

**TRAFİK İŞARETLERİNİ TANIYAN BİR SÜRÜCÜ
GÜVENLİK DESTEK SİSTEMİ**

Mehmet ZAM



T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**TRAFİK İŞARETLERİNİ TANIYAN BİR SÜRÜCÜ GÜVENLİK DESTEK
SİSTEMİ**

Mehmet ZAM
Orcid No: 0000-0003-4802-4522

Doç. Dr. Ersen YILMAZ
Orcid No: 0000-0002-6620-655X
(Danışman)

YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2019

TEZ ONAYI

Mehmet ZAM tarafından hazırlanan “TRAFİK İŞARETLERİNİ TANIYAN BİR SÜRÜCÜ GÜVENLİK DESTEK SİSTEMİ” adlı tez çalışması aşağıdaki jüri tarafından oy birliği ile Bursa Uludağ Üniversitesi Fen Bilimleri Enstitüsü Telekomünikasyon Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Doç. Dr. Ersen YILMAZ
Orcid No: 0000-0002-6620-655X

Başkan : Doç. Dr. Ersen YILMAZ
Orcid No: 0000-0002-6620-655X
Bursa Uludağ Üniversitesi, Mühendislik
Fakültesi,
Elektronik Mühendisliği Anabilim Dalı

İmza

Üye : Dr. Öğr. Üyesi Cengiz TOĞAY
Orcid No: 0000-0001-5739-1784
Bursa Uludağ Üniversitesi, Mühendislik
Fakültesi,
Bilgisayar Mühendisliği Anabilim Dalı

İmza

Üye : Doç. Dr. Cemal HANILÇI
Orcid No: 0000-0002-9174-0367
Bursa Teknik Üniversitesi, Mühendislik ve Doğa
Bilimleri Fakültesi,
Elektronik Mühendisliği Anabilim Dalı

İmza

Yukarıdaki sonucu onaylarım

Prof. Dr. Hüseyin Akse EREN
Enstitü Müdürü
17.11/2019

U.Ü. Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

17/12/2019



Mehmet ZAM

ÖZET

Yüksek Lisans Tezi

TRAFİK İŞARETLERİNİ TANIYAN BİR SÜRÜCÜ GÜVENLİK DESTEK SİSTEMİ

Mehmet ZAM

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Devreler ve Sistemler Anabilim Dalı

Danışman: Doç. Dr. Ersen YILMAZ

Karayollarında sürücü hatalarının neden olduğu trafik kazalarını en aza indirmeyi amaçlayan sürüş güvenliği artırılmış araçlar tasarlamak günümüz otomotiv teknolojisinin en önemli konularından biridir. Bu amaçla araçlardaki aktif ve pasif güvenlik sistemlerinin sayısı gün geçtikçe artmaktadır.

Aktif güvenlik sistemlerinden birisi olan trafik işaretlerini tanıma sistemleri ön kameralar aracılığı ile trafik işaretlerini tanıyan ve sürücülerini bilgilendiren sistemlerdir ve yeni nesil araçlarda yerini almaya başlamıştır. Gerçek zamanlı çalışan bu sistemler özellikle karmaşık yol koşullarında istenilen performansa henüz ulaşamamışlardır ve önemli bir araştırma konusu olmaya devam etmektedirler.

Bu çalışmada trafik işaretlerinin tanınmasını amaçlayan evrimsel sinir ağları (ESA) temelli bir sürücü güvenlik destek sistemi (SGDS) önerilmiştir. Trafik işareti veri kümesi olarak Alman Trafik İşareti Tanıma Veri Kümesi (GTSRB) kullanılmıştır. Eğitim kümesindeki imgelerin sayısal orantısızlığı nedeniyle veri çoğaltma metotları kullanılarak eğitim kümesindeki imge sayıları artırılmıştır.

LeNet-5, AlexNet, GoogleNet ve ResNet ESA mimarileri kullanılarak oluşturulan ESA modellerinin katman yapısı ile en uygun eğitime parametrelerine deneysel olarak karar verilmiştir.

Yapılan deneyler sonucunda ResNet mimarisi kullanılarak önerilen SGDS'nin %98,10 sınıflandırma doğruluğuna sahip olduğu gösterilmiştir. Ayrıca, GTSRB kullanılarak eğitilen SGDS'nin Türkiye karayollarında çekilmiş bir grup trafik işareti imgesi üzerindeki başarımları sunulmuştur.

Anahtar Kelimeler: Sürücü güvenlik destek sistemi, Trafik işaret tanıma, Makine öğrenmesi, Derin öğrenme, Evrimsel sinir ağları
2019, viii + 60 sayfa.

ABSTRACT

MSc Thesis

A DRIVER SAFETY SUPPORT SYSTEM WHICH RECOGNIZE TRAFFIC SIGNS

Mehmet ZAM

Bursa Uludağ University
Graduate School of Natural and Applied Sciences
Department of Electronic Engineering

Supervisor: Assoc. Prof. Dr. Ersen YILMAZ

Designing vehicles with improved driving safety that aim to minimize the number of traffic accidents caused by driver errors on highways is one of the most important issues of today's automotive technology. For this purpose, the number of active and passive safety systems in vehicles is increasing day by day.

Traffic sign recognition systems, which are one of the active safety systems, are the systems that recognize the traffic signs through the front looking cameras and inform the drivers and they have started to take their place in the new generation vehicles. These systems, which operate in real time, have not yet achieved the desired performance, especially in complex road conditions, and remain an important research topic.

In this study, a driver safety support system (DSSS) based on convolutional neural networks (CNN), which aims to recognize traffic signs, was proposed. The German Traffic Sign Recognition Benchmarks (GTSRB) dataset was used as the traffic sign data set. Due to the numerical disproportion of the images in the training set, the number of images in the training set was increased by using data augmentation methods.

The layer structure and the most appropriate training parameters of the CNN models using LeNet-5, AlexNet, GoogleNet and ResNet CNN architectures were determined experimentally.

As a result of the experiments, it was shown that the proposed DSSS using ResNet architecture has 98.10% classification accuracy. Also, DSSS performance results were presented on a group of traffic signs which are taken in Turkey highways.

Key words: Driver safety support system, Traffic sign recognition, Machine learning, Deep learning, Convolutional neural networks
2019, viii + 60 pages.

TEŐEKKÜR

Bu tezi yapmamda ve yazmamda bana her daim desteęi olan aileme, yapmıő olduęu desteklerden dolayı danıőmanım Doę. Dr. Ersen Yılmaz'a ve beni teővik eden mesai arkadaőlarımaya teőekkürlerimi sunarım.

Mehmet ZAM
17.12.2019

İÇİNDEKİLER

	Sayfa
ÖZET.....	i
ABSTRACT	ii
TEŞEKKÜR.....	iii
SİMGELER ve KISALTMALAR DİZİNİ	v
ŞEKİLLER DİZİNİ.....	vi
ÇİZELGELER DİZİNİ	viii
1. GİRİŞ	1
2. KURAMSAL TEMELLER VE KAYNAK ARAŞTIRMASI.....	4
2.1. Yapay Sinir Ağları	6
2.2. Derin Öğrenme.....	12
3. MATERYAL VE YÖNTEM	39
3.1. Veri Kümesi	39
3.2. Donanımlar ve Yazılımlar.....	40
3.2.1. Google Colab	40
3.2.2. Nvidia Jetson TX2 Geliştirme Kartı	41
3.2.3. Kişisel Bilgisayar	42
3.2.4. Yazılımlar.....	42
3.3. Önerilen Modeller	43
4. BULGULAR VE TARTIŞMA	50
5. SONUÇ	57
KAYNAKLAR	58
ÖZGEÇMİŞ	60

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler	Açıklama
G_i	Girdi sinyalleri
A_i	Ağırlık değerleri
B	Bias değeri
λ	Öğrenme Oranı

Kısaltmalar	Açıklama
BTS	Belçika Trafik İşaretleri
BTSC	Belçika Trafik İşaretleri Tanıma
CNN	Convolutional Neural Network
CPU	Merkezi İşlemci Birimi
DSSS	Driver Safety Support System
ESA	Evrşimsel Sinir Ağı
GPU	Grafik İşlemci Birimi
GTSRB	Alman Trafik İşaretleri Tanıma
ILSVRC	ImageNet Geniş Ölçekli İmge Tanıma Yarışması
KGM	Karayolları Genel Müdürlüğü
KGTK	Karayolu Trafik Güvenlik Kurulu
KNN	K En Yakın Komşu
PCA	Temel Bileşen Çözümlemesi
RELU	Düzleştirici Doğrusal Birim
RGB	Kırmızı Yeşil Mavi
SELU	Ölçekli Üstel Doğrusal Birim
SGDS	Sürücü Güvenlik Destek Sistemleri
SVM	Destek Matris Makinesi
TPU	Tensör İşlemci Birimi
YSA	Yapay Sinir Ağları

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 2.1. Bir biyolojik sinir hücresinin yapısı.....	7
Şekil 2.2. Yapay sinir ağı hücresi	8
Şekil 2.3. Yapay sinir ağı modeli	9
Şekil 2.4. Tek katmanlı yapay sinir ağı.....	10
Şekil 2.5. Çok katmanlı yapay sinir ağı	11
Şekil 2.6. Temel ESA mimarisi	14
Şekil 2.7. Evrişim işlemi	15
Şekil 2.8. Örnek evrişim işlemi.....	16
Şekil 2.9. Filtre örnekleri	16
Şekil 2.10. Dikey kenar filtresi ile evrişim uygulaması	17
Şekil 2.11. Çıktı boyutu hesaplama örneği,kaydırma miktarı=1	18
Şekil 2.12. Evrişim işlemi sonucu.....	18
Şekil 2.13. RGB imgesi ile evrişim işlemi.....	20
Şekil 2.14. Birinci filtre ile evrişim işlemi	21
Şekil 2.15. İkinci filtre ile evrişim işlemi.....	22
Şekil 2.16. RGB imgesi üzerinde 2 filtre ile gerçekleştirilen evrişim işlemi sonucu	22
Şekil 2.17 Evrişim katmanlardan oluşan ESA modeli	23
Şekil 2.18. Ortaklama işlem örneği.....	25
Şekil 2.19. Düzleştirme ve tam bağlı katman yapısı	26
Şekil 2.20. ESA model yapısı	26
Şekil 2.21. Yığın ve Mini-Yığın karşılaştırması	29
Şekil 2.22. Öğrenme oranı – Kayıp ilişkisi	32
Şekil 2.23. Öğrenme oranı – Kayıp değişim ilişkisi	32
Şekil 2.24. LeNet-5 model mimarisi	34
Şekil 2.25. AlexNet model mimarisi.....	34
Şekil 2.26. VGG-16 model mimarisi	35
Şekil 2.27. Temel inception yapısı	35
Şekil 2.28. Boyutsal küçülme uygulanmış inception modülü.....	36
Şekil 2.29. Resnet Özdeşlik bloğu.	37
Şekil 2.30. Resnet Evrişim bloğu.....	37
Şekil 2.31. Resnet-34 model mimarisi	38
Şekil 3.1. Trafik işaretleri sembolleri.....	39
Şekil 3.2. GTSRB veri kümesi trafik işareti örnekleri	40
Şekil 3.3. Nvidia Tesla K80 GPU modeli	41
Şekil 3.4. Google TPU modeli	41
Şekil 3.5. Nvidia Jetson Tx2 işlemcisi ve geliştirme kartı.....	42
Şekil 3.6. PC sistem bilgileri.....	42
Şekil 3.7. İki katmanlı ESA katman yapısı	44
Şekil 3.8. AlexNet temelli ESA mimarisi	46
Şekil 3.9. GoogleNet temelli ESA mimarisi	48
Şekil 3.10. Resnet temelli ESA mimarisi.....	49
Şekil 4.1. GTSRB veri kümesi için sınıf bazında imge örneği sayıları	50
Şekil 4.2. Arttırılmış imge örnekleri	51
Şekil 4.3. Türkiye karayollarından toplanan trafik işaret örnekleri	54
Şekil 4.4. Kasıslı yol işareti ve sistem tahmini	54
Şekil 4.5. GTSRB veri kümesi ve Türkiye'deki STOP işareti.....	54

Şekil 4.6 Türkiye'deki okul geçidi işareti ve GTSRB tahmini	55
Şekil 4.7. Türkiye'deki hız sınırı işareti (82km/sa) ve model tahminleri	55
Şekil 4.8. Türkiye'deki hız sınırı (45km/sa) ve model tahminleri	55

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 2.1. Aktivasyon fonksiyonları.....	8
Çizelge 2.2. ILSVRC yarışması sonuçları	38
Çizelge 3.1. GTSRB veri kümesi detayları	40
Çizelge 3.2. Derin öğrenme kütüphaneleri ve özellikleri.....	43
Çizelge 3.3. İki katmanlı ESA modeli katman yapısı	45
Çizelge 3.4. AlexNet temelli ESA modeli katman yapısı	47
Çizelge 3.5. GoogleNet temelli ESA modeli katman yapısı	49
Çizelge 4.1. Orijinal ve arttırılmış veri kümesi ile başarımları	51
Çizelge 4.2. Yığın boyutuna bağlı başarımları değişimi	51
Çizelge 4.3. Öğrenme oranına bağlı başarımları değişimi.....	52
Çizelge 4.4. Epok'a bağlı başarımları değişimi.....	52
Çizelge 4.5. ESA modellerinin donanımlara göre öğrenme süreleri	52
Çizelge 4.6. Önerilen sistemin GTSRB için başarımları	53
Çizelge 4.7. GTSRB kullanan çalışmalarda elde edilen başarımları	53
Çizelge 4.8. Türkiye karayollarından toplanan trafik işaretleri için başarımları sonucu	56

1. GİRİŞ

Son yıllardaki teknolojik gelişmeler ile birlikte birçok yenilik hayatımıza girmiştir. Bu gelişmeler otomotiv endüstrisine de yansımıştır. Yeni nesil otomobiller bilgi ve eğlence sistemleri ile sürücü ve yolculara eğlenceli ve konforlu bir yolculuk sağlamaktadır.

