6.13. Azalan Sönümleme (a) Sigmoid Sönümleme (b) Polinom Sönümleme (c) Devrik Sönümleme (d) Sabit (e) Üstel Sönümleme (f) temel öğrenme oranı ilkelerinin db_00 ile eğitim karşılaştırmaları

Çizelge 6.8. Öğrenme oranı sönümleme ilkelerinin 20 – 50 arası döngülerde accuracy (val), loss (val) ve loss (train) değerlerinin karşılaştırmaları

Accuracy (val)							
Öğrenme Oranı Sönümleme İlkeleri		*Adım azalan*	*Sigmoid*	*Polinom*	*Devrik*	*Sabit*	*Üstel*
Döngü	20	88,95	87,44	87,65	87,09	87,53	87,42
	30	88,51	87,7	87,8	87,59	87,78	87,44
	40	88,51	88,24	87,53	87,26	85,98	87,57
	50	88,63	88,38	87,83	87,05	87,61	87,53
loss (train)							
Döngü	20	0,0013	0,001592	0,184423	0,09	0,016	0,0976
	30	1,7 e-11	1,7 e-11	1,7 e-11	1,7 e-11	5,9 e-8	0,0277
	40	1,6 e-11	5,9 e-8	1,9 e-6	8,06 e-4	0,0035	5,9 e-8
	50	1,19 e-7	1,19 e-11	9,5 e-6	2,38 e-7	0,00026	1,6 e-6
loss (val)							
Döngü	20	0,31	0,34	0,37	0,36	0,3	0,44
	30	0,42	0,53	0,55	0,47	0,37	0,56
	40	0,47	0,64	0,68	0,5	0,64	0,57
	50	0,48	0,73	0,67	0,71	0,47	0,61

Şekil 6.14'te Bölüm 4.16'da bahsedilen öğrenme oranı katsayıları test edilmiştir. Test edilen öğrenme oranı katsayıları, 0.1, 0.01, 0.001, 0.0001 ve 0.00001 değerleridir. Test, adım sönümleme ilkesi kullanılarak yapılmıştır. Öğrenme katsayısı büyük seçildiğinde öğrenme işlemi gerçekleşmemektedir. Öğrenme işlemi gerçekleşmediği için erken durdurmaya gidilmiştir. 0.0001 ile 0.00001 katsayılarında öğrenme gerçekleşmiştir. En iyi accuracy (val) değeri 0.0001 katsayısı ile 20. döngüde gözlemlenmiştir. Çizelge 6.9'da eğitimlerin accuracy (val), loss (val) ve loss (train) çıktılarının rakamsal değerleri verilmiştir.



Şekil 6.14. Öğrenme katsayılarının karşılaştırılması

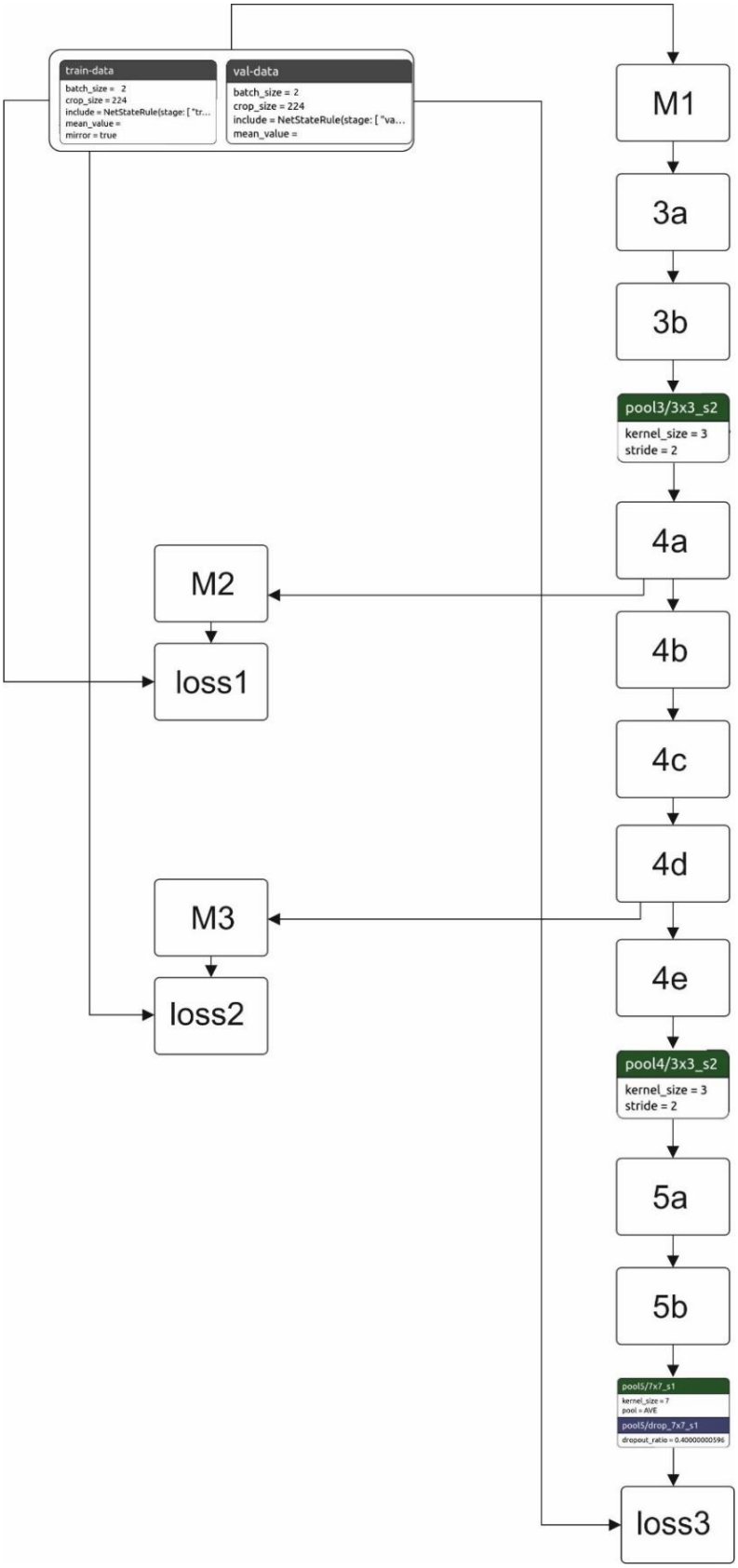
Çizelge 6.9. 0,1 – 0,01 – 0,001 – 0,0001 – 0,00001 öğrenme katsayı değerlerinin 20 – 50 arası döngülerde accuracy (val), loss (val) ve loss (train) değerlerinin karşılaştırmaları

Accuracy (val)						
Öğrenme Katsayısı		*0,1*	*0,01*	*0,001*	*0,0001*	*0,00001*
Döngü	20	66,2	66,2	66,2	88,95	86,56
	30	-	-	-	88,51	86,37
	40	-	-	-	88,51	86,83
	50	-	-	-	88,63	86,97
loss (val)						
Döngü	20	29,5	0,63	0,63	0,31	0,32
	30	-	-	-	0,42	0,33
	40	-	-	-	0,47	0,33
	50	-	-	-	0,48	0,32
loss (train)						
Döngü	20	0	0,74	0,41	0,0013	0,085
	30	-	-	-	1,7 e-11	0,000003933
	40	-	-	-	1,6 e-11	0,01965
	50	-	-	-	1,19 e-7	0,004927

6.3.4. Eğitim

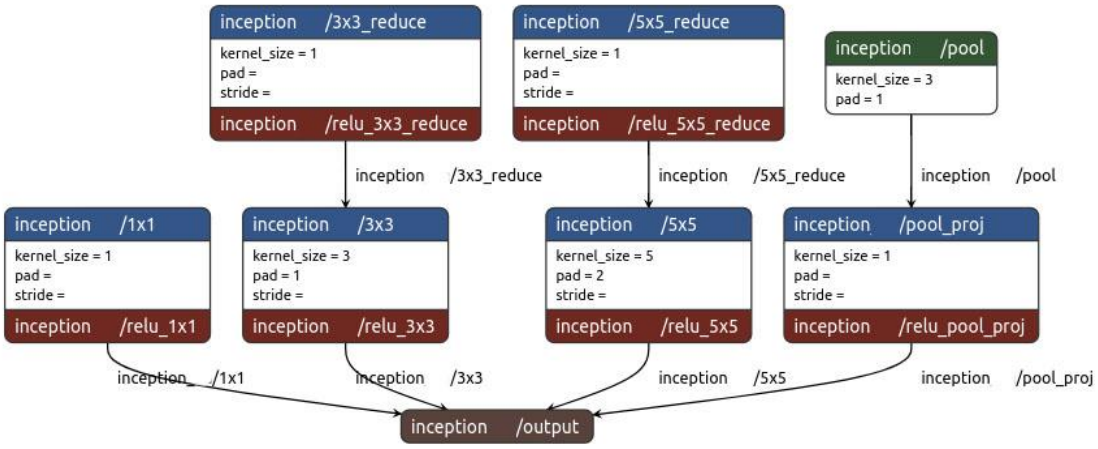
Hiper parametreler belirlendikten sonra eğitim işlemi üç aşamada ele alınmıştır. Veri seti *db_01* ile model_1, 30 döngü ile NVIDIA GeForce 1050Ti ekran kartı kullanılarak 8 gün 21 saatte eğitilmiştir. Model uygulama test sonucuna göre hatalı tahmin olduğu tespit edilen veriler ile veri seti genişletilerek *db_02* oluşturulmuştur. Model_1 eğitim ağırlıkları başlangıç değeri alınarak, 20 döngü ile NVIDIA GeForce 1050Ti ekran kartı kullanılarak model_2, 21 gün 1 saatte eğitilmiştir. Model_2 test sonucuna göre hatalı tahmin verileri ile *db_03* oluşturularak model_3 NVIDIA GeForce 1080Ti ile 30 döngüde 2 gün 17 saatte eğitilmiştir. Güvenlik duvarı uygulamasında model_3 kullanılmıştır.

Eğitimde ESA mimarisini temel alan GoogLeNet inception ağ modeli tercih edilmiştir. Uygulamada kullanılan GoogLeNet mimarisi blok şeması Şekil 6.15'te verilmiştir.



Şekil 6.15. GoogLeNet Inception Ağ Mimarisi

Şekil 6.15.'te verilen GoogLeNet mimarisinde her 3a, 3b, 4a, 4b, 4c, 4d, 4e, 5a ve 5b modülleri inception olarak tanımlanan katmanlardır. Kullanılan inception katmanları ortak blok şeması Şekil 6.16'da verilmiştir. 1x1, 3x3, 5x5 çekirdek evrişim katmanları ile havuzlama katmanı ReLu işlemi ile bir sonraki modülün girişine yönlendirilmektedir.



Şekil 6.16. GoogLeNet Inception modülü

3b inception modülünden sonra bir adet, 4e inception modülünden sonra da bir adet 3x3 havuzlama katmanı yer almaktadır. 5b inception modülü ardından bir adet 7x7 havuzlama ve dropout katmanı bulunmaktadır. Dropout değeri 0,40000000596 olarak belirlenmiştir. 4a ve 4d ardından M2 ve M3 modülü ile loss1 ve loss2 ara çıkış katmanları yer almaktadır. M1, M2 ve M3 modülleri ile loss1, loss2, loss3 çıkış modülleri blok şemaları sırasıyla *EK-2*, *EK-3*, *EK-4*, *EK-5*'te verilmiştir.

Model sonuç çıktısı olarak Softmax çıkış değerleri kullanılmıştır. Softmax çıkış değerleri 0 ile 1,0 arasında bir olasılık değeri döndürür. Sınıflandırma sayısı kadar çıkış olması gerekmektedir. Ele alınan eğitim probleminde "var" veya "yok" olarak iki sınıf çıktısı yer almaktadır. Bu sınıf çıktılarının toplamı 1'e eşittir. En yüksek çıkış değerine göre tahmin sonucu elde edilir.

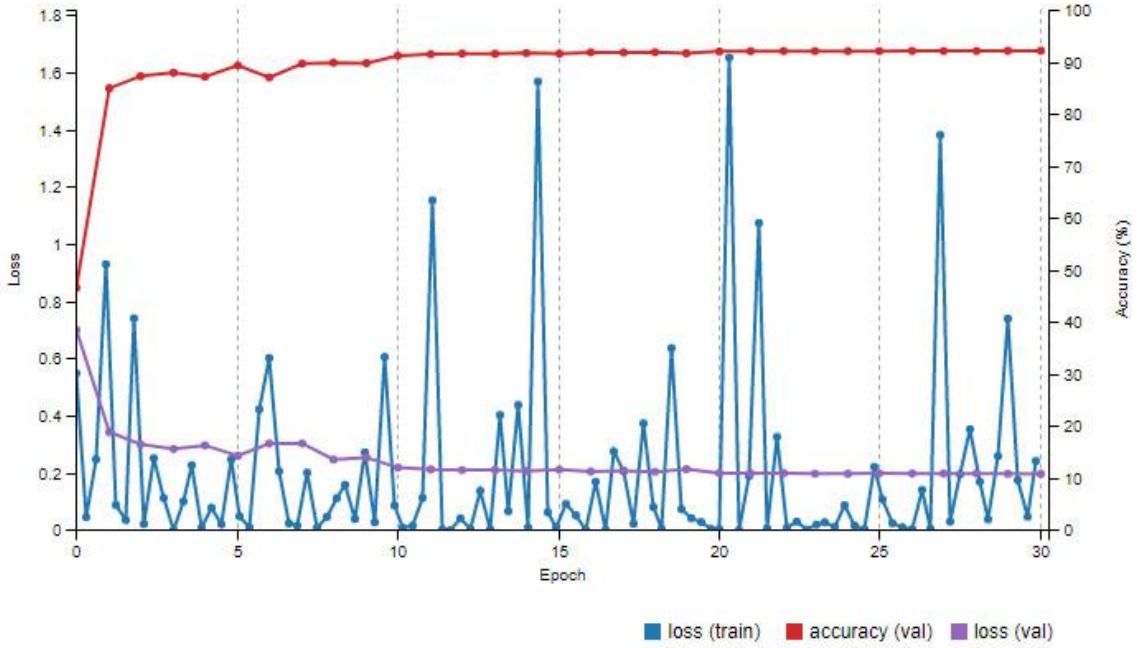
Eğitim sürecindeki Model_1, Model_2 ve Model_3'ün Hiper-paramtereleri Çizelge 6.10.'da verilmiştir. Veri seti boyutu, yığın boyutu ve döngü sayısına göre iterasyon sayıları verilmiştir. Test kümesi üzerinde çalıştırılan iterasyon sayısı "*test_iter*"

doğrulama veri kümesi ile yığın boyutu bölümü ile elde edilir. Test seti üzerindeki çalıştırma aralıkları "*test_interval*" başlığı ile verilmiştir. Buna göre maksimum iterasyon sayıları "*max_iter*" başlığı ile verilmiştir.

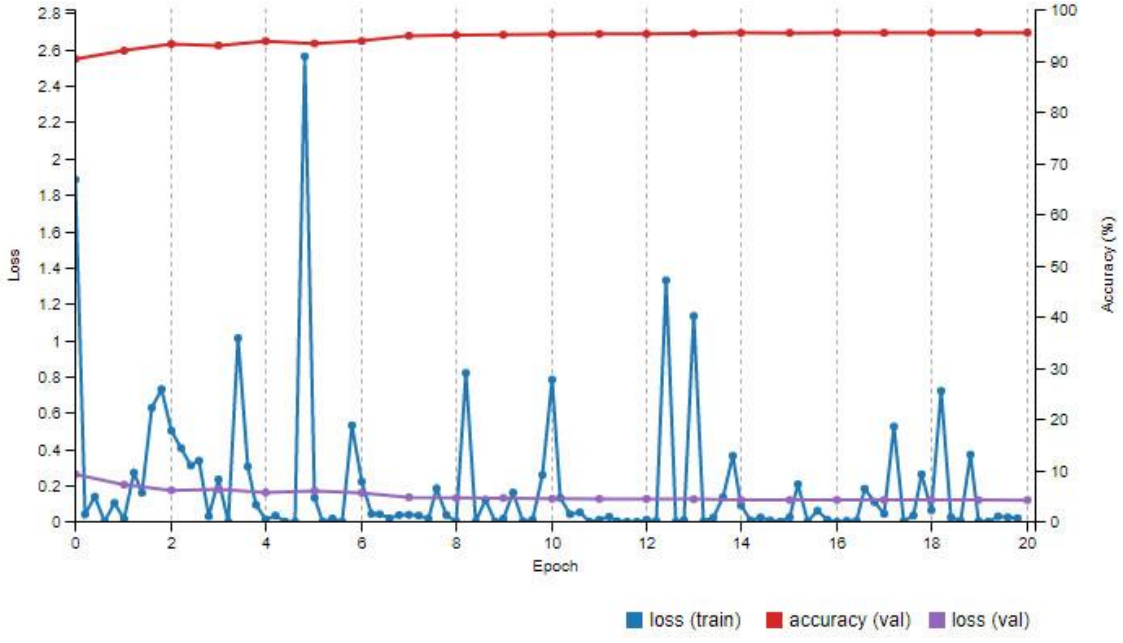
Çizelge 6.10. Eğitilen modellerin parametre bilgileri

Parametreler	***Model_1***	***Model_2***	***Model_3***
Veri Seti	Db_01	Db_02	Db_03
GPU	1050Ti	1050Ti	1080Ti
Eğitim Süresi (sa)	213	505	65
Döngü	30	20	30
test_iter	159857	448101	460548
test_interval	239785	672151	690822
max_iter	7193580	13443050	20724660
Mimari	GoogLeNet		
Eniyileyici	Adam		
Yığın Boyutu	2		16
Öğrenme Katsayısı İlkesi	Adım Sönümleme (Step Down)		
gamma	0,10000000149		
momentum	0,899999976158		
Weight_decay	9,99999997457 e-07		
Öğrenme Katsayısı	0,0001		

Şekil 6.17.'de model_1'in eğitim grafiği verilmiştir. 30. döngü sonunda accuracy (val) değeri, 92.2428 ile loss (train), 0,241523 ve loss (val), 0,195937 değerlerine ulaşılmıştır. Eğitim ve test verilerinde yapılan iyileştirmeler sonunda model_1'in eğitim sonundaki ağırlıkları başlangıç değerleri olarak alınarak, model_2 adıyla tekrar eğitildiğinde 20. döngü sonunda accuracy (val) değeri 95,5603, loss (train) 0,0199422, loss (val) 0,119326 değerlerine ulaşılmıştır. Eğitim grafiği Şekil 6.18.'de verilmiştir.

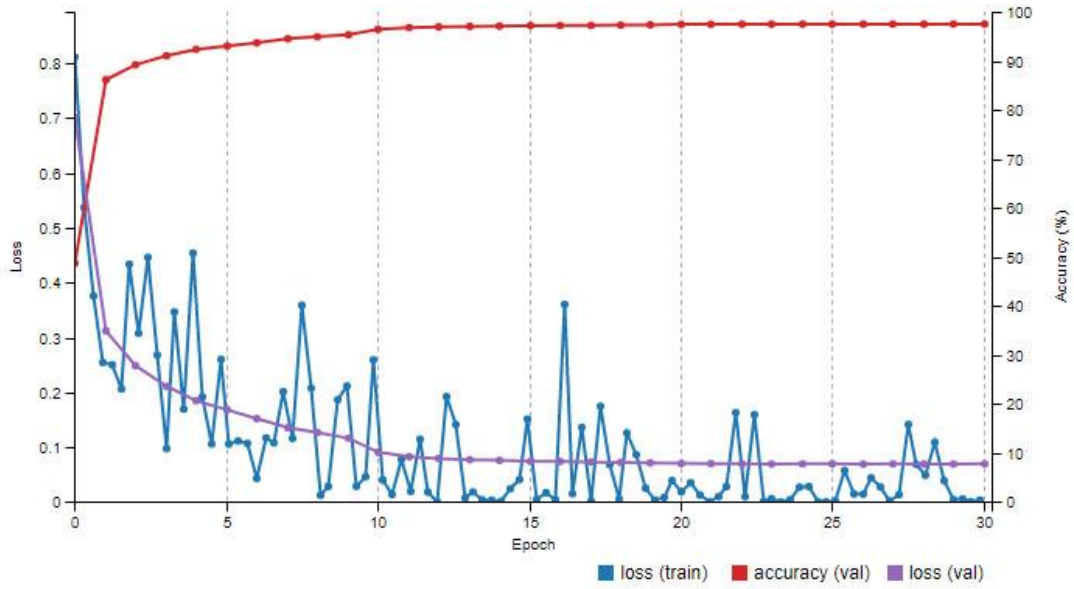


Şekil 6.17. Model_1 eğitim grafiği



Şekil 6.18. Model_2 eğitim grafiği

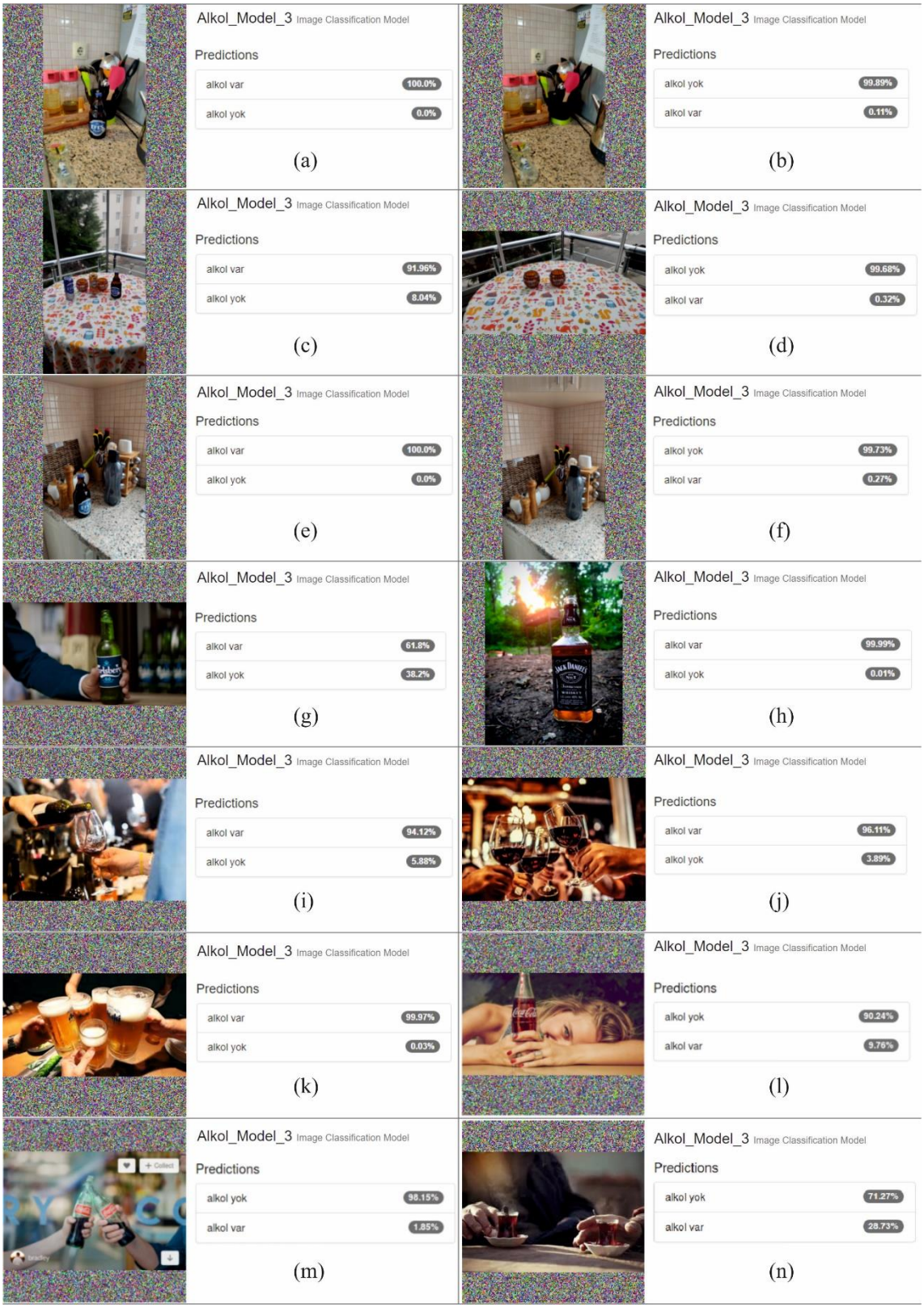
Hata durumlarına göre veri setinde tekrar iyileştirmeler yapılarak veri seti genişletilmiş ve model_3 olarak tekrar eğitilmiştir. Eğitim grafiği Şekil 6.19.'da verilmiştir. 30. döngü sonunda accuracy (val) değeri 97,6469, loss (train) 0,00384845, loss (val) 0,0697716 değerlerine ulaşılmıştır.



Şekil 6.19. Model_3 eğitim grafiği

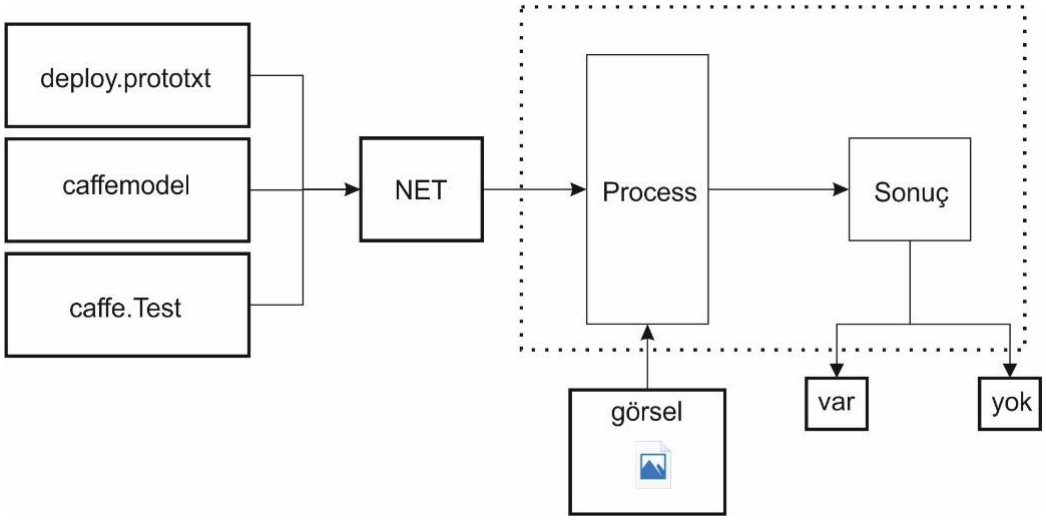
6.3.5. Modelin Test Edilmesi

Modelin ilk test aşaması olan tekil görsel testlerinde Digits arayüzünden faydalanılmıştır. Digits arayüzü ile evrişim, havuzlama ve normalizasyon gibi katmanların çıkış görüntüleri ayrı ayrı gözlemlenebilmektedir. Şekil 6.20'de tekil görsel üzerindeki alkol_var, alkol_yok sorgusuna ait sonuç ekran görüntüleri verilmiştir. Şekil 6.20'de bulunan (a), (b), (c), (d), (e) ve (f) etiketli görseller, aynı arka planda alkollü içecek bulunan ve bulunmayan durumlarda karşılaştırılmıştır. Bu karşılaştırma sonucunda tahmin oranları oldukça başarılıdır. Sadece ambalaj kıyaslamalarında değil, aynı zamanda karakteristik kadeh ve bardak görsellerinde de net olarak ayrım yapılabilmektedir. *EK-6*'da bazı katmanların çıkışlarının Digits arayüzü ile görselleri sunulmuştur.



Şekil 6.20. Digits arayüzü ile alkollü içecek tespit örnekleri

CAFFE kütüphanesi ile eğitilen model dosyaları, python ile görselleri yorumlayabilecek şekilde Şekil 6.21.'deki gibi düzenlenmiştir. PysimpleGui kütüphanesi kullanılarak bir görsel arayüz oluşturulmuştur. Bu oluşturulan arayüz, başarı durum hesaplaması için modelin test edilmesinde kullanılmıştır. Oluşturulan program arayüzü *EK-7* ve *EK-8*'de verilmiştir. Arayüz farklı konular seçilerek filtreleme yapılabilecek şekilde düzenlenmiştir.



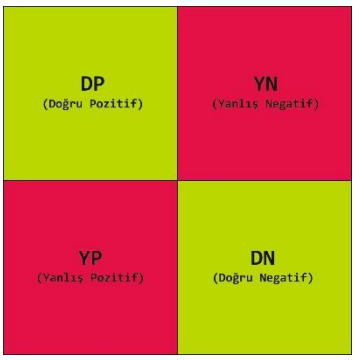
Şekil 6.21. Model test uygulama blok şeması

6.3.6. Modelin Karışıklık Matrisi Değerlendirmesi

Modelin eğitimi sırasında elde edilen doğruluk ve hata değerlerinin ne derece gerçek değerler ile örtüştüğünü tespit etmek, sadece eğitim sonuçlarına bakılarak anlaşılabilecek bir durum değildir. Bu nedenle eğitim setinde yer almayan gerçek veriler ile ikili karşılaştırma yapılarak doğru ve yanlış sınıflandırma tahmin değerlerinin karışıklık matrisi oranları hesaplanmıştır. Bu işlem için eğitilen modelin tahmin durumları Şekil 6.22.'de verilen gerçek pozitif, yanlış pozitif, doğru negatif ve yanlış negatif şeklinde sınıflandırılmıştır.

Modelin sınıflandırmasında *alkol_var* ve *alkol_yok* olarak iki sınıf belirlendiğinden sınıflandırma değerleri aşağıdaki tanımlamalara göre tespit edilmiştir;

- DP (doğru pozitif): İçerisinde alkollü içecek bulunan ve *alkol_var* olarak sınıflandırılan veriler.
- YN (yanlış negatif): İçerisinde alkollü içecek bulunan ancak *alkol_yok* olarak sınıflandırılan veriler.
- YP (yanlış pozitif): İçerisinde alkollü içecek bulunmayan ancak *alkol_var* olarak sınıflandırılan veriler.
- DN (doğru negatif): İçerisinde alkollü içecek bulunmayan ve *alkol_yok* olarak sınıflandırılan veriler.



Şekil 6.22. Karışıklık matrisi ikili sınıflandırma tablosu

Bu veriler doğrultusunda hesaplama metrikleri Çizelge 6.11.'de verilen doğruluk, kesinlik, geri çağırma, özgüllük ve f1 skoru başlıkları ile değerlendirilmiştir.

Doğruluk değeri, modelin genel doğruluk performansının göstergesidir. Kesinlik değeri, modelin doğru sınıflandırılan verilerinin oranını verir ve tahminlerin ne kadar kesin olduğunun göstergesidir. Negatif olduğu halde pozitif olarak sınıflandırılan veri, kesinlik değerinin hesaplanan oranını düşürmektedir. Geri çağırma değerleri, gerçek pozitif örneklerin oranını belirtir. Pozitif olduğu halde negatif sınıflandırılan veri, hesaplanan oranın geri çağırma değerini düşürmektedir. Özgüllük, gerçek negatif olan verilerin tahmin edilen negatif değerlere oranını gösterir. Negatif olduğu halde pozitif olarak sınıflandırılan veri hesaplanan özgüllük değerinin ortalama değerini düşürmektedir. F1 skoru, doğruluk ve geri çağırma değerlerinin harmonik ortalaması alınarak hesaplanır.

Karışıklık matrisi hesaplanmasında kullanılacak olan test verisi, eğitim, doğrulama veya test verisi olarak eğitim veri setinde yer almayan verilerden oluşmaktadır. Bu işlem için internet üzerinden temin edilen 154501 adet tekil görsel ile eğitilen model kullanılarak sınıflandırma işlemi gerçekleştirilmiştir. Burada kullanılan görseller, gerçek hayatta web sitelerinde karşılaşılabilecek türde veriler olup çeşitlilik açısından harmanlanmıştır. Çizelge 6.11'de karışıklık matrisi değerlendirme metrikleri verilmiştir (Stanford, 2021).