Bununla birlikte devletlerin belirlediği yasal zorunluluklar ve müşterilerin sunulan güvenlik opsiyonlarına ilgisi sayesinde sürüş güvenliğini arttıracak teknolojilerin geliştirilmesine ve araçlardaki yerlerini almasına zemin hazırlamıştır.

Sürüş güvenliğinin artırılmasını hedefleyen Sürücü Güvenlik Destek Sistemleri (SGDS) geliştirilmesine olan ilgi son yıllarda teknolojik gelişmelere paralel olarak artmıştır. (Kılıkçier ve Yılmaz 2018).

Yolcu güvenliğini artırmak için iki ana yaklaşım bulunmaktadır. İlk yaklaşım, pasif güvenlik sistemleri olarak adlandırılan ve sürücü ve yolcuların kaza esnasında alacağı zararları hafifletmeye çalışan sistemlerdir. Bu amaçla, hava yastığı sayısı artırılmış ve şasi tasarımları buna göre yapılmıştır. İkinci yaklaşım ise aktif güvenlik sistemleri olarak adlandırılır ve birtakım ek önlemler ile kaza olasılığının azaltılması hedeflenmektedir. Şehir içi fren desteği, Şerit takip sistemi, Kör nokta uyarı sistemi gibi farklı donanım ve yazılımlar ile sürücülerini bilgilendiren aktif güvenlik sistemleri bulunmaktadır.

SGDS araç üzerinde yer alan algılayıcılar ve kameralar aracılığı ile elde ettikleri bilgileri kullanarak sürücüyü bilgilendiren sistemlerdir. Gerçek zamanlı çalışan bu sistemler özellikle karmaşık yol koşullarında henüz istenilen performansa ulaşamamışlardır ve önemli bir araştırma konusu olmaya devam etmektedirler (Sürücü ve Doğan 2018, Canyon 2018, Hossain ve diğ. 2015).

Karayolu trafiğinde yaşanan kazaların büyük bir kısmı sürücü hatalarından kaynaklanmaktadır.

Türkiye Cumhuriyeti Karayolları Genel Müdürlüğü (KGM) raporuna göre trafik kazalarının %89,9'u sürücü hataları nedeniyle meydana gelmiştir. Bu kazaların %40,8'i, sürücülerin araç hızını yol, hava ve trafiğin şartlarına göre ayarlayamamasından dolayı olduğu gözlemlenmiştir. %13'ü ise kavşaklarda geçiş önceliğine uyulmamasından ve %8'i ise manevraları düzenleyen genel şartlara uyulmamasından dolayı kaynaklanmıştır (Anonim 2017).

2011 yılında Karayolu Trafik Güvenlik Kurulu (KTGK)'nın düzenlediği ankete Ankara ve çevresinden 1478 sürücü katılmış, sonuçlarda 39 farklı trafik işaretinden 12 tanesinin doğru bilindiği, 10 tanesinin ise bilinmediği gözlemlenmiştir (Kırmızıoğlu ve diğ. 2011).

2017 yılında Denizli'de gerçekleştirilen bir diğer ankette ise 27 adet trafik işareti 500 sürücüye sorulmuş, sorulan 27 adet işaretin %40'ında sürücülerin yeterli düzeyde bilgi ve farkındalığının olmadığı gözlemlenmiştir (Murat ve diğ. 2017).

Karayolu trafik işaretleri, ana ve yardımcı işaretler olmak üzere iki kategoriye ayrılmıştır. Ana işaretler yasaklama işaretlerini de içeren 6 alt-kategoriye ayrılmıştır. Yasaklama işaretleri güvenli bir sürüş için büyük öneme sahip olan hız sınırı ve sağ-sol dönüş yasakları gibi farklı 43 sınıfa ayrılmıştır ve trafik işaretlerinin tanınması üzerine mevcut araştırmaların odak noktasında yer almaktadır (Canyong, 2018).

Aktif sürücü güvenlik destek sistemleri içerisinde yer alan trafik işaretlerini tanıma sistemi günümüzde sürüş güvenliğini arttırmayı amaçlayan araştırma konularından biri olmaya devam etmektedir. Geliştirilmekte olan bazı sistemler henüz sürücülerini sadece belirlenmiş önemli trafik işaretleri hakkında uyarılmaktadır. Bu sistemler gelecekte diğer aktif güvenlik sistemleri ile birleşerek bazı durumlarda aracın kontrolünü ele geçirecek ve kazaların oluşmamasında etkili rol oynayacaklardır.

Trafik işaretlerini tanıma sistemi sürücülerin iş yükünü azaltacak ve trafikte ortaya çıkacak yorgunluk ve uykusuzluk gibi insan kaynaklı faktörlerden etkilenmeyecektir.

Bu tez kapsamında trafik işaretlerini tanıyan bir sürücü güvenlik destek sistemi geliştirilecektir.

Derin öğrenme modellerinden evrimsel sinir ağlarına ait farklı mimariler ile bu probleme çözüm aranacak ve önerilen sistemin başarımlı incelemesi Alman trafik işaretlerinden oluşan Alman Trafik İşareti Tanıma Veri Kümesi (GTSTRB) kullanılarak yapılacaktır.

Önerilen bu yeni yöntem ile trafik işaretlerinin tanınmasına katkıda bulunulması hedeflenmiştir.

2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI

Son yıllarda nesne tespiti ve tanıma problemlerinde derin öğrenme teknikleri kullanılmaya başlanmış ve yapılan çalışmalar derin öğrenme tekniklerinin mevcut tekniklerden daha iyi sonuçlar elde ettiğini göstermiştir.

Grafik işlemci birimlerinin (GPU) son yıllardaki gelişimine paralel olarak vektör ve matrisler üzerindeki matematiksel işlemlerin hızı artmıştır. Bu sayede derin öğrenme modellerindeki eğitilme süreleri azalmış ve paralel işlem yapma kabiliyeti artmıştır.

ESA öğrenme kabiliyeti ve başarımı sayesinde trafik işareti tanıma problemlerinde sıklıkla kullanılan yöntemlerden biri olmuştur (Hussain ve diğ. 2018, Stallkamp ve diğ. 2012, Sürücü ve Doğan 2018, Alvaro ve diğ. 2018).

Trafik işaretlerinin tespiti ve tanınması ile ilgili olarak literatürde önerilmiş çok sayıda çalışma bulunmaktadır (Öm., Wang ve diğ. 2009, Kiran ve diğ. 2009, Ramadhan ve Ergen 2017, Sermanet ve LeCun 2011, Sürücü ve Doğan 2018, Alvaro ve diğ. 2018, Yin ve diğ. 2017).

Wang ve diğ. (2009) tarafından yapılan çalışmada hız limiti trafik işaretlerinin tanınması amacıyla Adaboost ve Hough dönüşümüne dayanan iki aşamalı bir yöntem önerilmiştir. Önerilen yöntemin karanlık ve kötü hava koşullarında da gürbüz bir başarıma sahip olduğu gösterilmiştir.

Renk temelli bölütleme tekniği ile destek vektör makinelerini kullanarak trafik işaretlerinin tespiti ve tanınmasını amaçlayan bir yaklaşım Kiran ve diğ. (2009)'da sunulmuştur.

Ramadhan ve Ergen (2017) tarafından yapılan çalışmada Blob analizi yöntemi ile trafik işaretlerinin tespiti ve tanınması gerçekleştirilmiştir. Video imgesinden alınan çerçeveler ikili imgelere dönüştürüldükten sonra blob çıkarımı yapılmıştır. Yapılan deneyler

sonucunda önerilen yöntemin trafik işaretlerini tanıma hızının yüksek olduğu ve gerçek zamanlı uygulamalarda kullanılabileceği belirtilmiştir.

ESA ağları ile trafik işaretlerinin tanınmasını amaçlayan başka bir çalışma Alvaro ve diğ. (2018)'de sunulmuştur. Alman (GTSRB) ve Belçika (BTSC) trafik işareti veri kümelerini kullanan bu çalışmada ESA'nın her iki veri kümesi için de yüksek başarımlı sonuçlarına sahip olduğu gösterilmiştir.

ESA mimarisi kullanılarak trafik işaretlerinin tanınmasını amaçlayan bir yaklaşım Sermanet ve LeCun (2011)'de sunulmuştur. Çalışmada GTSRB veri kümesi kullanılmış ve imgeler YUV uzayına dönüştürülerek kullanılmıştır.

Xie ve diğ. (2016) tarafından yapılan çalışmada iki aşamalı seri bağlı ESA yapısı ile trafik işaretlerinin tanınması gerçekleştirilmiştir. Trafik işareti veri kümesi olarak GTSRB kullanılmış ve seri bağlı yapının modelin ayrıştırma kapasitesini arttırdığı gösterilmiştir.

Sürücü ve Doğan (2018) tarafından yapılan çalışmada iki katmanlı hiyerarşik ESA ile trafik işaretlerinin tanınması gerçekleştirilmiştir. GTSRB veri kümesi kullanılan çalışmada öncelikle trafik işaretleri yapısal benzerlik indisi kullanılarak gruplara ayrılmış ve daha sonra sınıflandırılmışlardır.

Canyong (2018) tarafından yapılan çalışmada ESA mimarisi olarak VGG-16 kullanılarak trafik işaretlerinin tespiti ve tanınması gerçekleştirilmiştir. Çalışmada GTRSB ile BTSC trafik işareti veri kümeleri kullanılmıştır.

Resnet mimarisi ile oluşturulmuş ESA modeli ile trafik işaretlerinin tanınmasını amaçlayan başka bir çalışma Yin ve diğ. (2017)'de sunulmuştur. GTSRB kullanılan bu çalışmada ağ içinde ağ yapısı oluşturularak artık bağlantıyı birleştiren 10 katmanlı bir ESA modeli önerilmiştir. İlk 7 katman evrişim ve ortaklama katmanlarından son 3 katman ise tam bağlı katmanlardan oluşmaktadır. Aşırı uyumu azaltmak için eğitim veri setinde veri çoğaltma işlemi uygulanmış ve ölçekli üstel doğrusal birim (SELU) aktivasyon

fonksiyonu kullanılmıştır. Resnet modeli ile GTSRB test kümesi için de yüksek başarımlı sonuçları elde edilmiştir.

Literatürde Alman veri kümesi GTSRB kullanılarak trafik işareti sınıflandırma çalışmalarına ait başarımlı sonuçları sonraki bölümlerde gösterilmiştir.

Bu tez kapsamında birbirinden farklı evrimsel sinir ağı mimarileri ile trafik işaretlerinin tanınması probleminin çözümü getirilecektir. Şimdi makine öğrenmesi ve derin öğrenme mimarileri detaylı olarak inceleyelim.

Makine Öğrenmesi, bilgisayarların insanlar gibi öğrenmelerini ve davranmalarını ve özerk bir şekilde, gözlemler ve gerçek dünya etkileşimleri şeklinde veri ve bilgi besleyerek öğrenmelerini zaman içinde geliştirmelerini sağlayan bir bilimdir. Makine öğrenmesi sayesinde kod yazmak yerine büyük veri setleri kullanılarak algoritmalar oluşturulabilir ve daha sonra kullanılan büyük veriye dayanarak öğrenme sağlanabilir.

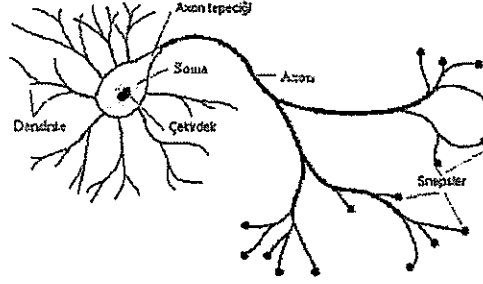
Makine öğrenmesi eğitilmiş ve eğitilmiş öğrenme olarak ikiye ayrılır. Eğitilmiş öğrenme, girdilere ve çıktılara sahip etiketli veri setlerini kullanarak öğrenmeyi gerçekleştirir. Hava durumu tahminleri eğitilmiş öğrenmeye örnek verilebilir. Eğitilmiş öğrenme ise sadece girdileri kullanarak öğrenmeyi gerçekleştirir. Eğitilmiş öğrenme ile imge tanıma, konuşmadan metne ve doğal dil işleme gibi daha karmaşık konularda milyonlarca eğitim veri kümesi örneği kullanılarak öğrenme gerçekleştirilir.