Çizelge 6.11. Karışıklık matrisi değerlendirme metrikleri

Metrik	***Formül***
Doğruluk	$\frac{DP + DN}{DP + DN + YP + YN}$
Kesinlik	$\frac{DP}{DP + YP}$
Geri Çağırma	$\frac{DP}{DP + YN}$
Özgüllük	$\frac{DN}{DN + YP}$
F1 skoru	$2.\frac{Kesinlik\,.\,GeriÇağırma}{Kesinlik\,+\,GeriÇağırma}$

Oluşturulan bu test veri seti yedi ayrı test ile, test başına düşen yaklaşık ortalama 22000 görsel ile gerçekleştirilmiştir. Gerçekleştirilen yedi ayrı test sonucunda DP, YP, DN, YN değerleri Şekil 6.23.'te verilmiştir. Test sonucunda elde edilen karışıklık matrisi değerleri Çizelge 6.12.'de verilmiştir.

test01		test02		test03		test04		test05		test06		test07	
2581	851	5337	2223	5299	615	7222	1876	4516	877	3349	885	4870	1099
99	18298	52	16295	103	15176	77	17452	71	13586	300	15763	132	15497

Şekil 6.23. Karışıklık matrisi hesaplaması için gerçekleştirilen yedi ayrı testin DP, YP, DN ve YN veri sayıları

Çizelge 6.12. Yapılan 7 ayrı testin kesinlik, geri çağırma, doğruluk, özgüllük ve f1 skor değerleri

Test No	*Kesinlik*	*Geri çağırma*	*Doğruluk*	*Özgüllük*	*F1*
Test01	0,963059701	0,752039627	0,956479912	0,994618688	0,844568
Test02	0,990350714	0,705952381	0,904839587	0,996818988	0,824311
Test03	0,980932988	0,896009469	0,966120889	0,993258721	0,93655
Test04	0,98945061	0,793800835	0,926653397	0,995607279	0,880893
Test05	0,984521474	0,837381791	0,95023622	0,994801201	0,90501
Test06	0,917785695	0,790977799	0,941616988	0,981323539	0,849677
Test07	0,973610556	0,815882057	0,943003982	0,991554162	0,887795
Ortalama	0,975476359	0,797451923	0,940065113	0,992612997	0,877526188

Gerçekleştirilen yedi ayrı teste ilave olarak, toplam 21938 görselden oluşan, yakın benzerlik ve uzak benzerlik veri setleri oluşturulmuştur. Burada yer alan verilerin de eğitimde kullanılan veri setinde kullanılmamış olmasına dikkat edilmiştir. Yakın benzerlik veri seti, ambalajlı olup alkolsüz içeceklerin bulunduğu görseller ile alkollü içeceklerin ambalajları arasında karma bir karşılaştırma yapmak için kullanılmıştır. Uzak benzerlik veri seti ise, ambalajlı alkollü içecekler ile doğa fotoğrafları arasında karma bir karşılaştırma yapmak için kullanılmıştır. DP, YP, DN, YN değerlerinin yer aldığı şekil, Şekil 6.24'te verilmiştir. Kesinlik, geri çağırma, doğruluk, özgüllük ve F1 skorları Çizelge 6.13'te sunulmuştur. Burada hesaplanan F1 skoru değerleri olan, 0,94196504 ve 0,94668967 ile gerçek hayat görsellerinden oluşan veri setinin, F1 skor test değerlerinden yüksek olduğu görülmüştür.



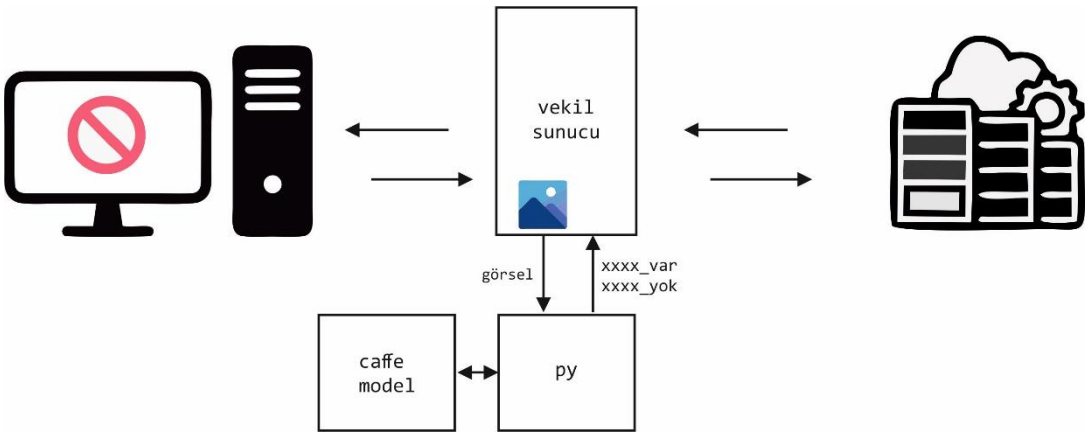
Şekil 6.24. Yakın benzerlik ve uzak benzerlik DP, DN, YP ve YN değerleri

Çizelge 6.13. Yakın ve Uzak benzerlik veri setleri ile yapılan sınıflandırmada elde edilen kesinlik, geri çağırma, doğruluk, özgüllük ve f1 skoru değerleri

	Kesinlik	*Geri Çağırma*	*Doğruluk*	*Özgüllük*	*F1 Skoru*
Yakın Benzerlik	0,983870968	0,903483073	0,925155925	0,969616027	0,94196504
Uzak Benzerlik	0,996367807	0,901730104	0,954137042	0,997293062	0,946689674

6.4. Güvenlik Duvarı Uygulaması

Oluşturulmak istenen filtreleme yapısının, görselleri algılayıp, değerlendirmesi beklenmektedir. Görsellerin zararlı içeriğe sahip olup olmadığının incelenmesi için ağ trafiğinin sürekli kontrol edilmesi gerekmektedir. Bu sebeple bir vekil sunucu oluşturma gereksinimi doğmuştur. Bütün trafik vekil sunucu üzerinden aktarılmaktadır, vekil sunucu, ağ trafiğinde tespit ettiği görselleri CAFFE model dosyasına yönlendirmektedir. CAFFE model dosyasından gelen sınıflandırma bilgisine göre görselin görüntülenip görüntülenmeyeceğine karar vermektedir. Uygulama blok şeması Şekil 6.25.'te verilmiştir.



Şekil 6.25. Uygulama blok şeması

Vekil sunucu yapısı için mitmproxy aracından faydalanılmıştır. Mitmproxy, http/1, http/2 ve websockets için etkileşimli bir SSL/TLS özellikli gözlemleme, değiştirme ve engelleme vekili sağlayan araçlar kümesidir (Mitmproxy, 2021). Mitmproxy ile tüm akış bir ara bellekte tutulmaktadır. Daha sonra istendiği takdirde bu

verilerin işlenmesine olanak tanımaktadır. Mitmproxy aracının en büyük avantajı, ağ trafiğine python komut dizileri müdahale edilebilmesine olanak tanımasıdır (Wang vd, 2020).

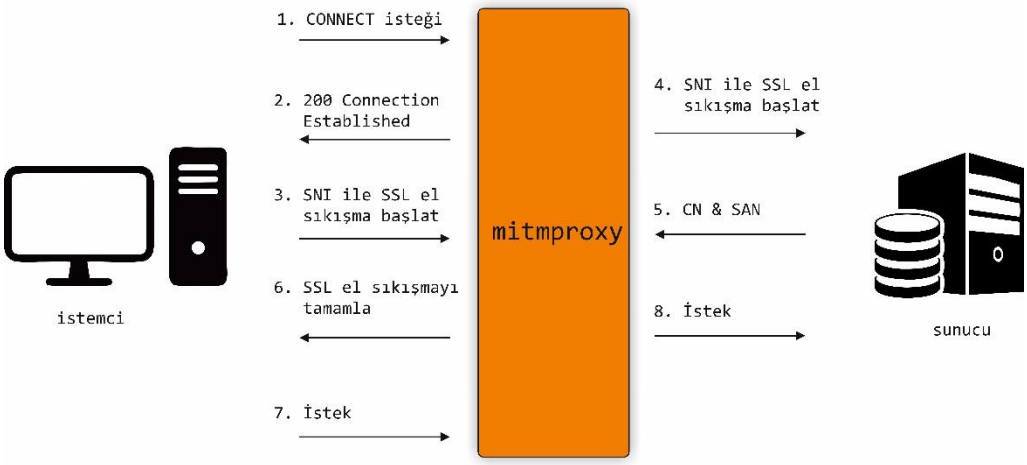
Sıradan bir http bağlantısında istemci, vekile "GET" yöntemi ile istekte bulunur ve vekil bunu sunucuya iletir. Mitmproxy için bir http bağlantısı aşağıdaki şekilde oluşturulabilir:

```
GET http://ornek.com/index.html HTTP/1.1
```

Açık https bağlantısında ise vekil, şifrelenmiş TLS veri akışını okuyamaz veya etkileyemez. Bunun için aşağıda verilen biçimde sadece basit bir bağlantı ile vekil, istemci ile sunucu arasında bir hat açılmasını sağlar.

```
CONNECT ornek.com:443 HTTP/1.1
```

Vekil bu bağlantı üzerinde sadece kolaylaştırıcı bir etkiden öte gidemez. Bu nedenle bağlantı üzerinden akan veri içeriği hakkında bilgi sahibi olunamaz. TLS ile bağlantı anlaşması bu hat üzerinden gerçekleştikten sonra bütün istek ve yanıtlar vekil için şeffaf hale gelmiş olur. Mitmproxy bu şeffaf bağlantı akışının tam ortasında yer almaktadır. Şifre çözme aşamasında, sunucuya istemci gibi, istemciye de sunucu gibi davranır. Sertifika yetkilisinden kaynaklanabilecek sorunlar için kendi sertifika yetkilisini kullanarak kendi kesme sertifikalarını oluşturmaktadır.



Şekil 6.26. Mitmproxy https bağlantı (Mitmproxy, 2021)

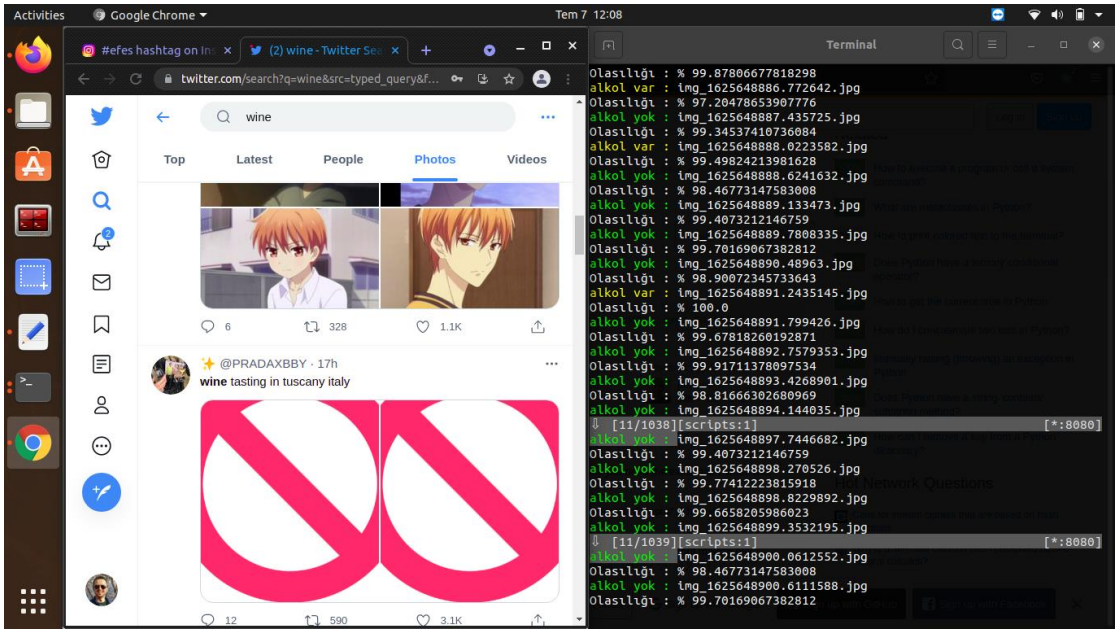
Mitmproxy ile Https bağlantı kurulması ve istek yanıt aşaması adımları Şekil 6.26.'da verilmiştir. Öncelikle istemci mitmproxy ile bağlantı kurar ve bir "HTTP CONNECT" isteğinde bulunur. 2. aşama olarak mitmproxy istemciye bağlantı kurulmuş gibi "200 Connection Established" yanıtı vererek istemciyi uzak sunucuyla konuştuğuna ikna eder. Bağlandığı ana bilgisayarın adını belirtmek için sunucu adı göstergesi (SNI) kullanarak taşıma katmanı güvenliği (TLS) bağlantısı başlatır. Bu şekilde mitmproxy sunucuya bağlanarak istemci tarafından belirtilen SNI ana bilgisayar adını kullanıp bir TLS bağlantısı kurmuş olur. Sunucu eşleşen sertifikaya ortak ad (CN) ve konu alternatif ad (SAN) değerlerini içeren bir yanıt verir. Bu adımdan sonra mitmproxy Şekil 6.26'da 3. Adımda duraklatılan bağlantı için sertifikasını oluşturur ve TLS anlaşması devam eder. İstemci ile sunucu arasında mitmproxy'nin aracılık ettiği bağlantı üzerinden istek ve yanıtlar aktarılarak iletişim gerçekleştirilir (Mitmproxy, 2021).

Mitmproxy aracı literatürde kabul görmüş olup güvenlik ve ağ trafiği analizi konusunda yapılan çalışmalarda etkili bir araç olarak kullanılmıştır. Ayrıca ismini aldığı "man in the middle" saldırısı için de kullanılabilecek etkili ve esnek bir araçtır.

Yapılan bir çalışmada, Android üzerinden yüklenen uygulamalarda kullanıcının kişisel bilgilerinin iletilmesi durumlarını saptamak amacıyla mitmproxy aracından faydalanıldığı bildirilmiştir (Wongwiwatchai, Pongkham & Sripanidkluchai, 2020).

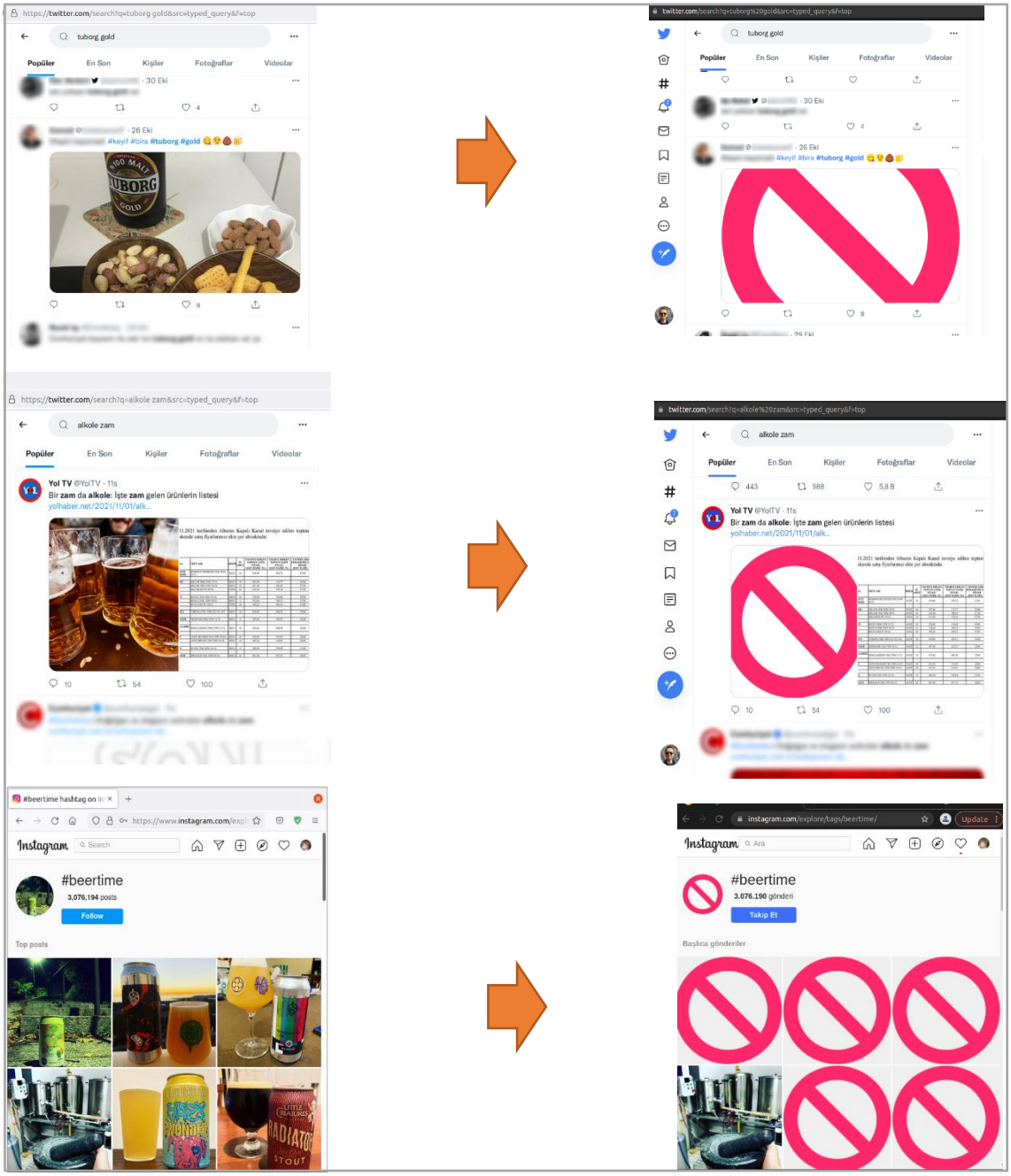
Xu vd. (2021) yaptıkları çalışmada, şifrelenmiş ağ trafiği üzerinde kötücül yazılımları tespit etmek için mitmproxy aracından faydalanmışlardır.

Modelin mitmproxy ile bütünleştirilmiş kullanımının ekran görüntüsü Şekil 6.27'de verilmiştir. Kullanıcı herhangi bir web sitesine istek gönderdiğinde, ağ üzerinden gönderilen görseller model tahmin sonucuna göre engellenmiştir. Bu görüntülerin engellendiğinin anlaşılması için engellenen görsel yerine görüntülenen bir blok ikon görseli seçilmiştir.



Şekil 6.27. Aktif güvenlik duvarı ile web içerik ekran görüntüleri ve terminal çıktısı

Şekil 6.28.'de Twitter ve Instagram üzerinden yapılan alkollü içecekler ile ilgili arama sonuçlarına göre görsellerin güvenlik duvarı aktifken ve pasifken alınan ekran görüntüleri sonuçları verilmiştir.



Şekil 6.28. Güvenlik duvarı aktifken ve pasifken web tarayıcıda görüntülenen sayfaların ekran görüntüsü

Güvenlik duvarı aktifken yapılan web sayfası görüntülemelerinde, derin öğrenme modelinin başarılı sonuç verdiği gözlemlenmiştir. Ancak arada, bir vekil sunucu kullanıldığından ötürü bağlantı hızında belirgin sayılmayacak düzeyde bir yavaşlama olması muhtemeldir. Belirgin olan yavaşlama ise görsellerin yüklenme hızında gözlemlenmiştir. Bu durum, trafik üzerinden gelen her görselin görüntülenmeden önce CAFFE model dosyasında sınıflandırma işleminin yapılmasından kaynaklanmaktadır. Ayrıca güvenlik duvarı uygulaması yapılan Intel i3 5500U işlemciye sahip sistemin grafik kartının yetersiz olması sebebiyle görsellerin yüklenmesindeki yavaşlık belirgin olmaktadır. Gerçek kullanım koşullarında ağ üzerinde, geliştirilen güvenlik duvarının kurulacak olduğu sunucunun grafik işlem kapasitesinin yüksek olması önerilmektedir.

BÖLÜM 7

7. SONUÇLAR VE TARTIŞMA

Bu bölüm sonuçlar ve tartışma olarak iki kısımda ele alınmıştır. Sonuçlar kısmında eğitim, test ve uygulamaya ilişkin elde edilen bulgular değerlendirilmiştir. Tartışma kısmında ise uygulama zorlukları, öneriler ve farklı uygulama alanları ortaya konulmuştur.

7.1. Sonuçlar

Önerilen uygulama geliştirme süreci, veri seti için görsel toplama, görseller ile veri seti oluşturma, hiper-parametre belirlenmesi ve iyileştirilmesi, eğitim, karmaşıklık matrisi kullanılarak başarı analizi ve güvenlik duvarı uygulaması olarak altı aşamada gerçekleştirilmiştir.

Ele alınan derin öğrenme modeli güvenlik duvarının zararlı içeriği filtrelemesi için ana gövde durumundadır. Filtreleme konusu olarak zararlı içeriklerden sadece alkollü içecekler ele alınmıştır. Ancak sadece alkollü içecekler konusu bile kendi içerisinde geniş bir örneklem çeşidi içerdiğinden, modelin başarılı şekilde eğitilebilmesi için veri seti olabildiğince geniş tutulmuştur. Eğitilen model alkollü içeceğin çeşidinden ziyade, sadece görsel üzerinde alkol var veya yok durumuna göre sınıflandırma yapabilmektedir. Alkol var durumu için sadece ambalaj ile sınırlı kalınmamış, aynı zamanda alkollü içecekler ile özdeşleşmiş karakteristik kadeh ve bardak şekilleri de veri setine dahil edilmiştir. Çocuk ve gençlerin algı gelişimini olumsuz etkileyebilecek görseller olarak sayılabileceğinden, alkollü içecek bulunan bar, meyhane veya parti ortamları ile de veri seti genişletilmiştir. Alkollü içecek var durumunun değerlendirilebilmesi içinse karşıt görseller olarak alkolsüz içecek ambalajları, karakteristik olmayan çay ve kahve bardakları ile alkolsüz ortamlardan oluşan görseller ile veri seti güçlendirilmiştir.

İlk örnek veri seti 80 bin görselden oluşturulmuş ve hiperparametrelerin belirlenmesi için eğitimlerde kullanılmıştır. Bu veri seti boyutunun küçük tutulmasının sebebi, asıl veri seti ile eğitim süreleri çok uzun olma ihtimalidir. Bu nedenle mümkün olan en kısa sürede hiper-parametreler üzerinde iyileştirme yapabilmek amacıyla test eğitimlerini gerçekleştirebilmek amaçlanmıştır.

İlk eğitim ve değerlendirmelerin ardından zararlı kategorisinde değerlendirilmeyecek olan makyaj malzemesi, balon, parfüm ambalajları ve benzeri gibi görsellerde modelin sınıflandırma sonuçlarında yanılma tespit edilmiştir. Bu doğrultuda veri seti yanılmanın yoğun olduğu görseller ile tekrar genişletilmiş ve üçüncü veri seti ile başarılı sonuçlar elde edilmiştir.

Veri seti için seçilen her görselin tekrar etme olasılığına karşı, ortalama kare fonksiyonu ile karşılaştırılmış, %5'ten daha az fark bulunan görseller veri setinden elenmiştir. Başarılı bir eğitim için tekil görsellerden oluşan veri setine veri çoğaltma işlemi uygulanmış ve toplamda üçüncü veri setinin boyutu 4,6 milyondan fazla görsel ile oluşturulmuştur.

Test aşamasında en iyi sonucu veren eniyileyici belirgin bir biçimde Adam algoritması olduğundan eğitimde bu algoritma tercih edilmiştir.

Eğitimde Digits 6.1 arayüzü kullanılmış olup CAFFE ortamı ile gerçekleştirilmiştir. Eğitim mimarisi olarak GoogLeNet inception yapısı kullanılmıştır. Eğitim sürecinde test ve iyileştirme adımları takip edilerek en iyi sonuç elde edilene kadar sırasıyla üç farklı modelde üç eğitim gerçekleştirilmiştir. Bu süreçte Nvidia Geforce 1050Ti ve Nvidia GeForce 1080Ti grafik kartlarına sahip iki farklı sistem kullanılarak toplamda 783 saat süren eğitim ile en iyi sonuç elde edilmiştir. Üçüncü veri seti ile eğitim sonunda accuracy değeri %97,6469 olarak belirlenmiş ve en iyi sonuca ulaşılmıştır. İdeal eğitim süresi için döngü miktarı 20 ile 30 arasında seçilebilir. Daha fazla döngü miktarının eğitim oranına çok yüksek bir etkisi olmamaktadır.

Karışıklık matrisi hesaplanmasında test verisi olarak 154501 tekil görsel kullanılmıştır. Test için kullanılan bu veri yedi ayrı test ile her test başına yaklaşık ortalama 20 bin görsel veri ile gerçekleştirilmiştir. Yaklaşık 20 bin veri ile elde edilen bu değerler tutarlı olarak çok küçük sapmalar ile sabit kalmıştır. Toplam değerler üzerinden hesaplama yapıldığında kesinlik: 0,975476359, geri çağırma: 0,797451923, doğruluk:

0,940065113, özgüllük: 0,992612997 ve f1 skoru: 0,877526188 olarak elde edilmiştir. Karışıklık matrisi hesaplama sonuçları ile tutarlı olarak test uygulamalarında da oldukça başarılı sonuçlar verdiği gözlemlenmiştir.

Ağ trafiğini kontrol edebilmek ve içeriğini okuyabilmek için bir vekil sunucuya ihtiyaç duyulmuştur. Bunun için python komutları ile ağ trafiğine esnek biçimde müdahale edilmesini sağlayan mitmproxy vekil sunucusu tercih edilmiştir. Mitmproxy sayesinde TLS şifreli veri akışı okunabilmekte ve python komutları ile müdahale edilebilmektedir. Veri akışında saptanan her görselin önce python komutları ile oluşturulan arayüz aracılığıyla ve CAFFE model dosyası sayesinde sınıflandırılır. Sınıflandırma sonucunda görselin zararlı olduğu veya olmadığı durumuna göre engellenmekte veya serbest bırakılmaktadır.

Facebook, Instagram, Twitter gibi sosyal medya sitelerinin de dahil olduğu birçok web sitesi üzerinde deneme gerçekleştirilmiş ve deneme sonucu ekran görüntülerine uygulama kısmında yer verilmiştir.

7.2. Tartışma

İnternet kullanımının telefon, bilgisayar ve tablet gibi birçok cihaz ile hayatımızda yer etmesi özellikle de iletişim ve haberleşme açısından bize kolaylık sağlaması artık vazgeçemeyeceğimiz bir alışkanlık haline gelmiştir. Ancak bu sonsuz veri akışı kimi zaman istemediğimiz veya zararlı gördüğümüz içeriğe maruz kalmamıza sebep olabilmektedir. Özellikle henüz kişilik gelişimini tamamlamamış genç bireyler üzerinde telafisi zor olacak hasarlara yol açabilmektedir. Zararlı içeriklerden korunabilmek için çeşitli engelleme ve kelime bazlı filtreleme mekanizmaları kullanılmaktadır. Ancak bu durumda bir web sitesinin tamamen engellenmesi durumu söz konusu olmaktadır. Ayrıca web sitesi içerik olarak zararsız olmasına rağmen dinamik reklamlar ile zararlı görseller aradan sızabilmektedir. Kökten bir engellemede ise farklı ticari sorunlar ortaya çıkabilmektedir.

Önerilen uygulama güvenlik duvarı modelinde, görsel içerik analizi yapılarak bir akıllı filtreleme mekanizması sunulmuştur. Bu akıllı güvenlik duvarı ile kullanıcı çeşitli açıdan sakıncalı gördüğü görsel içerikleri filtreleyebilmektedir. Uygulama modelinde sadece alkollü içecekler zararlı görsel olarak kabul edilip bu yönde bir model eğitilmiş ve

engelleme başarılı olarak gerçekleştirilmiştir. Bununla birlikte aynı mekanizmaya uygun veri setleri ile tütün, ateşli silahlar, şiddet ve +18 yaş yetişkin içerikler gibi çeşitli modeller dahil edilebilir. Gerektiğinde her filtreleme kuralı ayrı ayrı açılabilir veya kapanabilir.

Eğitimin başarılı sonuç verebilmesi için problem türüne göre veri seti boyutu çok büyük önem kazanmaktadır. Mevcut imkanlar ile en yüksek başarı oranına yaklaşmak ancak 4,6 milyon görsel veri ile mümkün olabilmiştir. F1 skoru %87 olarak hesaplanmıştır. Bu şekilde bile model başarısı %100 değildir. Fakat problemin zorluğu göz önüne alındığında yeterli bir başarı oranı olarak kabul edilebilir. Diğer taraftan 20 bin adet altında görsel verilerde sapma oranları çok daha yüksek olabilmektedir. Gerçek hayat uygulamasında bir web sitesi üzerinde görüntülenecek görsellerin sayısı çok daha küçük olduğundan, aradaki yanlış pozitif ve yanlış negatif değerlerinin sadece bir veya iki tane olması durumunda dahi hata oranları daha yüksek çıkabilmektedir. Ancak hiç hatalı sınıflandırma yapılmaması durumu da olabilir, bu durumda ise başarı oranı %100 çıkacaktır. Dolayısıyla daha fazla veri ile model başarı oranları arttırılabilir. Ateşli silahlar, tütün veya +18 yaş yetişkin içeriklerin görsel çeşitleri alkollü içecekler veri setine göre daha az olabilir ancak ele alınan alkollü içeceklerin tespiti konusunda durum böyle değildir. Mevcut alkollü içecekler markalarına ilave olarak yeni markalar çıkabilir veya bir markanın ambalaj türü değişebilir. Bu değişim ise alkolsüz bir içeceğin ambalaj türüne benzerlik gösterebilir. Tez çalışması sırasında var olan mevcut içecekler konusunda yeterli çalışma yapılmıştır. Ancak modelin sürekli bir şekilde başarılı tahmin yapabilmesi için belirli aralıklarla veri setinin güncellenerek, değişen alkollü veya alkolsüz içecekler ile eğitimin tekrarlanması gerekmektedir. Bu durumun tez çalışması için seçilen problemin ne derece zorlu olduğunun göstergesi olduğu söylenebilir. Diğer taraftan gelişen donanım teknolojisiyle web üzerinden sürekli temin edilen görseller yoluyla güncellenen veri seti ile daha kısa sürelerde eğitim işlemlerinin gerçekleşmesi mümkün olacaktır.

Güvenlik duvarı, Intel 5005U işlemciye sahip grafik işlemci açısından yetersiz bir diz üstü bilgisayarda uygulanmıştır. Önerilen güvenlik duvarı modelinde özellikle görsel sınıflandırma işleminden dolayı görsellerin yüklenme hızı belirgin şekilde yavaşlamaktadır. Görselin sınıflandırma işlemi için grafik işlemci desteği önemlidir. Ağ üzerinde vekil sunucuya kurulum yapılabilir ancak verimli bir kullanım için sunucunun

işlem kapasitesinin yüksek olması önerilmektedir. Zaman içerisinde donanım teknolojisinde yaşanacak iyileştirmeler ve gelişmeler ile önerilen güvenlik duvarının bağlantı hızına etkisi fark edilmeyecek kadar düşük olabilir. Böylelikle internetin zararlı içeriklerinden daha etkin bir şekilde koruma sağlanabilir.

Eğitilen model dosyasını vekil sunucu ile oluşturulacak bir güvenlik duvarı haricinde, tarayıcı eklentisinde kullanmak da mümkündür. Ayrıca video görüntülerinin içinde akıllı bir filtreleme ve düzenleme mekanizması kullanılarak görüntü içindeki sakıncalı görünen kısımlar filtrelenebilir.

KAYNAKLAR

Abuqaddom I., Mahafzah B. A. & Faris H. (2021). Oriented stochastic loss descent algorithm to train very deep multi-layer neural networks without vanishing gradients. *Knowledge-Based Systems* (Vol. 230). doi: https://doi.org/10.1016/j.knosys.2021.107391

Akca M. (2020). Gradient Descent Nedir? 08 Şubat 2022 tarihinde https://medium.com/deep-learning-turkiye/gradient-descent-nedir-3ec6afcb9900 adresinden erişildi.