2.1.Yapay Sinir Ağları

Yapay sinir ağları makine öğrenmesinde kullanılan ana araçlardan biridir. İnsan beyninin öğrenme mekanizmasından ilham alınarak oluşturulmuş yapay sinir ağları, yeni bilgi üretme ve keşfedebilme gibi özellikleri otomatik olarak gerçekleştirmeyi sağlayan sinir sisteminin basit bir matematiksel modeli çıkarılarak oluşturulmuştur.

İnsan sinir sisteminin çalışma mekanizmasını inceleyelim. Duyu organları aracılığı ile alınan sinyaller birbirine bağlı sinir hücreleri sayesinde beyine taşınır ve beyinde alınan

kararlar yine aynı yol ile vücudun ilgili organlarına iletilir. Şekil 2.1’de gösterilen bir sinir hücresi (nöron), hücre gövdesi (çekirdek), dendrit ve aksonlardan oluşmaktadır. Çevreden alınan uyarılar elektrik sinyalleri halinde dendritler tarafından hücre çekirdeğinde toplanır ve hücre çekirdeği bu sinyalleri derleyerek tek bir sinyal haline getirir ve akson uçları sayesinde kendisine bağlanan bir diğer sinir hücresine bu sinyali aktarır.



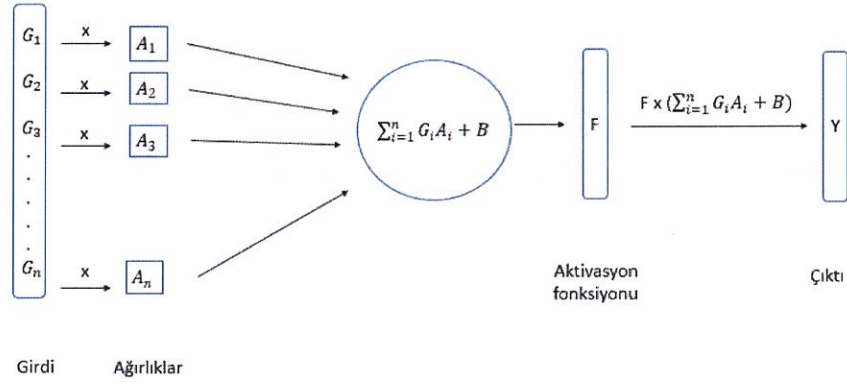
Şekil 2.1. Bir biyolojik sinir hücresinin yapısı

Sinir hücresinin çalışma yapısının matematiksel modeli 1957 yılında Frank Rosenblatt tarafından çıkarılmıştır.

Temel bir yapay sinir ağında G_i girdi sinyallerini, A_i her girdinin ağırlığını, n girdi sayısını, B bias değerini ve F aktivasyon fonksiyonunu temsil etsin. Bu durumda Y çıktısı 2.1 eşitliği ile elde edilir. Bir yapay sinir hücresi Şekil 2.2’de gösterilmiştir.

$$\text{Çıktı} = Y = F\left(\sum_{i=1}^n G_i A_i + B\right) \quad (2.1)$$

Girdi ile ağırlık çarpımının sonucunun sıfır üretmesi durumunda düğümün etkisini kaybetmemesi için bias değeri eklenmelidir. Girdiler ile ağırlıklarının çarpım toplamları aktivasyon fonksiyonu sayesinde anlamlandırılır.



Şekil 2.2. Yapay sinir ağı hücresi

Sistemin doğrusallığını değiştirebilmek ve girdi sinyalleri ile ağırlık değerlerinin çarpımlarından elde edilen toplam değer anlamlı bir hale getirilmesi için aktivasyon fonksiyonu kullanılmaktadır. Literatürde kullanılan aktivasyon fonksiyonları Çizelge 2.1’de verilmiştir. Her problem için kullanılacak aktivasyon fonksiyonları değişiklik gösterecektir.

Çizelge 2.1. Aktivasyon fonksiyonları

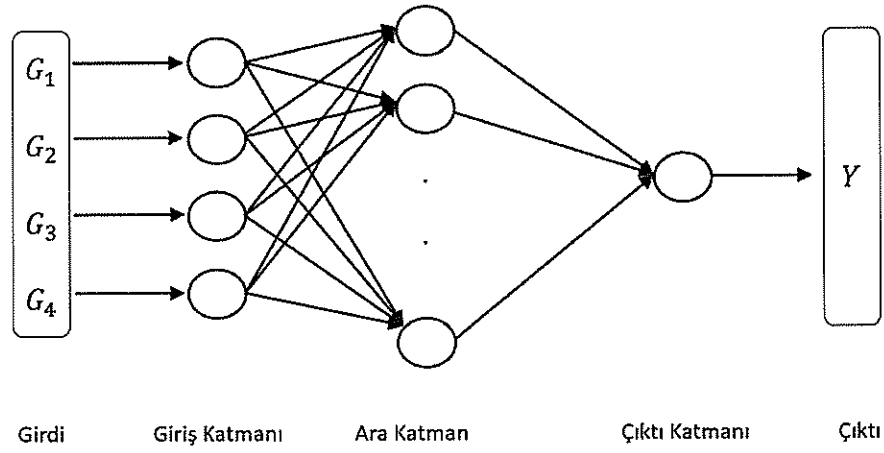
Aktivasyon Fonksiyonu	Denklemler	Aralık
Doğrusal Fonksiyon	$f(x) = \begin{cases} x < 0 \text{ için } 0 \\ x \geq 0 \text{ için } 1 \end{cases}$	(0,1)
Basamak Fonksiyonu	$f(x) = x$	$(-\infty, \infty)$
Sigmoid Fonksiyonu	$f(x) = \frac{1}{1 + e^{-x}}$	(0,1)
Hiperbolik Tanjant Fonksiyonu	$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	(-1,1)
ReLU	$f(x) = \begin{cases} x < 0 \text{ için } 0 \\ x \geq 0 \text{ için } x \end{cases}$	(0, ∞)

Bir yapay sinir hücresinin nasıl çalıştığını bir örnek ile gösterelim. Girdi sinyalleri 0.5, 0.6, 0.2 ve 0.7 olduğunu ve her sinyale karşılık gelen A ağırlık değerleri 0.2, 0.6, 0.2, -0.1 olsun. Bias değeri 0 olarak tanımlandığında y çıktısı eşitlik 2.1’de verilen eşitliğe göre $(0.5x-0.2) + (0.6x0.6) + (0.2x0.2) + (0.7x-0.1) = -0.1+0.36+0.4-0.07 = 0.23$ olarak elde

edilmiştir. Aktivasyon fonksiyonu sigmoid olarak seçilip girdilerin ağırlık parametreleri ile çarpımlarının toplamları bu fonksiyondan geçirildiğinde bir sinir ağının çıktısı belirlenmiş olur.

YSA Mimarisi

Yapay sinir ağı Şekil 2.3'te görüldüğü gibi üç katmandan oluşmaktadır. Giriş katmanı, ara katman ve çıkış katmanı.



Şekil 2.3. Yapay sinir ağı modeli

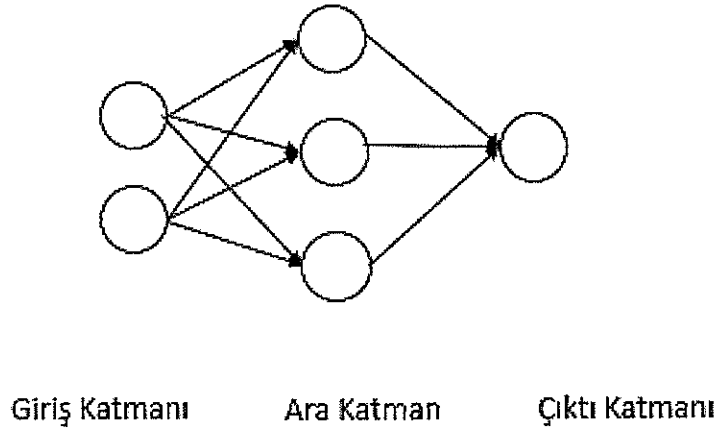
Giriş katmanı ile sinir ağlarına verilerin girişleri sağlanır. Bu katmanda herhangi bir matematiksel işlem gerçekleşmemektedir.

Ara katmanda ağırlıklar yer almaktadır. Her girdi bu katmanda kendisine karşılık gelen ağırlıklar ile çarpılmaktadır. Her girdinin ağırlıkları ile çarpılıp toplanması sonrası aktivasyon fonksiyonu uygulanır ve çıkış elde edilir.

Ağırlık değerleri model tasarlanırken rastgele belirlenmektedir. Ağırlık değerleri karar mekanizmasındaki önceliklendirmeyi belirler. Ağırlık değeri ne kadar büyük ise alınacak

kararı o derecede etkilemektedir. Her girdi sinyalinin bir ağırlık değeri vardır. Ağırlık değeri ne kadar küçük ise girdi değeri büyük olsa bile çarpım, sonucu etkilemeyecektir. Sistemin alacağı karar, girdi ve ağırlığının çarpımının büyüklüğü ile doğru orantılıdır. Ağırlık değerinin negatif olması girdinin sisteme ters etki yapacağı anlamını taşır.

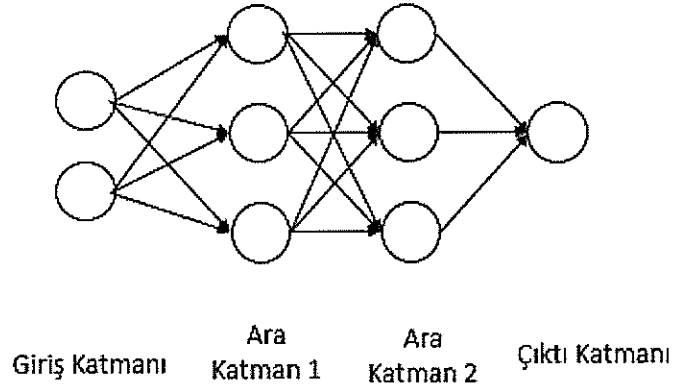
Yapay sinir ağları tek bir ara katmana sahip olacağı gibi birden çok ara katmana da sahip olabilir. Widrow ve Hoff (1960) tarafından ilk kez çok katmanlı bir yapay sinir ağı kullanılmışlardır.



Şekil 2.4. Tek katmanlı yapay sinir ağı

Tek ara katmanlı YSA modelinde toplam 4 adet sinir hücresi (nöron) bulunmaktadır. Giriş değerleri nöron olarak kabul edilmemektedir. Şekil 2.4'de görüldüğü üzere ara katman 3 adet, çıkış katmanı ise 1 adet sinir hücresine sahiptir. Bu modelin öğrenmesi gereken parametre sayısı ise katmanların arasındaki ilişkiye bağlıdır.

Giriş ve ara katman arasında $2 \times 3 = 6$ ağırlık ve ara katman ile çıkış katmanı arasında $3 \times 1 = 3$ ağırlık olmak üzere toplam 9 ağırlık parametresi yer almaktadır. Ara katman ve çıkış katmanında eklenecek bias değerlerini de eklediğimizde $3 + 1 = 4$, toplam $9 + 4 = 13$ adet parametrenin öğrenme sürecinde en optimum değerleri alması gerekmektedir.



Şekil 2.5. Çok katmanlı yapay sinir ağı

Şekil 2.5’de görülen iki ara katmanlı YSA modelinde ise ara katmanlardaki nöron sayısı $3+3=6$ dir. Çıkış katmanı 1 olarak görülmektedir. Bu durumda toplam nöron sayısı 7 dur. Giriş katmanı ile birinci ara katman arasında $2 \times 3 = 6$ adet ağırlık parametresi vardır. Birinci katman ile ikinci katman arasında $3 \times 3 = 9$ adet ağırlık parametresi hesaplanabilir. İkinci ara katman ile çıkış katmanı arasında ise $3 \times 1 = 3$ adet ağırlık parametresi olduğu görülmektedir. Bu durumda toplam $6+9+3=18$ adet ağırlık parametresi bulunur. Ara katmanlar ve çıkış katmanı arasındaki bias değerleri ise 3,3 ve 1 olacaktır ve toplam bias değer $3+3+1=7$ olmak üzere bu modelde öğrenilmesi gereken parametre sayısı $18+7 = 25$ olarak hesaplanır.