Alhanaee K., Alhammadi M., Almenhali N. & Shatnawi M. (2021). Face Recognition Smart Attendance System using Deep Transfer Learning. *Procedia Computer Science* (Vol. 192 pp. 4093-4102). Doi: https://doi.org/10.1016/j.procs.2021.09.184

Alhogail A. & Alsabih A. (2021). Applying machine learning and natural language processing to detect phishing email. *Computers & Security* (Vol. 110). Doi: https://doi.org/10.1016/j.cose.2021.102414

Apicella A., Donnarumma F., Isgro F. & Prevete R. (2021). A survey on modern trainable activation functions. *Neural Networks* (Vol. 138 pp. 14-32). doi: https://doi.org/10.1016/j.neunet.2021.01.026

Arı A., Berberler M. E. (2017). Yapay Sinir Ağları ile Tahmin ve Sınıflandırma Problemlerinin Çözümü İçin Arayüz Tasarımı. *Acta Infologica* (Vol. 1 I. 2 pp. 55-73). Erişim adresi: https://dergipark.org.tr/tr/pub/acin/issue/33868/335553

Atalay M. & Çelik E. (2017). Büyük Veri Analizinde Yapay Zekâ ve Makine Öğrenmesi Uygulamaları. *Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi* (Vol. 9, I. 22, pp. 155-172). doi: https://doi.org/10.20875/makusobed.309727

Augmentor (2021). 8 Kasım 2021 tarihinde https://augmentor.readthedocs.io/en/master/userguide/mainfeatures.html adresinden erişildi.

Aydın S. & Özkul E. (2015). Veri Madenciliği ve Anadolu Üniversitesi Açıköğretim Sisteminde Bir Uygulama. *Eğitim ve Öğretim Araştırmaları Dergisi* (Vol. 4 I. 3). ISSN: 2146-9199

Ayyüce Kızrak M. & Bolat B. (2018). Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma. *Bilişim Teknolojileri Dergisi* (Vol. 11, I. 3 pp. 263-286). doi: https://doi.org/10.17671/gazibtd.419205

Bertini F., Allevi D., Lutero G., Calza L. & Montesi D. (2022). An automatic Alzheimer's disease classifier based on spontaneous spoken English. *Computer Speech & Language* (Vol. 72). doi: https://doi.org/10.1016/j.csl.2021.101298

Bezdan, T. & Bačanin Džakula, N. (2019). Convolutional Neural Network Layers and Architectures. *Sinteza 2019-International Scientific Conference on Information Technology and Data Related Research*. doi:10.15308/Sinteza-2019-445-451

Bi L., Zhao H. & Jia M. (2016). Database-oriented storage based on LMDB and linear octree for massive block model. *Transactions of Nonferrous Metals Society of China* (Vol. 26 I. 9 pp. 2462-2468). doi: https://doi.org/10.1016/S1003-6326(16)64377-7

Bose I. & Mahapatra R. K. (2000). Business data mining — a machine learning perspective. *Information & Management* (Vol. 39, I. 3, pp. 211-225). doi: https://doi.org/10.1016/S0378-7206(01)00091-X

Caffe (2021). 8 Şubat 2022 tarihinde https://caffe.berkeleyvision.org/ adresinden erişildi.

Çubukçu A. & Bayzan Ş. (2013). Türkiye'de Dijital Vatandaşlık Algısı ve Bu Algıyı İnternetin Bilinçli, Güvenli ve Etkin Kullanımı ile Artırma Yöntemleri. *Middle Eastern & African Journal of Educational Research* (Vol. 5 I. 5 pp 148-174). Erişim adresi: https://www.researchgate.net/publication/348158490_Turkiye'de_Dijital_Vatandaslik_Algisi_ve_Bu_Algiyi_Internetin_Bilincli_Guvenli_ve_Etkin_Kullanimi_Ile_Artirma_Yontemleri

Dawoud, A., Shahristani, S. & Raun C. (2018). Deep learning and software-defined networks: Towards secure IoT architecture, *Internet of Things* (Vol. 3-4 pp. 82-89). doi: 10.1016/j.iot.2018.09.003

Dhomne A., Kumar R. & Bhan V. (2018). Gender Recognition Through Face Using Deep Learning. *Procedia Computer Science* (Vol. 132 pp. 2-10). doi: https://doi.org/10.1016/j.procs.2018.05.053

Dikaiakos M. D., Stassopolou A. & Papageorgiou L. (2005). An investigation of web crawler behavior: characterization and metrics. *Computer Communications* (Vol. 28 I. 8 pp. 880-897). doi: https://doi.org/10.1016/j.comcom.2005.01.003

Dong S., Wang P. & Abbas K (2021). A survey on deep learning and its applications. *Computer Science Review* (Vol. 40). doi: https://doi.org/10.1016/j.cosrev.2021.100379

Dozat T. (2016). Incorporatıng Nesterov Momentum into Adam. 6th International Conference on Learning Representations. Erişim adresi: https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ

Du X, Cai Y., Wang S. & Zhang L. (2016). Owerview of deep learning, *31st Youth Academic Annual Conference of Chinese Association of Automation* (pp. 159-164). doi: 10.1109/YAC.2016.7804882

Ertam F. (2019). An Efficient Hybrid Deep Learning Approach for Internet Security. *Physica A: Statistical Mechanics and its Applications* (Vol. 535). doi: https://doi.org/10.1016/j.physa.2019.122492

Eweje F. R., Bao B., Wu J., Dalal D., Liao W., He Y., Luo Y., Lu S., Zhang P., Peng X., Sebro R., Bai H. X. & States L. (2021). Deep Learning for Classification of Bone Lesions on Routine MRI. *EBioMedicine* (Vol. 68). doi: https://doi.org/10.1016/j.ebiom.2021.103402

Fahad S., Ranjan A., Yadav J. & Deepak A. (2021). A survey of speech emotion recognition in natural environment. *Digital Signal Processing* (Vol. 110). doi: https://doi.org/10.1016/j.dsp.2020.102951

Fujiyoshi H., Hirakawa T. & Yamashita T. (2019). Deep learning-based image recognition for autonomous driving. *The official Journal of the International Association of Traffic and Safety Sciences* (Vol. 43 I.4 pp. 244-252). doi: https://doi.org/10.1016/j.iatssr.2019.11.008

Ghosh A., Sufian A., Sultana F., Chakrabarti A. & De D. (2020). Fundamental Concepts of Convolutional Neural Network. Balas V., Kumar R., Srivastava R. (Ed), *Recent Trends and Advances in Artificial Intelligence and Internet of Things* (Vol 172 pp. 519-567). doi: https://doi.org/10.1007/978-3-030-32644-9_36

Gouda, M. G. & Liu, A. X. (2006). Structured firewall design, *Computer Networks* (Vol. 51 I. 4 pp. 1106 – 1120). doi: 10.1016/j.comnet.2006.06.015

Grassmann F., Mangelkamp J., Brandl C., Harsch S., Zimmermann M. E., Linkohr B., Peters A., Heid I. M., Palm C. & Weber B. H. F. (2018). A Deep Learning Algorithm for Prediction of Age-Related Eye Disease Study Severity Scale for Age-Related Macular Degeneration from Color Fundus Photography. *Ophthalmology* (Vol. 125 pp. 1410-1420). doi: https://doi.org/10.1016/j.ophtha.2018.02.037

Gupta A., Anpalagan A., Guan L. & Khwaja A. S. (2021). Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array* (V. 10). doi: https://doi.org/10.1016/j.array.2021.100057

Gülcü A. & Kuş Z. (2019). A Survey of Hyper-parameter Optimization Methods in Convolutional Neural Networks. *Gazi University Journal of Science Part C: Design and Technology* (Vol. 7 I. 2 pp. 503-522). doi: 10.29109/gujsc.514483

Hacıefendioğlu K., Başağa H. B., Yavuz Z. & Karimi M. T. (2021). Intelligent ice detection on wind turbine blades using semantic segmentation and class activation map approaches based on deep learning method. *Renewable Energy*. doi: https://doi.org/10.1016/j.renene.2021.10.025

Hafiz R., Haque M. R., Rakshit A. & Uddin M. S. (2020). Image-based soft drink type classification and dietary assessmentsystem using deep convolutional neural network with transfer learning. *Journal of King Saud University – Computer and Information Sciences*. doi: https://doi.org/10.1016/j.jksuci.2020.08.015

Han X., Dong Z. & Yang B. (2021). A point-based deep learning network for semantic segmentation of MLS point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* (Vol. 175 pp. 199-214). doi: https://doi.org/10.1016/j.isprsjprs.2021.03.001

Hannousse A. & Yahiouche S. (2021). Towards Benchmark Datasets For Machine Learning Based Website Phishing Detection: An Experimental Study. *Engineering Applications of Artificial Intelligence* (Vol. 104). doi: https://doi.org/10.1016/j.engappai.2021.104347

Hassan M. (2018). VGG16 – Convolutional Network for Classification and Detection. 8 Şubat 2022 tarihinde https://neurohive.io/en/popular-networks/vgg16/ adresinden erişildi.

Hassan R., Islam F., Uddin Z., Ghoshal G., Hassan M. M., Huda S. & Fortino G. (2021). Prostate cancer classification from ultrasound and MRI images using deep learning based Explainable Artificial Intelligence. *Future Generation Computer Systems*. doi: https://doi.org/10.1016/j.future.2021.09.030

Hespeler S. C., Nemati H., Dehghan-Niri E. (2021). Non-destructive thermal imaging for object detection via advanced deep learning for robotic inspection and harvesting of chili peppers. *Artificial Intelligence in Agriculture* (Vol. 5 pp. 102-117). doi: https://doi.org/10.1016/j.aiia.2021.05.003

Hinton G. E. (2007). Learning Multiple Layers Ofrepresentation. *Trends in Cognitive Sciences* (Vol. 11, I. 10, pp. 428-434). doi: https://doi.org/10.1016/j.tics.2007.09.004

Hinton G. E., Osindero S. & Teh Y. W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation* (Vol. 18, I. 7). doi: 10.1162/neco.2006.18.7.1527.

Hinton G., Srivastava N. & Swersky K. (2012). Lecture 6a Overview of mini-batch Gradient descent [PDF belgesi]. 08 Şubat 2022 tarihinde https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf adresinden erişildi.

Hu D., Hu X., Jiang W., Zheng S. & Zhao Z. (2019). Intelligent Digital Image Firewall System for Filtering Privacy or Sensitive Images. *Cognitive Systems Research* (Vol. 53 pp. 85-97). doi: https://doi.org/10.1016/j.cogsys.2018.03.003

Hu H., Pang L. & Shi Z. (2016). Image matting in the perception granular deep learning. *Knowledge-Based Systems* (Vol. 102 pp. 51-63). doi: https://doi.org/10.1016/j.knosys.2016.03.018

Huang P., Wen C., Fu L., Peng Q. & Tang Y. (2020). A deep learning approach for multi-attribute data: A study of train delay prediction in railway systems. *Information Sciences* (Vol. 516 pp. 234-253). doi: https://doi.org/10.1016/j.ins.2019.12.053

Image-net (2021). 08 Şubat 2022 tarihinde https://www.image-net.org/challenges/LSVRC/index.php adresinden erişildi.

Iqbal M., Sameem M. S. I., Naqvi N., Kanwal S. & Zhongfu Ye (2019). A deep learning approach for face recognition based on angularly discriminative features. *Pattern Recognition Letters* (Vol. 128 pp. 414-419). doi: https://doi.org/10.1016/j.patrec.2019.10.002

Ivakhnenko A. G. & Lapa V. G. (1967). Cybernetics and Forecasting. *Nature* (Vol. 219, pp. 202-203). doi: https://doi.org/10.1038/219202b0

İplikçi H. G. & Batu M. (2018). Dijital İletişim ve Çocuk: Türkiye'de Çocuk Web Sitelerindeki Reklamlara Yönelik Bir İçerik Analizi. *Akdeniz Üniversitesi İletişim Fakültesi Dergisi* (Vol. 29 pp. 242-256). doi: https://doi.org/10.31123/akil.454440

Jeon E., Kim S., Park S., Kwak J. Choi I. (2021). Semantic segmentation of seagrass habitat from drone imagery based on deep learning: A comparative study. *Ecological Informatics*. doi: https://doi.org/10.1016/j.ecoinf.2021.101430

Jeon M., Lee Y., Shin Y. S., Jang H. & Kim A. (2019). Underwater Object Detection and Pose Estimation using Deep Learning. *International Federation of Automatic Control-PapersOnLine* (Vol. 52 I. 21 pp. 78-81). doi: https://doi.org/10.1016/j.ifacol.2019.12.286

Jia Y., Shelhamer E., Donahue J., Karayev S., Long J., Girshick R., Guadarrama S. & Darrell T. (2014). Caffe: Convolutional Architecture for Fast Feature Embedding. *MM '14: Proceedings of the 22nd ACM international conference on Multimedia* (pp. 675-678). doi: https://doi.org/10.1145/2647868.2654889

Jiang M., Zhai F. & Kong J. (2021). A novel deep learning model DDU-net using edge features to enhance brain tumor segmentation on MR images. *Artificial Intelligence in Medicine* (Vol.121). doi: https://doi.org/10.1016/j.artmed.2021.102180

Kakehi S., Sekiuchi T., Ito H., Ueno S., Takeuchi Y., Suzuki K. & Togawa M. (2021). Identification and counting of Pacific oyster Crassostrea gigas larvae by object detection using deep learning. *Aquacultural Engineering* (Vol. 95). doi: https://doi.org/10.1016/j.aquaeng.2021.102197

Kanbur B. N. (2020). Görsel Medyanın ve Subliminal Mesajların Çocuk Sağlığı Üzerine Etkileri. *İstanbul Gelişim Üniversitesi Sağlık Bilimleri Dergisi* (Vol. 10 pp. 94-106). doi: https://doi.org/10.38079/igusabder.665516

Kandel I. & Castelli M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express* (Vol. 6 I. 4 pp. 312-315). doi: https://doi.org/10.1016/j.icte.2020.04.010

Karim R. (2018, 22 Ekim). 10 Stochastic Gradient Descent Optimisation Algorithms + Cheat Sheet. 8 Şubat 2022 tarihinde https://towardsdatascience.com/10-gradient-descent-optimisation-algorithms-86989510b5e9 adresinden erişildi.

Kasthuri E. & Balaji S. (2021). Natural language processing and deep learning chatbot using long short term memory algorithm. *Materials Today: Proceedings*. doi: https://doi.org/10.1016/j.matpr.2021.04.154

Kingma D. P. & Ba J. L (2015). Adam: A Method for Stochastıc Optimization. *3rd International Conference for Learning Representations*. Erişim adresi: https://arxiv.org/abs/1412.6980

Koh J., Yoon Y., Kim S., Han K. & Kim E. K. (2021). Deep Learning for the Detection of Breast Cancers on Chest Computed Tomography. *Clinical Breast Cancer*. doi: https://doi.org/10.1016/j.clbc.2021.04.015

Krizhevsky A., Sutskever I. & Hinton G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM* (Vol. 60 pp. 84-90). doi: https://doi.org/10.1145/3065386

Kumar H., Solanki A. K. & Singh K. K. (2020). Progressive Machine Learning Approach with WebAstro for Web Usage Mining, *Procedia Computer Science*, (Vol. 167 pp. 1400-1410). doi: https://doi.org/10.1016/j.procs.2020.03.351

Kumar K. K., Kasiviswanadham Y., Indira D. V. S. N. V., Palesetti P. P. & Bhargavi V. (2021). Criminal face identification system using deep learning algorithm multi-task cascade neural network (MTCNN). *Materials Today: Proceedings*. Doi: https://doi.org/10.1016/j.matpr.2021.06.373

Lan X., Zou C., Kang Z. & Wu X. (2021). Log facies identification in carbonate reservoirs using multiclass semi-supervised learning strategy. *Fuel* (Vol.302). doi: https://doi.org/10.1016/j.fuel.2021.121145

Lauriola I., Lavelli A. & Aiolli F. (2021). An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools. *Neurocomputing*. Doi: https://doi.org/10.1016/j.neucom.2021.05.103

Ławrynowıcz A. & Tresp V. (2014). Introducing Machine Learning. Jens Lehmann, Johanna Voelker (Ed.). *Perspectives on Ontology Learning* (Vol. 18 pp.35-50). Erişim adresi: https://www.researchgate.net/publication/268804320_Introducing_Machine_Learning

Le, Q., Boydel, O., Namee, B. M. & Scanlon M. (2018). Deep learning at the shallow end: Malware classification for non-domain experts, *Digital Investigation* (Vol. 26 pp. 118-126). doi: 10.1016/j.diin.2018.04.024

Lecun Y., Bottou L., Bengio Y. & Haffner P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE* (Vol. 86 I. 11 pp. 2278 – 2324). doi: https://doi.org/10.1109/5.726791

Lee H., Grosse R., Ranganath R. & Ng A. Y. (2011). Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks. *Communications of the ACM* (Vol. 54 I. 10 pp. 95-103). doi: https://doi.org/10.1145/2001269.2001295