İleri beslemeli ve geri beslemeli YSA modelleri mevcuttur. İleri beslemeli yapay sinir ağlarında akış tek yönlüdür. Giriş katmanı ile alınan sinyaller ara katmandaki ağırlık değerleri ile çarpıldıktan sonra aktivasyon fonksiyonuna tabi tutulurlar ve çıkış katmanı elde edilir.

Geri beslemeli sinir ağlarında akış çift yönlüdür. Bu sayede yapay sinir ağının ilk oluştuğu anda rastgele belirlenen ağırlıklar güncellenir ve ileri beslemeli akışta elde edilen çıktı ile beklenen değer arasındaki fark yani sistemin kaybı azaltılabilir ve ağın başarımlı performansı artırılabilir.

Yapay sinir ađlarının bařarım performansını etkileyecek olan yapay sinir ađının katman sayısı, ilk ađırlık parametreleri, girdinin çeřitliliđi ve boyutu, aktivasyon fonksiyonu ve geri yayılım algoritması gibi parametrelerin belirlenmesi evriřimsel sinir ađları bölümünde daha detaylı anlatılacaktır.

2.2.Derin Öğrenme

Derin öğrenme, algoritmaların büyük veriden öğrendiđi bir makine öğrenmesi sınıfıdır. Algoritmalar eđitmenli veya eđitmensiz öğrenme yöntemiyle eđitilmektedir.

Derin öğrenme sayesinde makine öğrenimi algoritmalarında kullanılan parametrelerin ilk deđerlerinin belirlenmesi problemi giderilmekte ve en iyi parametrelerin keřfedilmesini sađlamaktadır.

Bu tez kapsamında ele alınan trafik iřaretlerinin sınıflandırılması problemi derin öğrenme algoritmalarından evriřimsel sinir ađları mimarileri ile ele alınmıřtır.

Evriřimsel Sinir Ađları (ESA)

Evriřimsel sinir ađları (ESA), hayvanların görsel korteksin alıřmasından esinlenerek tasarlanmıřtır. Görsel korteksin amacı göz retinasına gelen bilgiyi alıp yorumlamaktır.

ESA imge sınıflandırma ve nesne tanıma problemlerinde en bařarılı sonuçları elde etmesinden dolayı bilgisayarlı görü alanında çođunlukla tercih edilmektedir.

ESA 3 boyutlu girdiler üzerinde hızlı matematiksel iřlemler yapma becerisine sahiptir. Ađ yapısında bulunan katmanlar arasındaki matematiksel iřlemler gerekleřtirildikten sonra ıktı katmanında sınıflandırma problemine özüm getirecek bir vektör üretebilmektedir.

İmge üzerinde yapılan çalışmalarda yapay sinir ağlarındaki katmanlar arasında gerçekleştirilen matematiksel işlemlerin fazlalığı nedeniyle aşırı uyumlama problemleri ile karşı karşıya kalınmaktadır. Bu problemi aşmak için ise daha güçlü işlemciler ihtiyacı vardır.

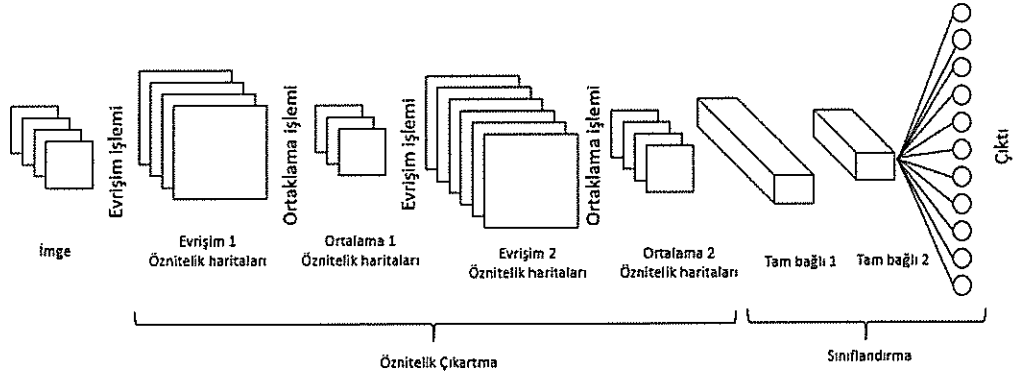
ESA yapısında ise parametre paylaşımı yapılmaktadır. Bu sayede parametre sayısında azalma meydana gelmekte ve yapılan matematiksel işlemler yapay sinir ağlarına göre azalmaktadır. Bu nedenle ESA çoklu katman yapısı ve yüksek işlemciler ihtiyacı duymaması sayesinde yapay sinir ağları yerine tercih edilmektedir.

Bunu bir örnek ile açıklayalım. $32 \times 32 \times 3$ boyutunda bir imgenin standart bir YSA ile sınıflandırılmasını düşünelim. Bu imgenin YSA girişine uygulanması için vektör haline getirilmesi gerekmektedir. Bu nedenle düzleştirme işlemi sonucunda $32 \times 32 \times 3 = 3072$ satırlık bir vektör haline gelen imge ardından YSA girişine alınır. YSA'nın bu giriş vektörünü alması için 3072 adet ağırlığa sahip olması gerekmektedir. Daha sonra bu vektörün dönüşüm (çarpım) işlemleri sonrasında gizli katmanlara aktarımı sağlanır. Gizli katmanlar arasında da dönüşüm işlemi gerçekleştirildikten sonra sınıflandırma katmanına gelerek sonuç üretilir.

Görüldüğü gibi küçük boyutlarda bir imge karşısında YSA'nın oldukça fazla parametre çarpımı yapması gerekmektedir. $[7680 \times 4320 \times 3]$ boyutunda 8K çözünürlükteki bir imge için $7680 \times 4320 \times 3 = 99.532.800$ parametreye sahip bir vektörün işlenmesi için YSA'nın çok daha güçlü bir işlemciye ihtiyacı olacaktır. Aksi takdirde aşırı uyumlanmaya sebebiyet verecektir.

ESA Mimarisi

ESA mimarisi genel olarak evrişim, ortaklama ve tam bağlı katman olmak üzere 3 ana katmandan oluşmaktadır. Her katman kendisinden önce gelen katmanın çıktısını girdi olarak kullanmaktadır. Temel ESA mimarisi Şekil 2.6'da gösterilmiştir.



Şekil 2.6. Temel ESA mimarisi

Bu işlemler sırasında boyut azalmıştır ancak derinlik artmıştır. Tam bağı katman çıkışında ise istenilen sınıflandırma sayısı kadar hacme sahip bir vektör elde edilir. Daha sonra sınıflandırma işlemlerine tabi tutularak her bir sınıf için skor bilgisi üretilmektedir.

ESA modelinde yer alan bu katmanların detayları ve her katmana ait çıktuların boyutlarının belirlenmesindeki matematiksel işlemler sonraki bölümlerde anlatılacaktır.

Evrşim katmanı:

Evrşim katmanı, ESA mimarisinin ana yapı taşıdır ve en kritik öneme sahip katmandır. Ağdaki matematiksel işlemlerin en yoğun olduğu kısımdır.

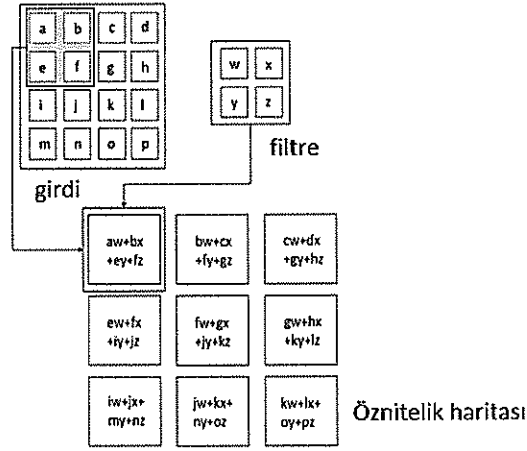
Bu katmandaki parametreler filtrelerden oluşmaktadır. Filtrenin boyutları genellikle girdinin boyutundan küçüktür ancak girdinin derinliği (kanal sayısı) ile aynı olmak zorundadır. Ağdaki geri beslemeli işlemler sayesinde bu katmanda yer alan filtreler yapay sinir ağlarındaki ağırlıklar gibi güncellenmektedir ve öğrenme bu sayede gerçekleşmektedir.

Evrşim işlemi öncesinde filtrenin x ve y eksenine simetriği alınır ve girdi bu katmandaki filtreler ile evrşim işlemine tabi tutulur. Genellikle uygulamalarda filtrenin simetriği

alınmamaktadır. Bu durumda yapılan işleme korelasyon adı verilmektedir. Bu tez kapsamında verilen örneklerdeki evrişim işlemlerinde işlemlerin kolaylığı nedeniyle korelasyon uygulanacaktır.

Evrişim işlemi boyunca filtre, Şekil 2.7’de görüldüğü gibi girdinin genişliği ve yüksekliği boyunca kaydırılır ve her adımda çakışan değerler çarpılarak toplanır. Girdinin boyutları $[n \times n]$ ve filtrenin boyutları $[f \times f]$ olarak kabul edersek, evrişim işlemi sonucu oluşacak çıktının boyutları eşitlik 2.2 ile elde edilir.

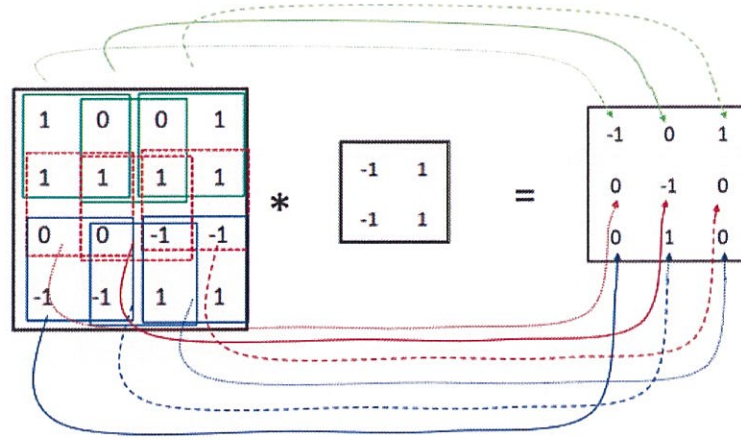
$$\text{Çıktı boyutları} = [n - f + 1, n - f + 1] \quad (2.2)$$



Şekil 2.7. Evrişim işlemi

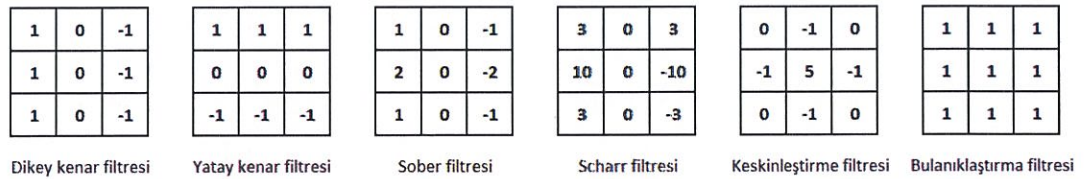
Örnek bir evrişim işlemi ile bu işlemlerin nasıl yapıldığını gösterelim. $[4 \times 4]$ boyutundaki girdi ile $[2 \times 2]$ boyutundaki filtrenin evrişim sonucu oluşan çıktının boyutları $[4-2+1 \times 4-2+1] = [3 \times 3]$ olacaktır.

Filtreyi girdinin sol üst bölgesi ile örtüştürüp noktasal çarpım işlemi uygulandığında $[1 \times (-1) + 0 \times 0 + 0 \times 1 + 1 \times 0 + 1 \times (-1) + 1 \times 0 + 0 \times 0 + 0 \times 1 + (-1) \times 0] = -1$ değeri elde edilir. Daha sonra filtre bir piksel yatayda ve dikeyde hareket ettirilerek Şekil 2.8’deki giriş imgesinin sağ alt köşesine getirene kadar her olası piksel konumu için evrişim işlemi tekrar edilir.



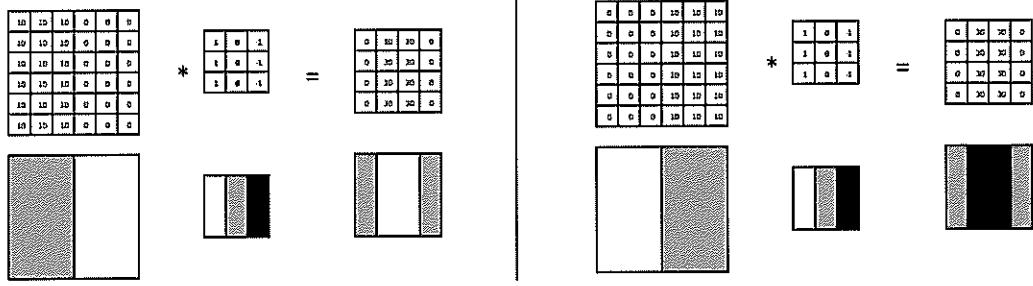
Şekil 2.8. Örnek evrişim işlemi

ESA mimarisinde yer alan filtrelerin öğrenilebilir olduğundan bahsetmiştik. Evrişim işlemi ile girdi olarak alınan imgeye ait farklı özellikler öğrenilebilir. Bu sayede imgelerin istenilen sınıflandırma özellikleri çıkarılabilir. Farklı filtreler uygulanarak imgeye ait kenar özellikleri, renk özellikleri, karanlıktan aydınlığa veya aydınlıktan karanlığa geçiş gibi farklı özellikler tespit edilebilir. Şekil 2.9’da imgeye ait farklı özellikleri tespit etmeyi sağlayacak filtre örnekleri verilmiştir.