Li G., Yang Y., Qu X., Cao D. & Li K. (2021). A deep learning based image enhancement approach for autonomous driving at night. *Knowledge-Based Systems* (Vol. 213). doi: https://doi.org/10.1016/j.knosys.2020.106617

Li Y., Zhang Q. & Yoon S. W. (2021). Gaussian process regression-based learning rate optimization in convolutional neural networks for medical images classification. *Expert Systems with Applications* (Vol. 184). doi: https://doi.org/10.1016/j.eswa.2021.115357

Lin M., Chen Q. & Yan S. (2014). Network In Network. ICLR 2014 konferasında bildiri olarak sunulmuştur. Erişim adresi: https://arxiv.org/abs/1312.4400

Liu Y., Wang X., Wang L. & Liu D. (2019). A modified leaky ReLU scheme (MLRS) for topology optimization with multiple materials. *Applied Mathematics and Computation* (Vol. 352 pp. 188-204). doi: https://doi.org/10.1016/j.amc.2019.01.038

Loh D. R., Yong W. X., Yapeter J., Subburaj K. & Chandramohanadas R. (2021). A deep learning approach to the screening of malaria infection: Automated and rapid cell counting, object detection and instance segmentation using Mask R-CNN. *Computerized Medical Imaging and Graphics* (Vol. 88). doi: https://doi.org/10.1016/j.compmedimag.2020.101845

Lopez - Sanchez D., Arrieta A. G. & Corchado J. M. (2019). Visual Content-Based Web Page Categorization With Deep Transfer Learning and Metric Learning, *Neurocomputing* (Vol. 338 pp. 418-431). doi: https://doi.org/10.1016/j.neucom.2018.08.086

Lu L., Shin Y., Su Y. & Karniadakis G. E. (2020). Dying ReLU and Initialization: Theory and Numerical Examples. *Communications in Computational Physics* (Vol. 28 I. 5 pp. 1671-1706). doi: 10.4208/cicp.OA-2020-0165

Maldonado-Cruz E. & Pyrcz M. J. (2021). Tuning machine learning dropout for subsurface uncertainty model accuracy. *Journal of Petroleum Science and Engineering* (Vol. 205). doi: https://doi.org/10.1016/j.petrol.2021.108975

McCulloch W. S. & Pitts W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* (Vol. 5 pp. 115-133). doi: https://doi.org/10.1007/BF02478259

Mekrache A., Bradai A., Moulay E. & Dawaliby S. (2021). Deep reinforcement learning techniques for vehicular networks: Recent advances and future trends towards 6G. *Vehicular Communications*. doi: https://doi.org/10.1016/j.vehcom.2021.100398

Mhaskar H. N. & Micchelli C. A. (1993). How to Choose an Activation Function. *Advances in Neural Information Processing Systems* (Vol. 6). Erişim adresi: https://proceedings.neurips.cc/paper/1993/hash/51ef186e18dc00c2d31982567235c559-Abstract.html

Mitmproxy (2021). 8 Şubat 2022 tarihinde https://docs.mitmproxy.org/stable/ adresinden erişildi.

Muehlebach M. & Jordan M. I. (2019). A Dynamical Systems Perspective on Nesterov Acceleration. Erişim adresi: https://arxiv.org/abs/1905.07436

Mulahuwaish A., Gynorick K., Ghafoor K. Z., Maghdid H. S. & Rawat D. B. (2020). Efficient Classification Model of Web News Documents Using Machine Learning Algorithms for Accurate Information. *Computer & Security* (Vol. 98). doi: https://doi.org/10.1016/j.cose.2020.102006

Mutasa S., Sun S & Ha R. (2020). Understanding artificial intelligence based radiology studies: What is overfitting? *Clinical Imaging* (Vol. 65 pp. 96-99). doi: https://doi.org/10.1016/j.clinimag.2020.04.025

Nandini G. S., Kumar A. P. S. & Chidananda K. (2021). Dropout technique for image classification based on extreme learning machine. *Global Transitions Proceedings* (Vol. 2 I. 1 pp. 111-116). doi: https://doi.org/10.1016/j.gltp.2021.01.015

Nasir J. A., Khan O. S., Varlamis I. (2021). Fake News Detection: A hybrid CNN-RNN Based Deep Learning Approach. *International Journal of Information Management Data Insights* (Vol. 1 I. 1). doi: https://doi.org/10.1016/j.jjimei.2020.100007

Nengroo A.S. & Kuppusamy K. S. (2018). Machine learning based heterogeneous web advertisements detection using a diverse feature set, *Future Generation Computer Systems* (Vol. 89 pp. 68-77). doi: https://doi.org/10.1016/j.future.2018.06.028

Niu H., Xu W., Akbarzadeh H., Parvin H., Beheshti A. & Alinejad-Rokny H. (2020). Deep feature learnt by conventional deep neural network. *Computers & Electrical Engineering* (Vol. 84). doi: https://doi.org/10.1016/j.compeleceng.2020.106656

Nvidia (2021). 8 Şubat 2022 tarihinde https://developer.nvidia.com/digits adresinden erişildi.

Ocw.mit (2011). 8 Şubat 2022 tarihinde https://ocw.mit.edu/resources/res-6-007-signals-and-systems-spring-2011/lecture-notes/MITRES_6_007S11_lec04.pdf adresinden erişildi.

Openstax (2021). 8 Şubat 2022 tarihinde https://openstax.org/books/biology/pages/35-1-neurons-and-glial-cells adresinden erişildi.

Özcan, H. (2014). Çok Düşük Çözünürlüklü Yüz İmgelerinde Derin Öğrenme Uygulamaları. Deniz Harp Okulu/Deniz Bilimleri ve Mühendisliği Enstitüsü, İstanbul, Erişim adresi: https://tez.yok.gov.tr/UlusalTezMerkezi/tezDetay.jsp?id=wHhd3C71GU3CkJ9k6NArwA&no=wanQd3YBt5lJIz9v52oe1Q

Özhan, E. (2013). Güvenlik Duvarı Günlüklerinin Makine Öğrenmesi Yöntemleri ile Analizi ve Bir Model Çıkartılması (Yayımlanmamış doktora tezi). Trakya Üniversitesi/Fen Bilimleri Enstitüsü, Edirne, Erişim adresi: https://tez.yok.gov.tr/UlusalTezMerkezi/TezGoster?key=1zw6GvYMe-q3Hf6HR-3US-svYL7_8nzhhMZ5YjEt7ATh6sjjIDg8txaEvabtpRPb

Pathak, A. R., Pandey, M. & Rautaray, S. (2018). Application of Deep Learning for Object Detection. *Procedia Computer Science* (Vol. 132 pp. 1706 – 1717). doi: 10.1016/j.procs.2018.05.144

Plappert M., Mandery C & Asfour T. (2018). Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks. *Robotics and Autonomous Systems* (Vol.109 pp. 13-26). doi: https://doi.org/10.1016/j.robot.2018.07.006

Possemiers A. & Lee I. (2021). Evaluating deep learned voice compression for use in video games. *Expert Systems with Applications* (V.181). doi: https://doi.org/10.1016/j.eswa.2021.115180

Ruder S. (2017). An overview of gradient descent optimization algorithms. Erişim adresi: https://arxiv.org/abs/1609.04747v2

Rumelhart D. E., Hinton G. E. & Williams R. J. (1986). Learning representations by back-propagating errors. *Nature* (Vol. 323 pp. 533-536). doi: https://doi.org/10.1038/323533a0

Samuel A. L., (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development* (Vol. 3, I.3, pp. 210-229). doi: 10.1147/rd.33.0210

Sanders J. & Kandrot E. (2010). CUDA by Example An IntroductIon to GenerAl-PurPose GPu ProGrAmmInG. Addison-Wesley: Michigan

Saranya C., Priya J. G., Jayalakshmi P. & Pavithra E. H. (2021). Brain tumor identification using deep learning. *Materials Today: Proceedings*. doi: https://doi.org/10.1016/j.matpr.2020.11.555

Sarkar N. (1996). Mean square error matrix comparison of some estimators in linear regressions with multicollinearity. *Statistics & Probability Letters* (Vol. 30 I. 2 pp. 133-138). Doi: https://doi.org/10.1016/0167-7152(95)00211-1

Sel, A. (2020). Pandemi Sürecinde Altın Fiyatları ile Kripto Para İlişkisinin Makine Öğrenme Metotları ile İncelenmesi. *İstatistik ve Uygulamalı Bilimler Dergisi* (V.1 I.2 pp. 85-98). Erişim adresi: https://dergipark.org.tr/tr/pub/jsas/issue/58593/842821

Seyyarer E., Ayata F., Uçkan T. & Karci A. (2020). Derin Öğrenmede Kullanılan Optimizasyon Algoritmalarının Uygulanması ve Kıyaslanması. *Computer Science* (Vol. 5 I. 2 pp. 90-98). Erişim adresi: https://dergipark.org.tr/tr/pub/bbd/issue/57870/752132

Shambhu S., Koundal D., Das P. & Sharma C. (2022). Binary Classification of COVID-19 CT Images Using CNN: COVID Diagnosis Using CT. *International Journal of E-Health and Medical Communications* (Vol. 13 I. 2). doi: 10.4018/IJEHMC.20220701.oa4

Shorten C. & Khoshgoftaar T. M. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data (Vol. 6 I. 60). Doi: https://doi.org/10.1186/s40537-019-0197-0

Simonyan K. & Zisserman A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR 2015 bildiri olarak sunulmuştur. Erişim adresi: https://arxiv.org/abs/1409.1556

Sutskever I., Martens J., Dahl G. & Hinton G. (2013). On the importance of initialization and momentum in deep learning. *Proceedings of the 30th International Conference on Machine Learning* (Vol. 23 I. 3 pp. 1139 – 1147). Erişim adresi: https://proceedings.mlr.press/v28/sutskever13.html

Sun K., Zhang J., Zhang C & Hu J. (2017). Generalized extreme learning machine autoencoder and a new deep neural network. *Neurocomputing* (Vol. 30 pp. 374-381). doi: https://doi.org/10.1016/j.neucom.2016.12.027

Stanford (2021). 8 Şubat 2022 tarihinde https://stanford.edu/~shervine/l/tr/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks#classification-metrics adresinden erişildi.

Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V. & Rabinovich A. (2014). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) bildiri olarak sunulmuştur. doi: https://doi.org/10.1109/CVPR.2015.7298594

Taylor L. & Nitschke G. (2017). Improving Deep Learning with Generic Data Augmentation. IEEE Symposium Series on Computational Intelligence (pp. 1542-1547). doi: 10.1109/SSCI.2018.8628742.

Tian C., Xu Y. & Zuo W. (2020). Image denoising using deep CNN with batch renormalization. Neural Networks (Vol. 121 pp. 461-473). doi: https://doi.org/10.1016/j.neunet.2019.08.022

Tiken, C. (2015). Derin Öğrenme Uygulamaları. İstanbul Üniversitesi/Fen Bilimleri Enstitüsü, İstanbul. Erişim Adresi: https://tez.yok.gov.tr/UlusalTezMerkezi/TezGoster?key=WBc656i315e2eV6-EZV1oiKgH--6bj3j-oR2IeOljtS0NQWd50EavGwSVN0970T0

Uzair A., Beg M. O., Mujtaba H. & Majeed H. (2019). WEEC: Web Energy Efficient Computing: A machine learning approach, *Sustainable Computing: Informatics and Systems* (Vol. 22 pp.230-243). doi: https://doi.org/10.1016/j.suscom.2018.08.005

Uzun R. (2014). Çocukların Medyadan ve Medyada Korunması Çocuklara Yönelik Etik Davranış Kurallarının İncelenmesi. *Akdeniz Üniversitesi İletişim Fakültesi Dergisi* (Vol. 22 pp. 152-167). doi: https://doi.org/10.31123/akil.441955

Vani S. & Rao T. V. M. (2019). An Experimental Approach towards the Performance Assessment of Various Optimizers on Convolutional Neural Network. *2019 3rd International Conference on Trends in Electronics and Informatics* (ICOEI) (pp. 331-336). doi: 10.1109/ICOEI.2019.8862686.

Wang T. & Knap J. (2020). Stochastic gradient descent for semilinear elliptic equations with uncertainties. *Journal of Computational Physics* (Vol. 426). doi: https://doi.org/10.1016/j.jcp.2020.109945

Wang Y., Xu G., Liu X., Mao W., Si C., Pedrycz W. & Wand W. (2020). Identifying vulnerabilities of SSL/TLS certificate verification in Android apps with static and dynamic analysis. *Journal of Systems and Software* (Vol. 167). doi: https://doi.org/10.1016/j.jss.2020.110609

Wearesocial. (2021). 8 Şubat 2022 tarihinde https://wearesocial.com/digital-2021 adresinden erişildi.

Wei W., Haishan X., Alpers J., Rak M. & Hansen C. (2021). A deep learning approach for 2D ultrasound and 3D CT/MR image registration in liver tumor ablation. *Computer Methods and Programs in Biomedicine* (Vol. 206). doi: https://doi.org/10.1016/j.cmpb.2021.106117

Wikipedia (2021). 8 Şubat 2022 tarihinde https://en.wikipedia.org/wiki/List_of_alcoholic_drinks adresinden erişildi.

Wilson C., Roelofs R., Stern M., Srebro N. & Recht B. (2018). The Marginal Value of Adaptive Gradient Methods in Machine Learning. Erişim adresi: https://arxiv.org/abs/1705.08292v2

Wongwiwatchai N., Pongkham P. & Sripanidkulchai K. (2020). Detecting personally identifiable information transmission in android applications using light-weight static analysis. *Computers & Security* (Vol. 99). doi: https://doi.org/10.1016/j.cose.2020.102011

Xu B., He G. & Zhu H. (2021). ME-Box: A reliable method to detect malicious encrypted traffic. *Journal of Information Security and Applications* (Vol. 59). doi: https://doi.org/10.1016/j.jisa.2021.102823

Xu H., Wang J., Li H., Ouyang D. & Shao (2021). Unsupervised meta-learning for few-shot learning. *Pattern Recognition* (V. 116). doi: https://doi.org/10.1016/j.patcog.2021.107951

Yaraş E., Yetkin Özbük R. Y. & Çorlu P. (2018). Emmy Ödüllü Dizilerde Alkol ve Sigara Ürün Yerleştirme Uygulamalarının İçerik Analizi Yöntemi ile İncelenmesi. *Kastamonu Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi* (Vol. 20 I. 3). doi: 10.21180/iibfdkastam

Yildirim K., Gundogan Bozdag P., Talo M., Yildirim O., Karabatak M & Achaeya U. R. (2021). Deep learning model for automated kidney stone detection using coronal CT images. *Computers in Biology and Medicine* (Vol. 135). doi: https://doi.org/10.1016/j.compbiomed.2021.104569

Yu B., Guo Z., Asian S., Wang H. & Chen G. (2019). Flight delay prediction for commercial air transport: A deep learning approach. *Transportation Research Part E: Logistics and Transportation Review* (Vol. 125 pp. 203-221). doi: https://doi.org/10.1016/j.tre.2019.03.013

Yu C., Qi X., Ma H., He X., Wang C. & Zhao Y. (2020). LLR: Learning learning rates by LSTM for training neural networks. *Neurocomputing* (Vol. 394 pp. 41-50). doi: https://doi.org/10.1016/j.neucom.2020.01.106

Zao H., Gallo O., Frosio I. & Kautz J. (2018). Loss Functions for Image Restoration with Neural Networks. *IEEE Transactions on Computational Imaging* (Vol. 3 I. 1 pp. 47-57). doi: 10.1109/TCI.2016.2644865.