Şekil 2.9. Filtre örnekleri

Örnek bir evrişim işlemi ile filtreleri kullanarak gri ölçekli imgelerdeki kenar özelliklerinin tespit edilmesini gösterelim. 6x6 boyutundaki bir imgenin aslı ve y eksenine göre simetrisi ile dikey kenar filtresinin evrişim işleminin gerçekleştirilmesi sonucunda dikey kenarların tespiti kolaylıkla sağlanmaktadır. Şekil 2.10’da karanlıktan aydınlığa veya aydınlıktan karanlığa geçiş gibi özelliklerin bu sayede tespit edildiği gösterilmiştir.



Şekil 2.10. Dikey kenar filtresi ile evrişim uygulaması

Evrişim işlemi sonunda oluşacak çıktının boyutlarını filtre derinliği, filtrenin girdi üzerindeki kaydırma miktarı ve sıfır ekleme parametreleri değiştirmektedir.

Evrişim işleminde filtrenin derinlik değerinden bahsetmiştik. Derinlik parametresi kullanılacak filtre sayısına bağlı değişmektedir. 3 boyutlu RGB bir imgeye uygulanacak en az 3 boyutlu bir filtre ile imgenin dikey, yatay ve renkleri tespit edilecektir.

Evrişim işlemi sonunda oluşacak çıktıyı belirleyen kaydırma miktarı ve sıfır ekleme gibi parametrelerin tanımlanması gerekmektedir. Bu sayede evrişim işlemi sonucunda istenen çıktı boyutları elde edilebilir.

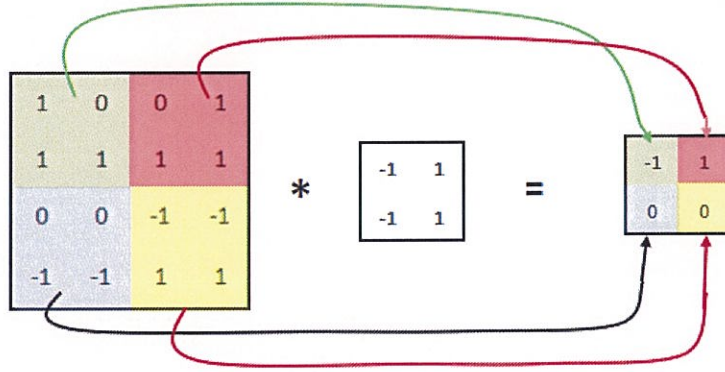
Şimdiye kadar yapılan evrişim işlemlerinde kaydırma miktarı 1 olarak alınmış ve imge üzerinde filtrenin 1 piksel kaydırılması sonucunda noktasal çarpım işlemi uygulanmıştır. Evrişimsel işlem çıktısının boyutlarının küçültülmesi hedefleniyor ise kaydırma miktarı arttırılabilir.

Çıktı boyutunu etkileyen bir diğer parametre piksel ekleme işlemidir. Bu işlem genellikle girdi boyutu ile çıktı boyutunun aynı olması istendiğinde yapılır. Girdinin kenarlık çevresi sıfırlar ile doldurularak işlem gerçekleştirilir.

Girdi boyutu $[n \times n]$, filtre boyutu $[f \times f]$, piksel 'p' ve kaydırma miktarı 's' kabul edildiğinde evrişim işlemi sonucu oluşan çıktının boyutları $[(n+2p-f)/s+1] \times [(n+2p-f)/s+1]$ olacaktır.

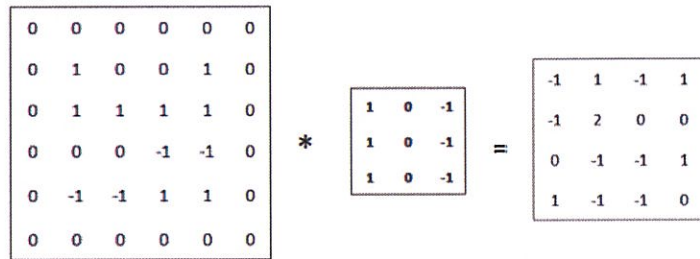
Aynı örnekte kaydırma miktarı 2 seçildiğinde ve piksel ekleme olmadığında evrişim işlemi sonucu boyutlar $[(4-2)/2+1] \times [(4-2)/2+1] = [2 \times 2]$ olacaktır.

Şekil 2.11’de görüldüğü gibi evrişim işlemi sonucu oluşan çıktının boyutları 3×3 boyutundan 2×2 boyutuna küçülmüştür.



Şekil 2.11. Çıktı boyutu hesaplama örneği, kaydırma miktarı=1

Piksel ekleme 1 ve kaydırma miktarı 1 seçildiğinde evrişim sonucu Şekil 2.12’de gösterildiği gibi $[4+2-2/1+1] \times [4+2-2/1+1] = [4 \times 4]$ olacaktır. Bu sayede girdinin boyutları ile çıktının boyutları aynı olacaktır.



Şekil 2.12. Evrişim işlemi sonucu

Parametreler seçilirken dikkatli olmak gereklidir. Uygun olmayan parametreler seçildiğinde çıktı boyutlarının değerleri tamsayı olmayabilir.

ESA'da 3 boyutlu imgeler üzerinde evrişim işlemi hızlı bir biçimde gerçekleştirilebilir.

Evrişim katmanına ait parametrelerin boyutlarını incelediğimizde, evrişimsel katman l olarak tanımlandığında,

$f^l =$ filtre boyutu, $p^{[l]} =$ piksel ekleme, $s^{[l]} =$ kaydırma sayısı
 $n_d^{[l]} =$ filtre sayısı,

$$\text{Girdi} = x = n_y^{[l-1]} \times n_g^{[l-1]} \times n_d^{[l-1]} \quad (2.3)$$

$$f = f^{[l]} \times f^{[l]} \times n_d^{[l-1]} \quad (2.4)$$

Buna göre evrişim işlemi sonucu oluşacak y değerinin boyutları Denklem 2.5'e göre hesaplanır.

$$n_{y,g}^{[l]} = \frac{n_{y,g}^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \quad (2.5)$$

Evrişim işlemi sonucu oluşacak y değerinin boyutları Denklem 2.6'ya göre hesaplanır.

$$y^l = n_y^{[l]} \times n_g^{[l]} \times n_d^{[l]} \quad (2.6)$$

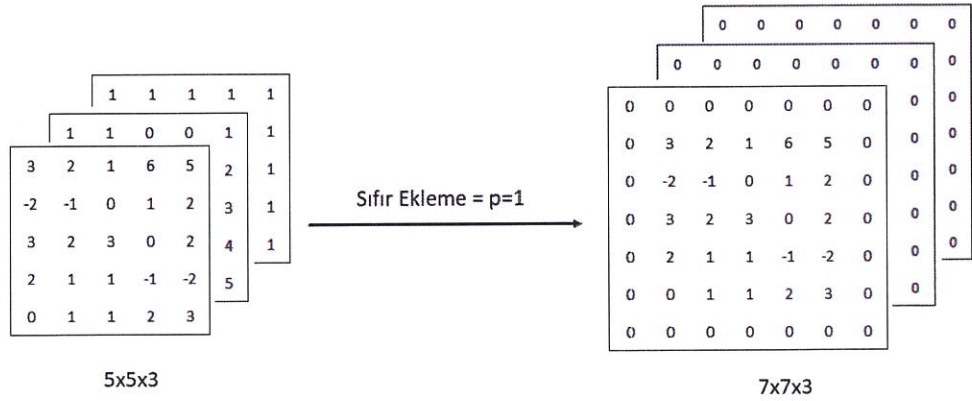
Daha önce de belirtildiği gibi girdinin derinliği ile kullanılacak filtrenin derinliği aynı olmak zorundadır.

Evrişim katmanındaki giriş ($x \in R^{Y \times G \times D}$) ile bir filtre kümesinin ($f \in R^{Y \times G \times D \times K}$) evrişim işlemi Denklem 2.7'de verildiği gibi gerçekleştirilir.

$$y_{i_y, j_y, k} = \sum_{i=1}^{Y_k} \sum_{j=1}^{G_k} \sum_{d=1}^D f_{ijd} \times x_{i_y+i, j_y+j, d, k} + b_k \quad (2.7)$$

Şimdiye kadar evrişim katmanında yapılan işlemler iki boyutlu imgeler üzerinde gerçekleştirildi. Evrişim katmanlarının işleyişinin daha net anlaşılması için üç boyutlu bir imge üzerinde somut bir örnek verilecektir.

5x5x3 boyutundaki bir RGB imgesine $n_d = 2$ adet 3x3 boyutunda bir filtre uygulanacaktır. Kaydırma sayısı $s=2$ ve piksel ekleme $p=1$ olacaktır. Şekil 2.13'de sıfır ekleme sonucu oluşan girdinin matrisi görülmektedir.



Şekil 2.13. RGB imgesi ile evrişim işlemi

Çıkış boyutları Denklem 2.5'teki eşitliğe göre hesaplanmıştır. Buna göre çıktı boyutları 3x3x2 olacaktır.

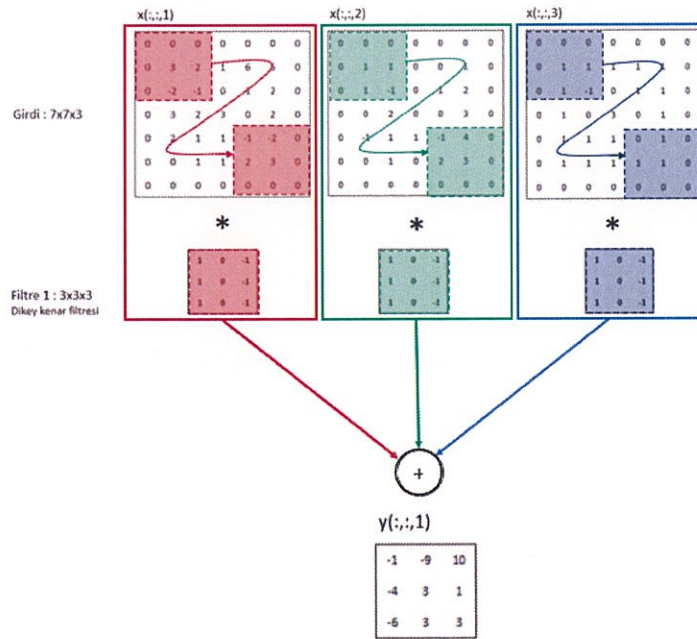
$$n_{y,g} = \frac{5 + (2 \times 1) - 3}{2} + 1 = 3$$

Şekil 2.14'de girdi ile birinci filtrenin evrişim işlemi gerçekleştirilmiştir. Girdiye ait her katman üzerinde aynı kanal sayısına sahip filtreler ile noktasal çarpım işlemi uygulanır. Daha sonra her katmandaki değerler toplanarak çıktının ilgili alanına bu değer yazılır. İlk

adımında filtreler katmanlar üzerinde kaydırıldığında $x(:, :, 1)$ için uygulanan $f(:, :, 1)$ ile $1x0 + 0x0 + (-1)x0 + 1x0 + 0x3 + (-1)x2 + 1x0 + 0x(-2) + (-1)x(-1) = -1$ elde edilir. $x(:, :, 2)$ ile $f(:, :, 2)$ evrişim işlemi gerçekleştirdiğinde $1x0 + 0x0 + (-1)x0 + 1x0 + 0x0 + (-1)x1 + 0x1 + (-1)x(-1) = 1$ elde edilir. $x(:, :, 3)$ ile $f(:, :, 3)$ evrişim işlemi gerçekleştirdiğinde $1x0 + 0x0 + (-1)x0 + 1x0 + 0x1 + (-1)x1 + 0 + 0x1 + (-1)x(-1) = 1$ elde edilir. 3 değer toplandığında $-1 + 1 + 1 = 1$ elde edilir. Bu değer $y(:, :, 1)$ matrisinin ilk değeridir.

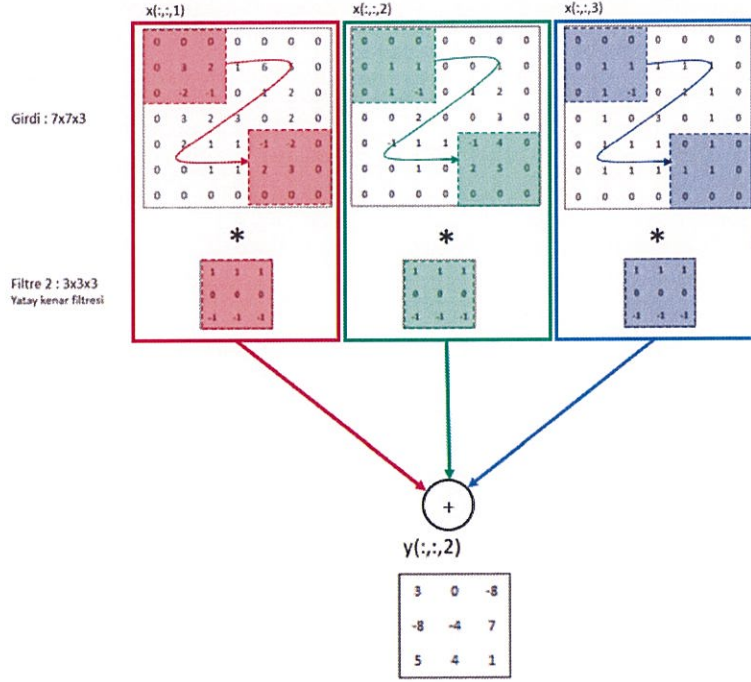
Aynı işlemi son adımda $x(:, :, 1)$ ve $f(:, :, 1)$ için uyguladığımızda $1x(-1) + 0x(-2) + (-1)x0 + 1x2 + 0x3 + (-1)x0 + 1x0 + 0x0 + (-1)x0 = 1$, $x(:, :, 2)$ ve $f(:, :, 2)$ için uyguladığımızda $1x(-1) + 0x4 + (-1)x0 + 1x2 + 0x5 + (-1)x0 + 1x0 + 0x0 + (-1)x0 = 1$ ve $x(:, :, 3)$ ve $f(:, :, 3)$ için uyguladığımızda $1x0 + 0x1 + (-1)x0 + 1x1 + 0x1 + (-1)x0 + 1x0 + 0x0 + (-1)x0 = 1$ elde edilir. 3 değer toplandığında $1 + 1 + 1 = 3$ elde edilir. Bu değer $y(:, :, 1)$ matrisinin son değeridir.

Noktasal çarpımlar ve bu çarpım değerlerinin toplamları sonucunda çıktının birinci katmanı oluşmuştur.



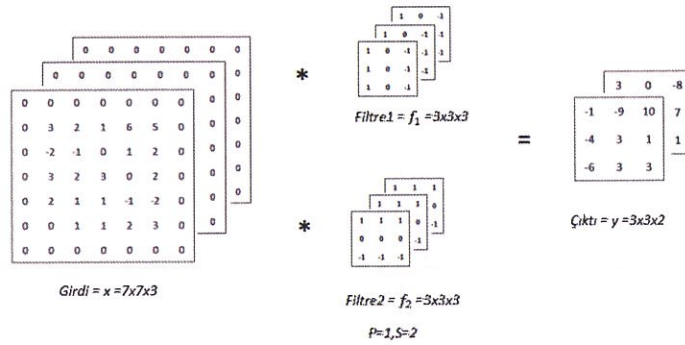
Şekil 2.14. Birinci filtre ile evrişim işlemi

Aynı işlemler ikinci filtre ile girdi arasında gerçekleştirilmiş ve Şekil 2.15'teki çıktıya ait ikinci katman oluşmuştur. Sonuç olarak boyutları 3x3 ve kanal sayısı 2 olan bir çıktı elde edilmiştir.



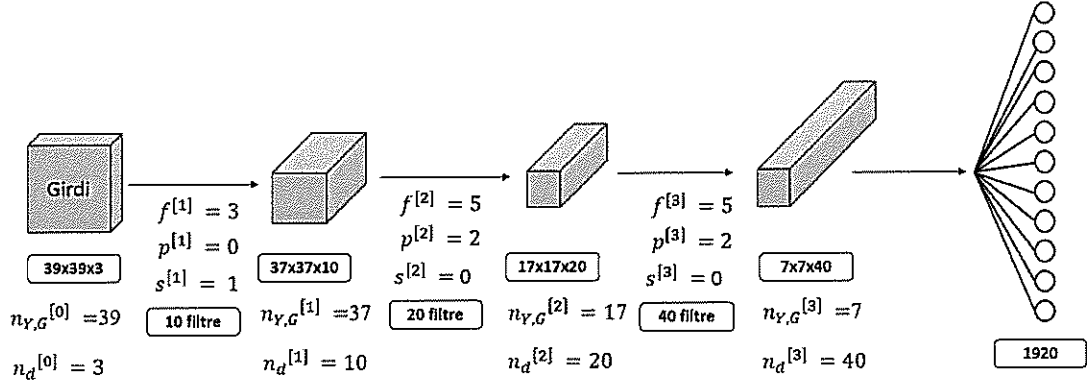
Şekil 2.15. İkinci filtre ile evrişim işlemi

Yapılan tüm evrişim işlemlerine ait sonuç Şekil 2.16'da gösterilmiştir.



Şekil 2.16. RGB imgesi üzerinde 2 filtre ile gerçekleştirilen evrişim işlemi sonucu

Bir RGB imge üzerinde sadece evrişim katmanı uygulanarak sınıflandırma yapmak mümkündür. Küçük boyutlarda bir imge üzerinde Şekil 2.17’teki gibi ardarda evrişim katmanları uygulandığında sınıflandırma işlemi gerçekleşmektedir.



Şekil 2.17 Evrişim katmanlardan oluşan ESA modeli

39x39x3 boyutundaki bir imgeye 1.nci evrişim katmanında 10 farklı 3x3x3 boyutunda filtre uygulandığında ve $p^{[1]} = 0$ ve $s^{[1]} = 1$ hassas parametreleri seçildiğinde evrişim işlemi sonucu 37x37x10 boyutlarında bir çıktı elde edilmiştir. Bu çıktı bir sonraki evrişim katmanının girdisi olacaktır.

2.evrişim katmanında 20 farklı 5x5x10 boyutundaki filtre ile birlikte $p^{[2]} = 0$ ve $s^{[2]} = 2$ parametreleri ayarlanarak 17x17x20 boyutlarında bir çıktı elde edilmiştir. 3.evrişim katmanı bu çıktıya 40 farklı 5x5x20 boyutundaki filtre ile birlikte $p^{[3]} = 0$ ve $s^{[3]} = 2$ olarak seçilen hassas parametreler ile birlikte uygulandığında 7x7x40 boyutunda bir çıktı elde edilmiştir. Bu matris düzleştirme işlemi ile vektöre dönüştürüldüğünde 1960 adet sınıf oluşturulabilir.

Evrişim işlemleri sonucunda imgeye ait yükseklik ve genişlik değerleri azalmış ancak kullanılan filtre sayısına bağlı olarak derinliğin arttığı gözlemlenmiştir.

Aktivasyon katmanı, evrişim katmanını takip eder. Aktivasyon katmanında genellikle Denklem 2.6’da verilen düzleştirici doğrusal birim (ReLU) fonksiyonu kullanılır.

$$ReLU(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.8)$$

Ortaklama katmanı:

Ortaklama katmanı, aktivasyon işleminden sonra yer alır. Bu katmanda anlamsal olarak benzer özellikler birleştirilerek takip eden evrişim katmanı için boyut indirgeme işlemi gerçekleştirilir. Bu katmanda öğrenme gerçekleşmemektedir.

Ortaklama katmanında filtre işlemine tabi tutulan girdinin boyutları azaltılırken derinliği sabit tutulur. Genellikle 2x2 boyutunda filtre kullanılır. Ortaklama katmanına giren evrişim çıktısının yeni boyutları evrişim katmanındaki aynı formül ile hesaplanabilir. Ancak bu katmanda dikkat edilmesi gereken nokta sıfır eklemenin mümkün olmamasıdır. P hassas parametresi 0 olacaktır.

Ortaklama katmanının boyutları Denklem 2.9 ve 2.10'daki eşitliğe göre hesaplanabilir.

$$n_{y,g}^{[l]} = \frac{n_{y,g}^{[l-1]} - f^{[l]}}{s^{[l]}} + 1 \quad (2.9)$$

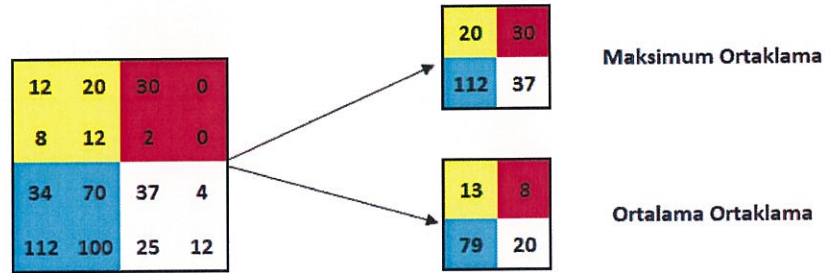
$$n_d^{[l]} = n_d^{[l-1]} \quad (2.10)$$

Ortaklama katmanında Denklem 2.11 eşitliğinde verilen maksimum ve Denklem 2.12 eşitliğinde gösterilen ortalama ortaklama yöntemleri kullanılmaktadır.

Maksimum ortaklama işleminde filtre içerisindeki girdiye ait değerlerin en büyüğü, ortalama ortaklama işleminde ise filtre içerisindeki girdiye ait değerlerin toplamının ortalaması çıkış matrisine yazılmaktadır. Uygulamalarda genellikle maksimum ortaklama tercih edilmektedir.

$$y_{i^{l+1},j^{l+1},d} = \max_{0 \leq i < Y, 0 \leq j < G} x^l_{i^{l+1}x_{Y+i,j^{l+1}x_{G+j},d} \quad (2.11)$$

$$y_{i^{l+1},j^{l+1},d} = \frac{1}{YG} \text{ort}_{0 \leq i < Y, 0 \leq j < G} x^l_{i^{l+1}x_{Y+i,j^{l+1}x_{G+j},d} \quad (2.12)$$



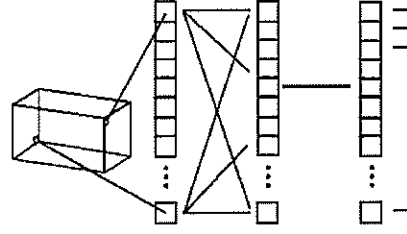
Şekil 2.18. Ortaklama işlem örneği

Örnek ortaklama işlemi Şekil 2.18’de gösterilmiştir. Kaydırma miktarı 2 ve filtre boyutları 2x2 seçilmiştir. Her iki ortaklama işlemi girdiye uygulandığında çıktının değerlerinin değiştiği görülmektedir. 4x4 boyutundaki girdinin yeni boyutları 2x2 olmuştur. Görüldüğü gibi ortaklama işleminde boyutlar azalmıştır.

Tam bağlantılı katman:

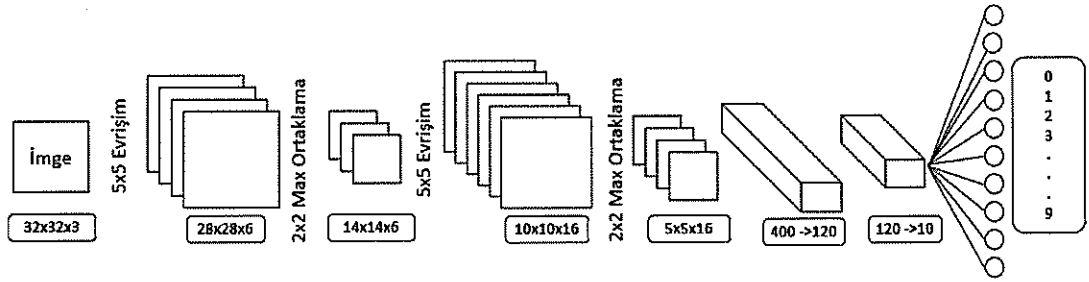
ESA mimarisinde evrişim ve havuzlama katmanının çıktısı düzenlenerek mevcut katmanın her bir düğümü bir sonraki katmanın diğer düğümüne bağlanır. Bu işlem yapılırken matrisler düzleştirme fonksiyonundan geçirildikten sonra vektör haline gelir ve tam bağlantılı katmana bağlanır. Tam bağlı katmanlardaki işlemler herhangi bir yapay sinir ağında gerçekleşen işlemler ile aynıdır. Tam bağlı katman yapısı Şekil 2.19’da gösterilmiştir.

Daha önce 39x39x3 boyutundaki bir imgenin sınıflandırma probleminde sadece evrişim katmanlarının ardarda uygulanarak sınıflandırma işleminin gerçekleştirilebileceği gösterilmiştir.



Şekil 2.19. Düzleştirme ve tam bağlı katman yapısı

Şimdi aynı imgeyi evrişim, ortaklama ve tam bağlı katmanlardan oluşan bir ESA ile sınıflandıralım.