Zeiler M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. Erişim adresi: https://arxiv.org/abs/1212.5701

Zhang C. & Lu Y. (2021). Study On Artificial Intelligence: The State of The Art and Future Prospects. *Journal of Industrial Information Integration* (Vol. 23). doi: https://doi.org/10.1016/j.jii.2021.100224

Zhang N., Lei D. & Zhao J. F. (2018). An Improved Adagrad Gradient Descent Optimization Algorithm. Chinese Automation Congress (CAC) (pp. 2359-2362). doi: 10.1109/CAC.2018.8623271

Zhang J. (2019). Gradient Descent based Optimization Algorithms for Deep Learning Models Training. Erişim adresi: https://arxiv.org/abs/1903.03614v1

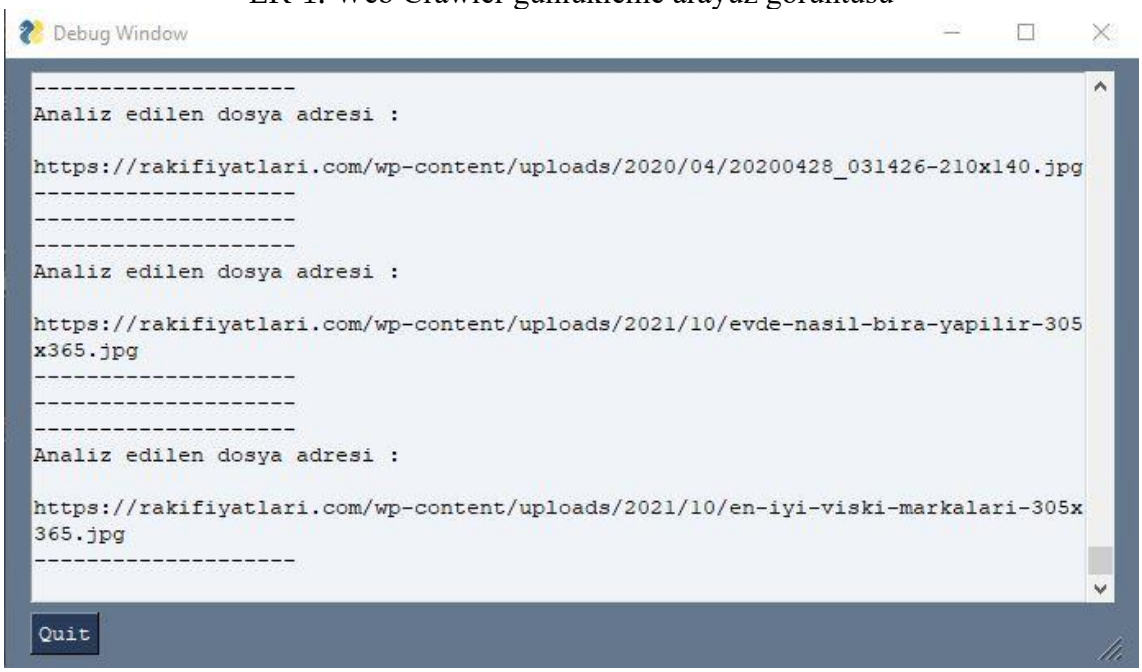
Zhao J., Qu T. & Xu F. (2020). A Deep Reinforcement Learning Approach for Autonomous Highway Driving. *International Federation of Automatic Control-PapersOnLine* (Vol. 53 I. 5 pp. 542-546). doi: https://doi.org/10.1016/j.ifacol.2021.04.142

Zhu X. & Li L. (2021). Flight time prediction for fuel loading decisions with a deep learning approach. *Transportation Research Part C: Emerging Technologies* (Vol. 128). doi: https://doi.org/10.1016/j.trc.2021.103179

Zuo, Y., Wu, Y., Min, G. & Cui, L. (2018). Learning-based network path planning for traffic engineering, *Future Generation Computer Systems* (Vol.92 pp. 59-67). doi: https://doi.org/10.1016/j.future.2018.09.043