Şekil 2.20. ESA model yapısı

32x32x3 boyutundaki bir rakam sembolü içeren 3 kanallı bir imge boyu 32, genişliği 32 ve derinliği 3 olan bir matris olarak ağa girdi olacaktır. Evrişim katmanında bu imgeye ait özelliklerin çıkarılmasını sağlayacak 6 adet filtre ile evrişim işlemleri tamamlanır ve evrişim katman çıktısı 28x28x6 olur. Evrişim ve ortaklama katmanları arasında aktivasyon fonksiyonu uygulanır ve boyut değişmez. Ortaklama katmanında genellikle 2x2 boyutunda filtreler kullanılır ve evrişim işlemleri sonrasında elde ettiğimiz 28x28x6 boyutundaki çıktının boyutu yarıya düşürülür ancak derinlik değişmez. Bu işlem sonrası çıktının boyutu 14x14x6 olmuştur ve hacim azalmıştır.

Elde edilen bu çıktı tekrar evrişim ve ortaklama işlemlerine tabi tutulur ve tam bağlı katman öncesinde 5x5x16 boyutunda bir çıktı elde edilmiştir. Şekil 2.20’de verilen evrişim ve ortaklama katmanları uygulandıktan sonra çıktı 1x1x400 boyutuna dönüştürülür ve 400 boyutunda bir vektör girdisi alan tam bağlı katmana dönüştürülür.

Tam bağlantılı katman üzerinden mimaride yer alan son katman olan sınıflandırma katmanına bağlanır. Sınıflandırma katmanı çok sınıflı sınıflandırma problemlerinde (4) denkleminde verilen softmax aktivasyon fonksiyonu ile sınıflandırma işlemini gerçekleştirir.

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{k=1}^K e^k} \quad (2.13)$$

ESA mimarisi oluşturulduktan sonra eğitim kümesinden bir veri alınarak eğitim süreci başlatılır. Bu işlemlerin sonucunda alınan çıktı ile eğitim verisi içerisinde yer alan doğru sonuçlar karşılaştırılarak hata hesaplanır ve geriye yayılım algoritması kullanılarak her yinelemede ağırlıklar güncellenir ve hatanın minimuma getirilmesi amaçlanır (Xie ve diğ., 2016, İnik ve Ülker, 2017).

Hassas parametrelerin belirlenmesi işlemleri:

Derin öğrenme modelleri ile birlikte özneliklerin çıkarılması kolaylaşmıştır. Özneliklerin belirlenmesinden ziyade derin öğrenme ağ mimarisinin tasarlanması ve kullanılan algoritmalarındaki parametrelerin belirlenmesi daha fazla önem arz etmektedir.

SVM algoritmalarında kernel fonksiyonunun belirlenmesi, KNN algoritmasında k değerinin belirlenmesi gibi derin öğrenme modellerinde de hassas parametrelerin belirlenmesine karar verilir.

Veri setinin boyutu, epok, yığın boyutu, katman sayısı, ağırlık başlangıç değerleri, aktivasyon fonksiyonu, öğrenme hızı ve optimizasyon algoritmasının belirlenmesi gibi parametreler hassas parametre olarak adlandırılmaktadır.

Veri Seti:

Eğitmenli veya eğitmensiz öğrenme tiplerinde veri kümesinin çeşitliliği ve boyutu çok önemlidir. Veri kümesinin büyüklüğü öğrenme süresini ve model derinliğini arttıracaktır.

Derin öğrenme modelleri çok fazla veriye ihtiyaç duyar. Eğer veri kümesinde yer alan örneklerin yetersiz olması durumunda veri çoğaltma teknikleri ile veri kümesinin boyutları artırılabilir. Bu sayede model başarımları artırılabilir. Özellikle veri kümesinin imgelerden oluştuğu problemlerde veri çoğaltma işleminde orijinal imgedeki hedef bölgenin işlem sonrasında korunması gerekmektedir.

Veri çoğaltma metotları arasındaki en basit metot dikey eksende yansıtmadır. İmge yatay eksende ters çevrilerek yeni imge elde edilir.

Yaygın olarak kullanılan veri çoğaltma metotları arasında rastgele kırpma da yer almaktadır. Ancak bu metotta dikkat edilmesi gereken durum gelişigüzel kırpmaların asıl imgenin yeterince büyük alt kümelerini oluşturuyor olmasıdır. Aksi takdirde tanıtılması hedeflenen bölge yeni imge içerisinde yer almayacaktır.

Renk kaydırma metodu ile veri çoğaltma işlemi gerçekleştirilebilir. Orijinal imge üzerinde R, G, B değerlerinde değişiklikler uygulanarak veri çoğaltma işlemi gerçekleştirilir. R, G, B değerlerindeki rastgele değişiklikler renk bozulmasına yol açabilir. Bu problem Temel Bileşen Çözümlemesi (PCA) yöntemi ile aşılabilmektedir. PCA renk artırımı yöntemiyle imgenin R, G ve B değerleri belirlendikten sonra çoğaltma işlemi değerler oranında gerçekleştirilir. Eğer imge çoğunlukla mor renge sahip ise, kırmızı ve mavi değerlere çokça ekleme ve çıkarma yapılabilir. Yeşil renge ise ekleme çıkarma diğer renklere göre daha az yapılmalıdır. Bu sayede imgenin renk tonlaması değişmeyecek ve renk bozulmalarının önüne geçilecektir.

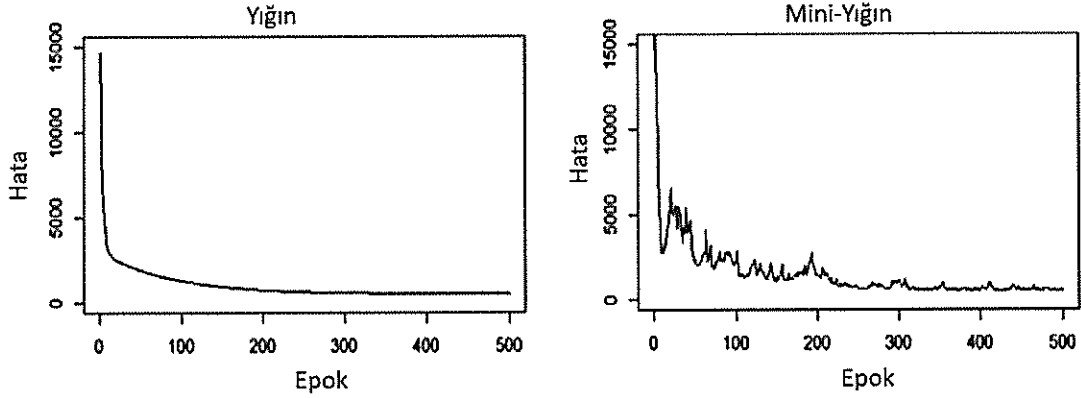
İmge üzerinde döndürme ve parlaklığını değiştirme, yamultma gibi efektler de uygulanabilir ancak imgede bozulmalara yol açabilir.

Veri çoğaltma işlemlerinde hassas parametreler yer almaktadır. Bu parametreler sayesinde uygulanacak renk kaydırma miktarları ve kırma işlemleri ayarlanabilir.

Yığın ve Mini-Yığın Boyutu:

Veri kümesindeki tüm verilerin aynı anda kullanılması ve bu şekilde ağı eğitilmesi işlemine yığın adı verilir. Geriye yayılım algoritması ile ağ eğitiminin her bir epoktaki veri sayısının miktarı, güncellenecek ağırlıkların miktarını ve işlemci başına düşen matematiksel işlem sayısını etkilemektedir.

Ağı bu şekilde eğitilmesi yerine verilerin gruplara ayrılarak eğitilmesi bu problemleri gidermektedir. Mini-yığın parametresi ile ESA modelinin aynı anda kaç veriyi işleyeceği belirlenir. Eğitilmenin mini-yığın parametresi belirlenerek yapılması durumunda eğitilme zamanının azaldığı, kayıp değerinin arttığı gözlemlenmiştir.



Şekil 2.21. Yığın ve Mini-Yığın karşılaştırması

Şekil 2.21'de mini yığın uygulandığında hata değerindeki dalgalanmaların nedeni her epokta yeni ve farklı veriler kullanıldığı için daha önce kullanılan parametrelerin uygun olmaması durumudur. Epok sayısı arttıkça dalgalanmalar azalacaktır. Yığın değerindeki hassas değişiklikler ESA mimarisinde büyük başarımleri etkileri gösterir. Yığın boyutu genellikle GPU belleğine uygun olarak 64 ile 512 arasında seçilmektedir. Yığın

boyutunun küçük seçilmesi gerekmektedir. Modelin büyük gruplar halinde verileri eğitim için alması ezberleme problemlerini ortaya çıkaracaktır.

Veri kümeleri küçük kümelere ayrılmadan önce karıştırılmalıdır. Bu sayede mini yığın işlemi ile küçük veri kümeleri oluşturulduğunda kümelere yer alan veriler rastgele seçilecek ve ezberleme problemlerinin önüne geçilecektir.

Ağırlık Başlangıç Değerleri:

Yapay sinir ağları veya evrimsel sinir ağlarında yer alan A ağırlık değerlerinin ilk değerleri farklı yöntemlere göre belirlenebilir. Genellikle rastgele küçük değerler atanarak modelin eğitilmesi başlatılır. Bir diğer yöntem ise daha önce eğitilmesi tamamlanmış başka bir modelin ağırlık parametreleri kullanılabilir.

Ağırlık değerleri model başlangıcında 0 olacak şekilde belirlenmemelidir.

Epok:

Derin öğrenme modeli eğitilmeden önce veri kümesi küçük parçalara ayrılmaktadır. Veri kümesinin küçük parçalara rastgele bir şekilde ayrılmasının nedenini daha önce açıklanmıştır. Mini yığın olarak adlandırılan bu veri kümesi parçaları sırasıyla eğitim işlemine tabi tutulurlar. İlk parçanın eğitilmesinden sonra modelin başarıyı test edilir ve geriye yayılım algoritması ile ağırlıklar güncellenir. Bu işlemden sonra güncel ağırlık parametreleri ile bir sonraki parça eğitilir ve bu şekilde veri kümesinin son parçası eğitilip ağırlık parametreleri güncellenene kadar döngü devam eder. Eğitim sırasındaki ileri ve geri beslemeli her bir eğitim tur sayısına epok adı verilir.

Eğitim sırasında ilk epoklarda model başarıyı düşük olacaktır. Epok arttıkça model başarıyı da bir artacaktır.

Öğrenme oranı:

Öğrenme oranı, model eğitimi için ayarlanması gereken en önemli hassas parametrelerden biridir ve λ ile gösterilir.

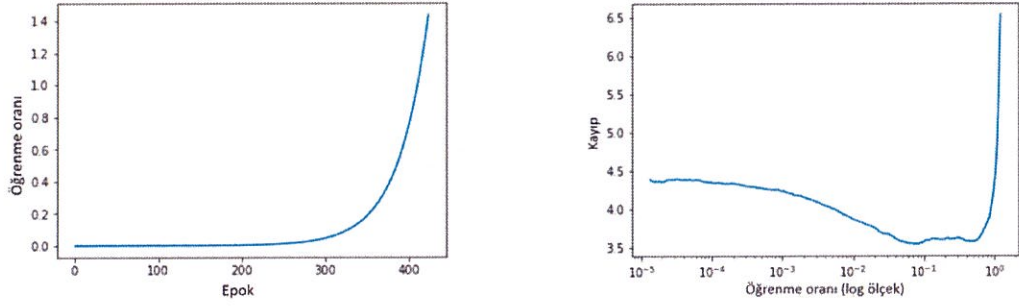
Derin öğrenme algoritmalarındaki eğitim boyunca uygulanan geriye yayılım işlemi ile ağırlık parametreleri her epokta güncellenir. Bu güncelleme işlemi, geriye doğru türev alınarak farkın ortaya çıkarılması ve bu farkın öğrenme oranı ile çarpılarak çıkan değerlerin mevcut ağırlık değerlerinden çıkarılarak yeni ağırlık değerlerinin bulunması ile tamamlanır.

Öğrenme oranı düşükse, eğitim daha güvenilirdir, ancak optimizasyon çok zaman alacaktır, çünkü kayıp fonksiyonunun minimumuna doğru adımlar küçüktür. Öğrenme oranı yüksekse, eğitim yakınsamayabilir, hatta birbirinden uzaklaşmayabilir.