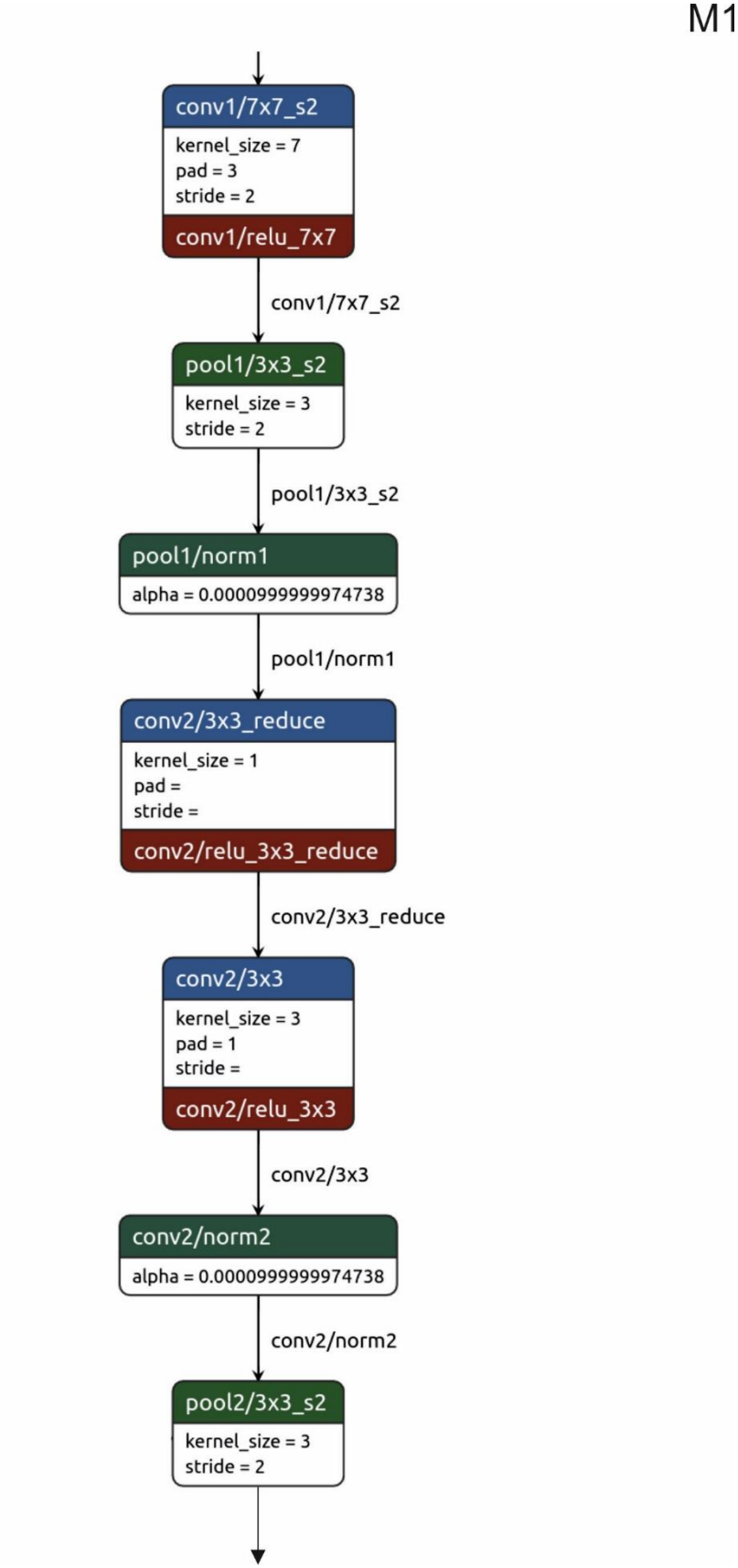
EKLER

EK-1. Web Crawler günlükleme arayüz görüntüsü



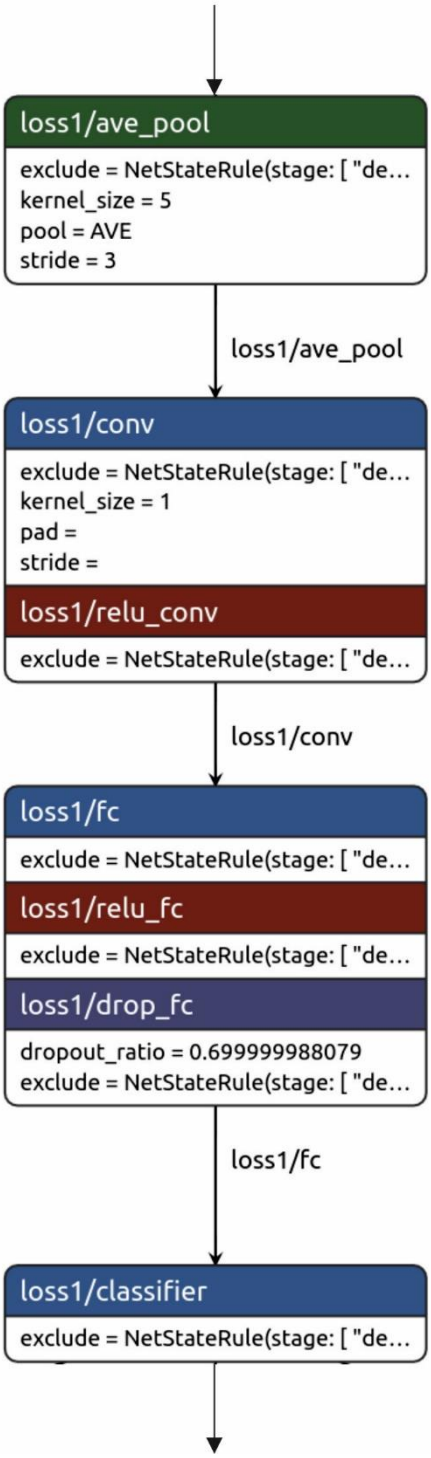


EK-2 M1 Modülü



EK-3 M2 Modülü

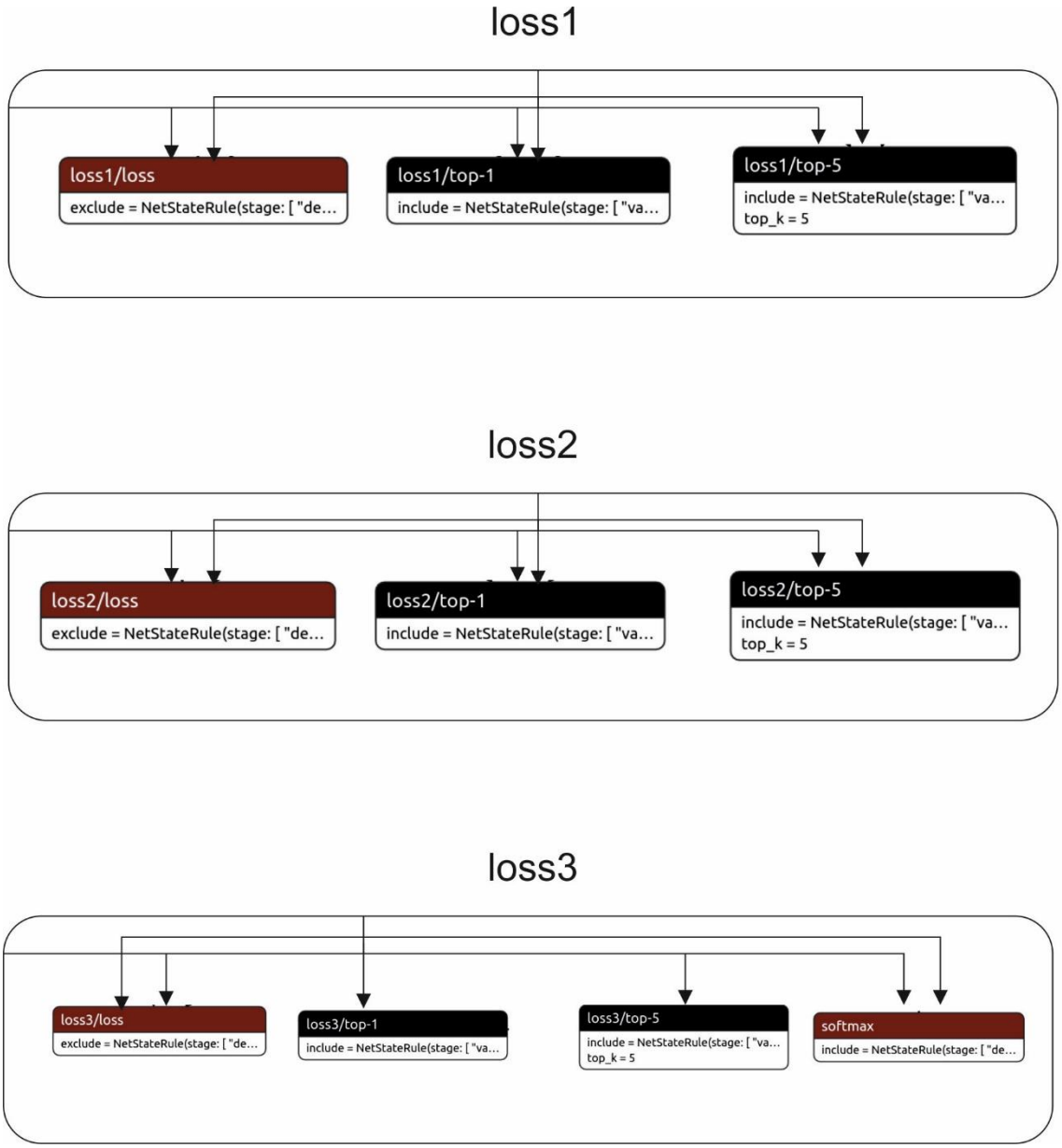
M2



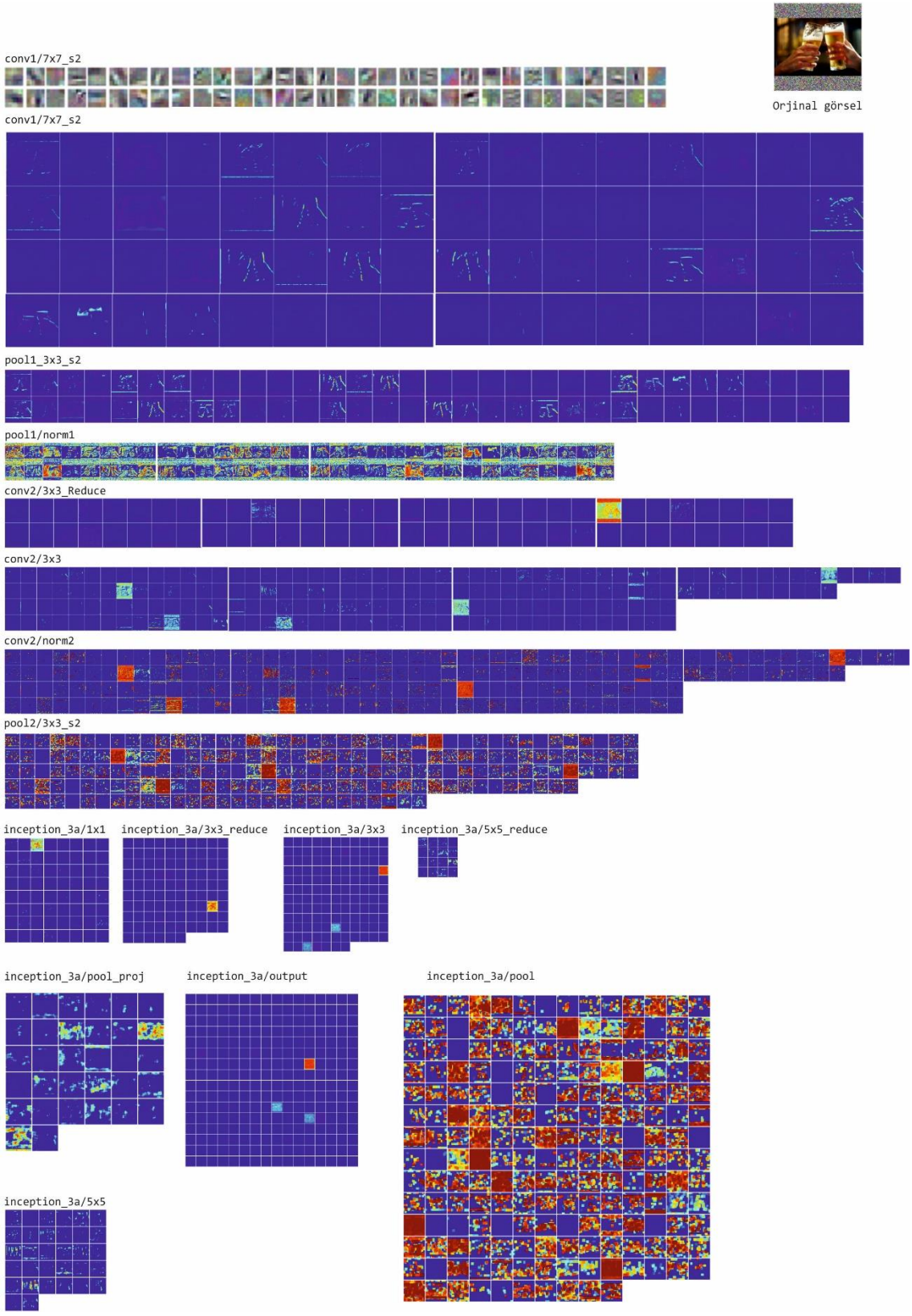
EK-4 M3 Modülü

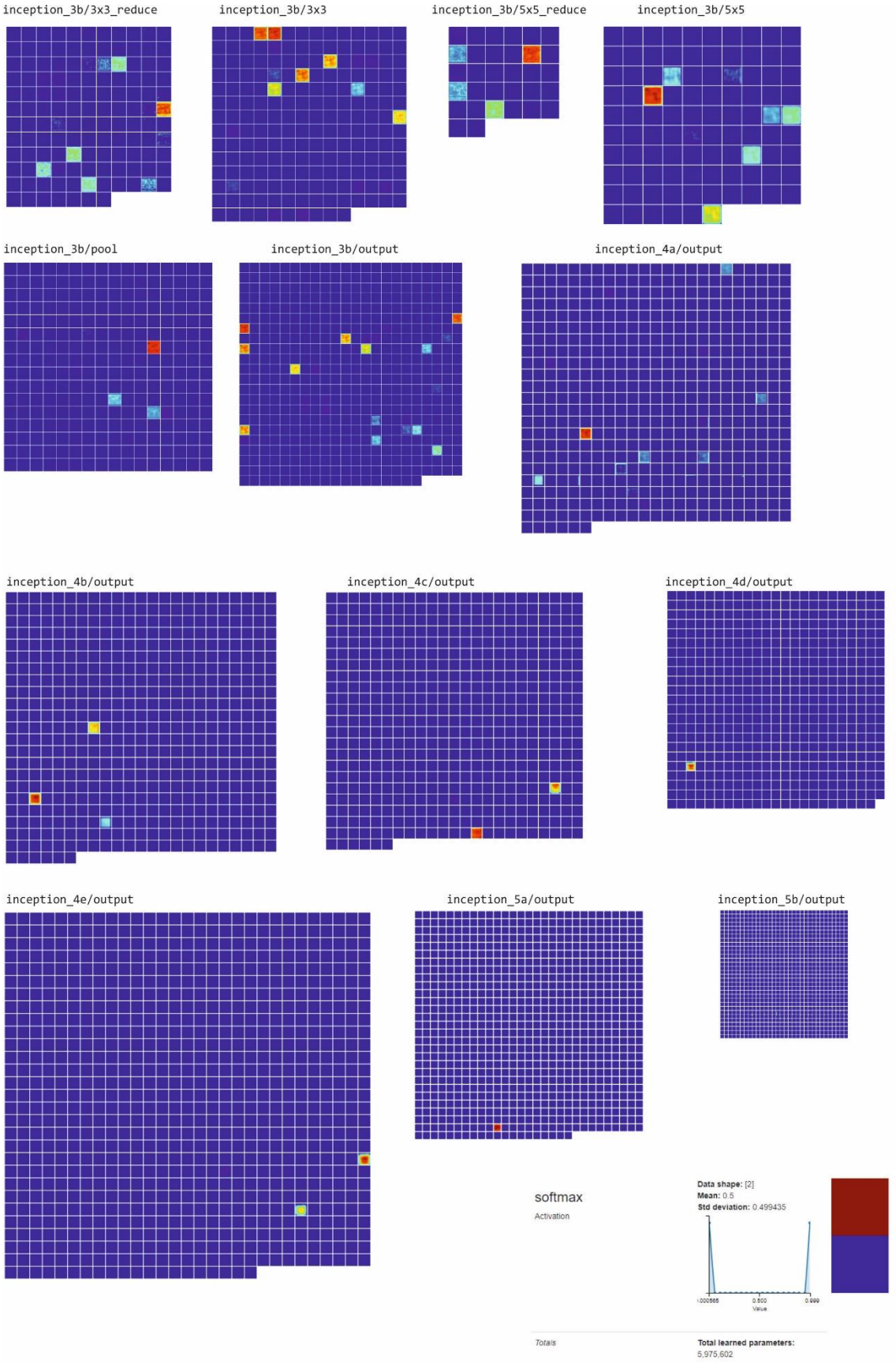


EK-5 Loss Çıkış Modülleri

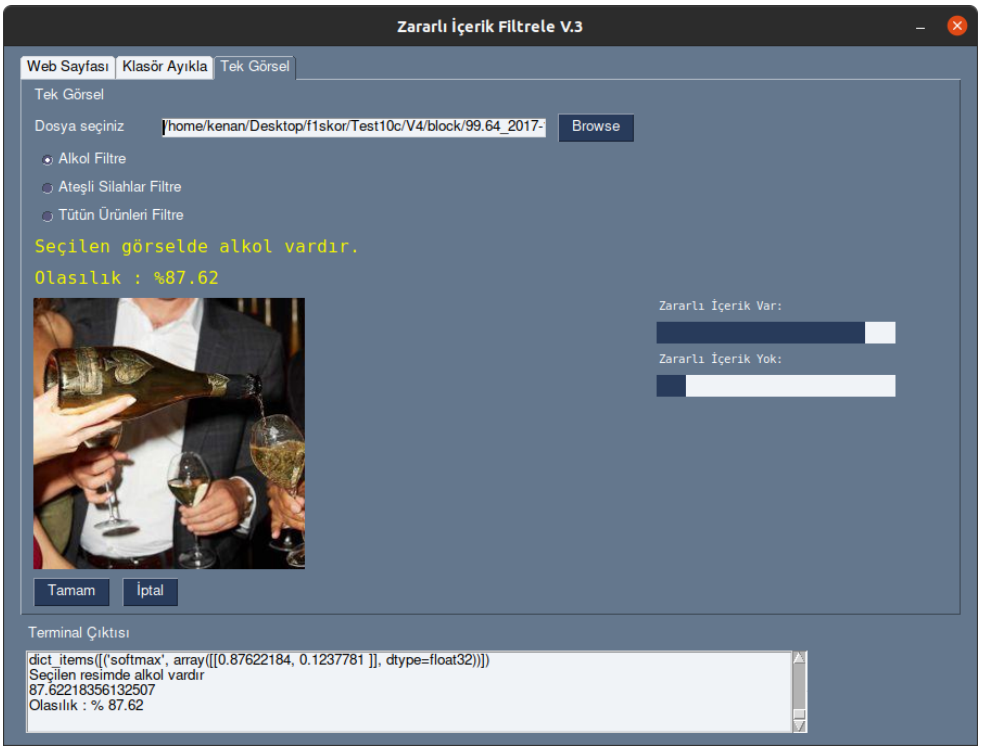


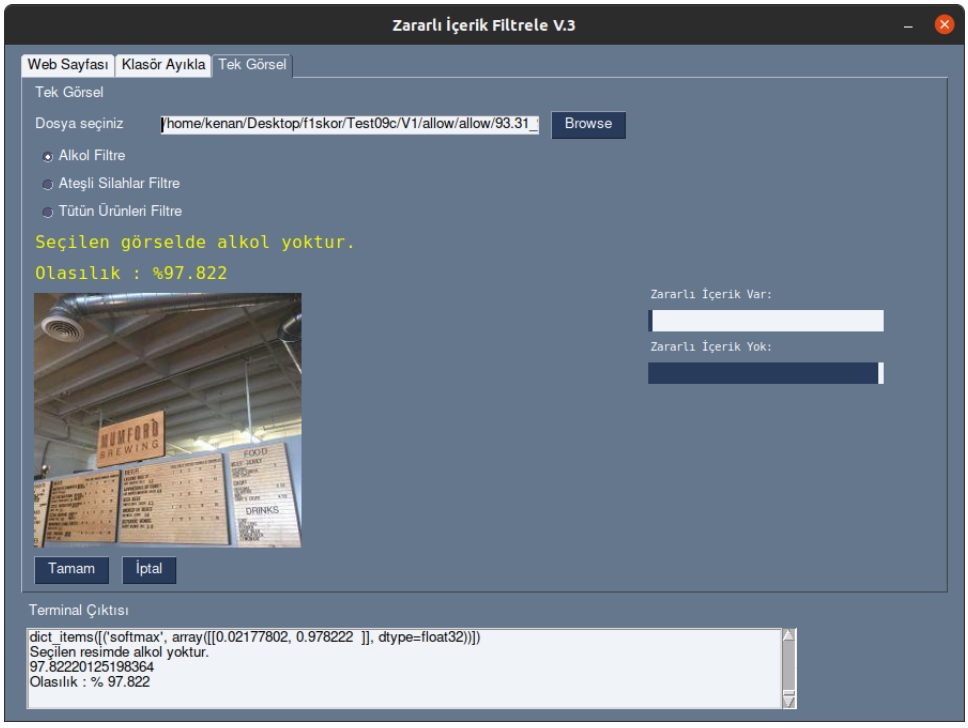
EK-6 Katmanların çıkış görselleri

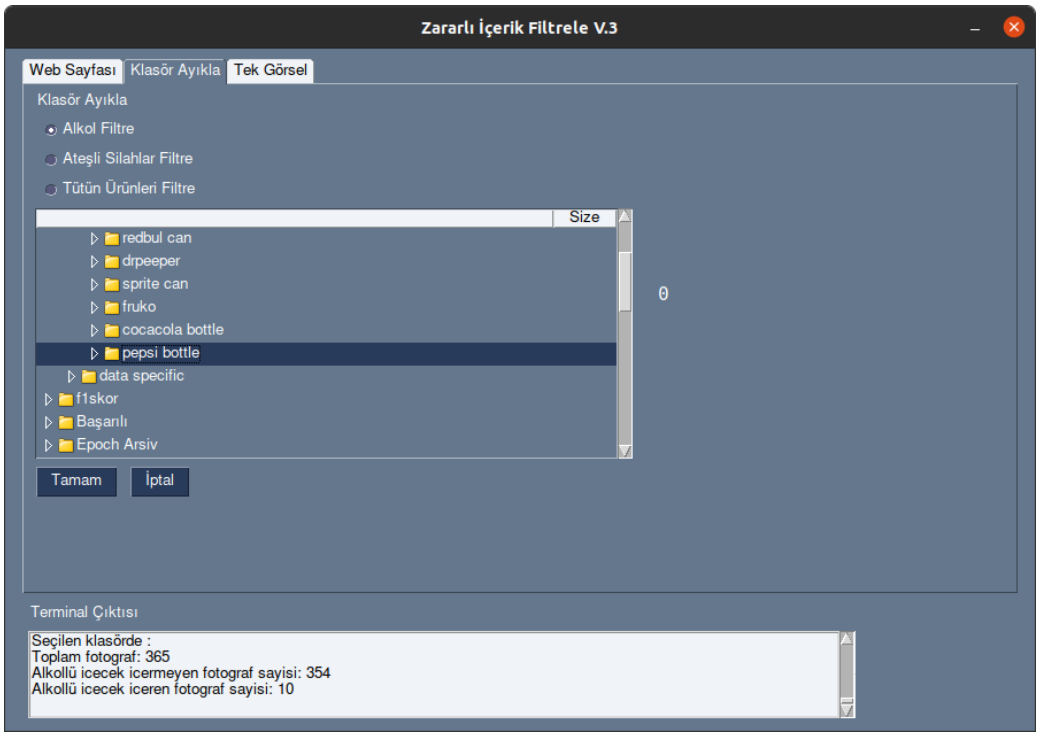


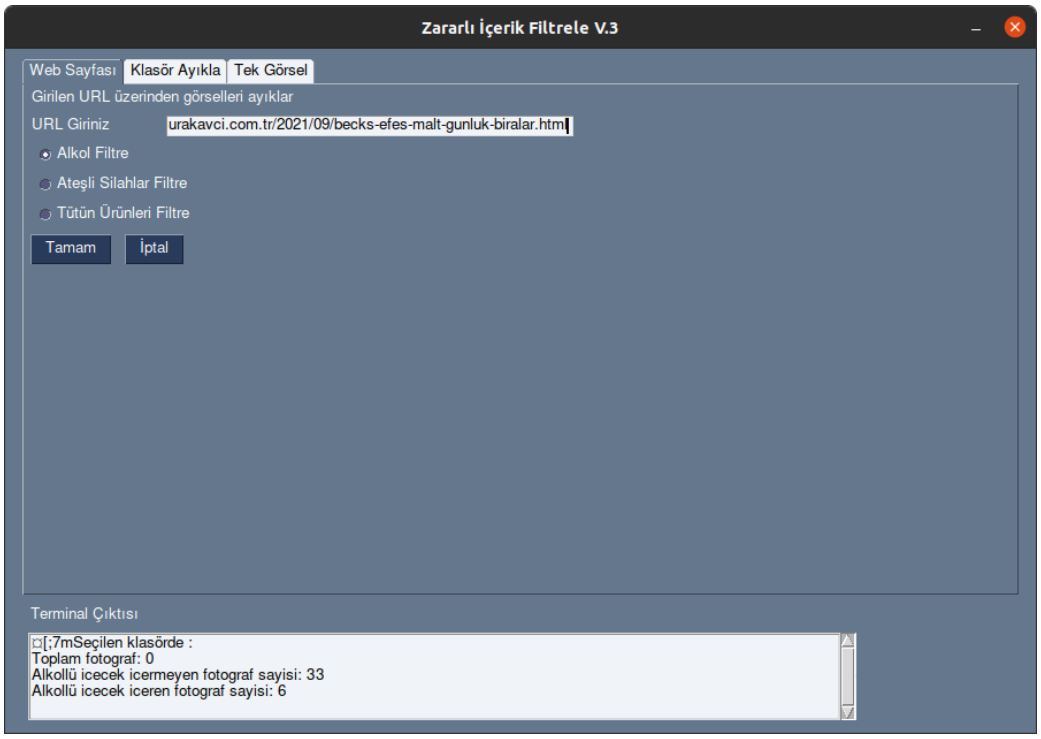

inception_3b/3x3_reduce
inception_3b/3x3
inception_3b/5x5_reduce
inception_3b/5x5
inception_3b/pool
inception_3b/output
inception_4a/output
inception_4b/output
inception_4c/output
inception_4d/output
inception_4e/output
inception_5a/output
inception_5b/output
softmax
Activation
Data shape: [2]
Mean: 0.5
Std deviation: 0.499435
Totals
Total learned parameters:
5,975,602

EK-7 Test arayüzü






Zararlı İçerik Filtrele V.3
Web Sayfası
Klasör Ayıkla
Tek Görsel
Klasör Ayıkla
Alkol Filtre
Ateşli Silahlar Filtre
Tütün Ürünleri Filtre
Size
redbul can
drpeeper
sprite can
fruko
cocacola bottle
pepsi bottle
data specific
f1skor
Başarılı
Epoch Arsiv
0
Tamam
İptal
Terminal Çıktısı
Seçilen klasörde :
Toplam fotograf: 365
Alkollü icecek icermeyen fotograf sayisi: 354
Alkollü icecek iceren fotograf sayisi: 10


Zararlı İçerik Filtrele V.3
Web Sayfası
Klasör Ayıkla
Tek Görsel
Girilen URL üzerinden görselleri ayıklar
URL Giriniz
urakavci.com.tr/2021/09/becks-efes-malt-gunluk-biralar.html
Alkol Filtre
Ateşli Silahlar Filtre
Tütün Ürünleri Filtre
Tamam
İptal
Terminal Çıktısı
□[;7mSeçilen klasörde :
Toplam fotograf: 0
Alkollü icecek icermeyen fotograf sayisi: 33
Alkollü icecek iceren fotograf sayisi: 6

ÖZGEÇMİŞ

Adı Soyadı	:	Kenan BAYSAL
Doğum Yeri ve Tarihi	:	
Medeni Durumu	:	
e-posta	:	

Eğitim Bilgileri

Lise	:	Tekirdağ Anadolu Teknik Lisesi, Elektronik Bölümü, Tekirdağ 2003
Lisans	:	Teknik Eğitim Fakültesi, Elektronik ve Bilgisayar Öğretmenliği / Sakarya Üniversitesi, Sakarya 2008
Yüksek Lisans	:	Fen Bilimleri Ens., Bilgisayar Müh., Trakya Üniversitesi, Edirne 2015

İş Deneyimi

2011 -	:	Tekirdağ Namık Kemal Üniversitesi, Öğretim Görevlisi