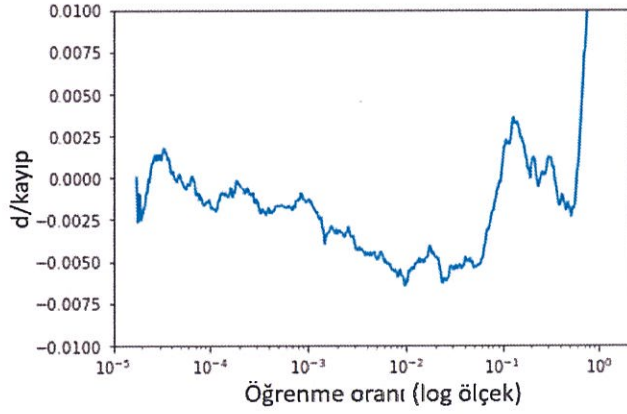
Smith (2015) tarafından yapılan çalışmada bir sinir ağı için öğrenme oranının belirlenmesi konusunda birtakım yöntemler geliştirmiştir. Bir ağı düşük bir öğrenme oranı ile eğitmeye başladığında ve her bir yığın için öğrenme oranı katlanarak arttırıldığında kaybın yavaş yavaş iyileştiği ancak öğrenme oranının çok fazla artmasıyla beraber kaybın da bir anda artışa geçtiği gözlemlenmiştir.

Öğrenme oranı kaybın en hızlı azaldığı noktada seçilmelidir. Şekil 2.22'de gösterildiği gibi epok artışı ile birlikte her mini yığından sonra öğrenme hızı artmaktadır. Kayıp ise başlangıçta azalmaktadır. Öğrenme oranı 0.001 ile 0.01 aralığında kayıp fonksiyonu hızlı bir şekilde azalmıştır.

Öğrenme oranı, epok sayısına göre kayıp fonksiyonunun birinci dereceden türevi hesaplandıktan sonra tespit edilebilir. Y eksenindeki değişim oranını ve x eksenindeki öğrenme oranı Şekil 2.23'deki gibi çizildiğinde kayıp fonksiyonunun minimum noktasındaki öğrenme oranı olması gereken optimum değerdir.



Şekil 2.22. Öğrenme oranı – Kayıp ilişkisi



Şekil 2.23. Öğrenme oranı – Kayıp değişim ilişkisi

Optimizasyon Algoritması:

Derin öğrenme uygulamalarında stokastik gradyan inişi, adam, adagrad, adadelta, adamax gibi optimizasyon algoritmaları kullanılmaktadır. Optimizasyon algoritmaları sayesinde doğrusal olmayan problemlerin çözümünde optimum değer tespit edilir.

Stokastik gradyan inişi, tüm ağırlık güncellemeleri için tek bir öğrenme oranını tutar ve eğitim oranı eğitim sırasında değişmez. Her ağ ağırlığı için bir öğrenme hızı korunur ve öğrenme açılımı olarak ayrı ayrı uyarlanır.

Kingma ve Ba (2015) tarafından geliştirilen Adam optimizasyonu, eğitim verilerine dayanarak yinelemeli ağ ağırlıklarını güncellemek için klasik stokastik gradyan iniş prosedürü yerine kullanılacak bir optimizasyon algoritmasıdır. MNIST veri kümesi ile gerçekleştirilen deneylerde çok katmanlı bir algılayıcının eğitiminde Adam optimizasyon algoritmasının diğer optimizasyon algoritmalarını ile karşılaştırılması gösterilmiştir.

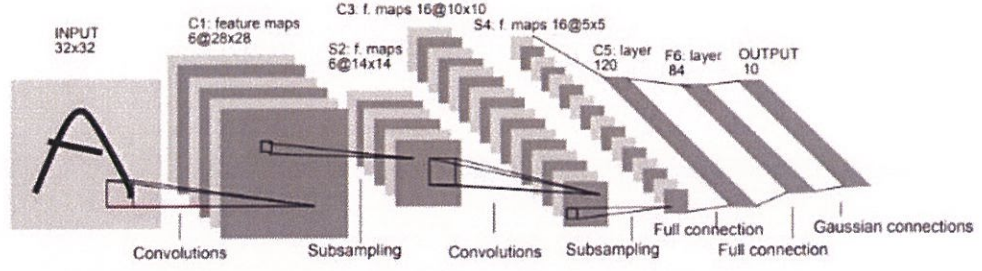
Tez çalışmasında stokastik optimizasyon metotlarından Adam optimizasyon uygulaması kullanılmıştır. Diğer optimizasyon metotları bu çalışma kapsamında kullanılmadığı için detayları verilmemiştir.

ESA Mimarileri:

Literatürde başarılı uygulamaları bulunan LeNet, VGGNet, GoogLeNet, AlexNet ve ResNet gibi ESA mimarileri yer almaktadır. Bu mimarilerin başarımlarını karşılaştırmalarını hedefleyen çok ölçekli bir çalışma örneği Canziani ve diğ. (2017)'de bulunabilir.

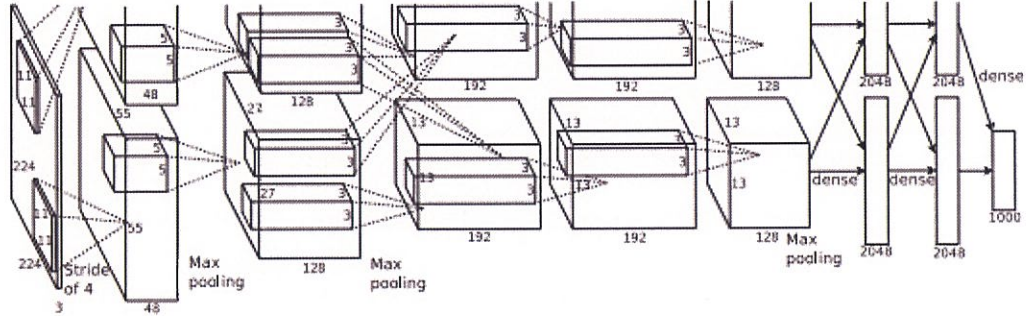
LeNet-5, LeCun ve ark. (1998) tarafından yayınlanmış ilk evrişimli sinir ağı mimarisidir. Banka çekleri ve posta kodları gibi dokümanlardaki numaraların tanınması amacıyla geliştirilmiş ve MNIST veri kümesi ile eğitilmiştir. Şekil 2.24'te gösterilen LeNet mimarisinde iki adet evrişim ve ortaklama katmanları ardarda yerleştirilmiş ve ardından tam bağlı katmanlar yer almıştır. Boyut azaltma işleminde ortalama ortaklama uygulanmıştır. Sigmoid ve hiperbolik-tanjant fonksiyonları da aktivasyon fonksiyonu olarak kullanılmıştır.

[32x32] boyutundaki imgeye sırasıyla [5x5] boyutundaki filtreler ile evrişim katmanı uygulanarak $16 \times 5 \times 5 = 400$ parametre tam bağlı katman girişine uygulanmış ve daha sonra esnek eşikleyici (softmax)'den geçirilmiştir. Bu sayede ağ 0-9 arasındaki 10 adet rakamı sınıflandırabilmektedir. Girdi boyutu (yükseklik x genişlik) katmanlar boyunca ilerledikçe azalmakta, derinlik ise artmaktadır.



Şekil 2.24. LeNet-5 model mimarisini (LeCun ve ark. 1998).

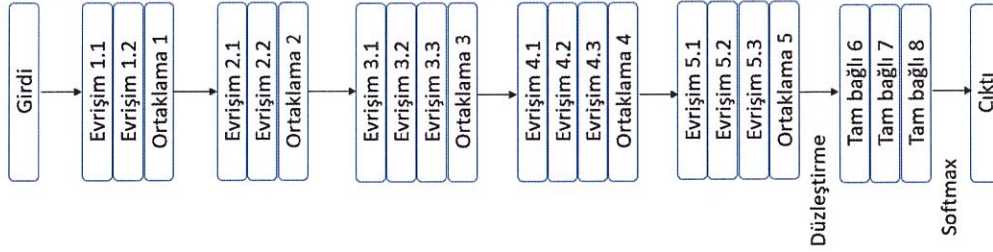
AlexNet, Krizhevsky ve ark. (2012) tarafından tasarlanmış ve 2012 yılında ImageNet geniş ölçekli imge tanıma yarışmasında (ILSVRC) %15.3 kayıp oranı ile başarımını ispat etmiştir. Şekil 2.25’de gösterilen AlexNet mimarisini LeNet modeline benzemektedir ancak LeNet modelinden farklı olarak çift GPU üzerinde çalışabilmesi için iki parçadan oluşmaktadır ve ReLU aktivasyon fonksiyonu ve maksimum ortaklama kullanılmaktadır.



Şekil 2.25. AlexNet model mimarisini (Krizhevsky ve ark. 2012)

VGG-16, Simonyan ve ark.(2014) tarafından ILSVRC yarışması için tasarlanan, sınıflandırma ve yerelleştirme problemlerinde kullanılan ve %7.3 kayıp oranı ile başarımını kanıtlayan bir ESA mimarisidir. Geliştirilme amacı, sinir ağı derinliğinin geniş ölçekli imge tanıma problemlerindeki doğruluğu nasıl etkilediğini araştırmaktır. 16 evrişim katmanından oluşmaktadır ve evrişim katmanları [3x3] boyutundadır. Maksimum ortaklama katmanı ise [2x2] boyutundadır. En önemli farkı, aşağıdaki Şekil 2.26’da

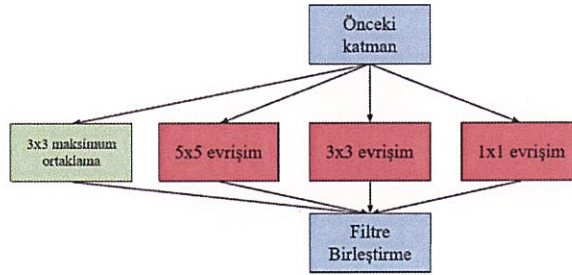
gösterilen evrişim ve ortaklama katmanları ardı ardına gelmek zorunda değildir. İkili veya üçlü evrişim katmanları sonrasında ortaklama katmanı konumlandırılabilir.



Şekil 2.26. VGG-16 model mimarisi (Simonyan ve ark. 2014)

GoogleNet, Google araştırma ekibinde yer alan Szegedy ve ark. (2015) tarafından geliştirilen ve % 6.2'lik hata oranı ile ILSVRC yarışmasında birinciliği kazanan bir ESA mimarisidir.

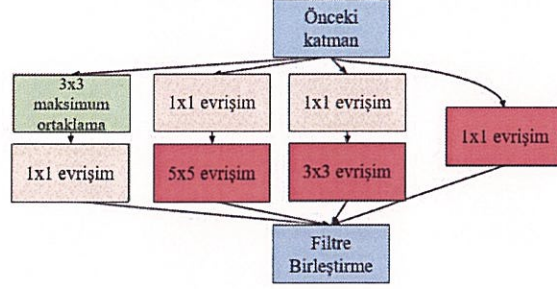
Inception adı verilen modüller içerisinde ağ içinde ağ yapısı oluşturulmuş ve model başarımının arttığı gözlemlenmiştir. Şekil 2.27'de temel inception yapısı gösterilmiştir. Bu yapı ile farklı filtreler ile evrişim ve maksimum ortaklama işlemleri paralel olarak gerçekleştirilmektedir.



Şekil 2.27. Temel inception yapısı (Szegedy ve ark. 2015)

Inception modüllerinin girdiye uygulanması sonrasında çıkış boyutunun büyüklüğü ve parametre sayısı paralel işlemler nedeniyle çok büyümektedir. Boyut azaltma işlemi için

1×1 boyutlarındaki evrişim katmanları her evrişim katmanlarının öncesine eklenerek bu problemin önüne geçilmiştir. Şekil 2.28’da boyutsal küçülme uygulanmış inception modülü verilmiştir.



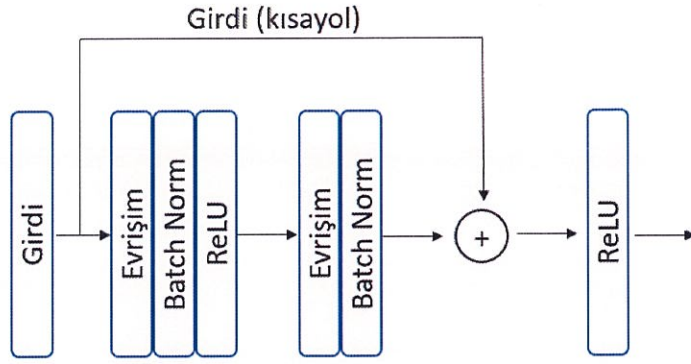
Şekil 2.28. Boyutsal küçülme uygulanmış inception modülü (Szegedy ve ark. 2015)

GoogleNet mimarisi 9 inception modülünden oluşturulan 22 katmanlı bir model yapısına sahiptir. Aktivasyon fonksiyonu olarak ReLU fonksiyonu kullanılmıştır.

ResNet, Microsoft araştırma ekibinde yer alan He ve ark. (2016) tarafından geliştirilen ve ILSVRC yarışmasında %3.57’lik hata oranı ile birinciliği kazanan bir ESA mimarisidir. Doğrusal ve ReLU aktivasyon fonksiyonları kullanılır. ESA’larda teorik olarak katman sayısındaki artış ile model başarısının artacağı beklenmektedir. Ancak pratikte aktivasyon fonksiyonlarının ürettiği sonuçların geriye yayılım algoritmasındaki yapılacak matematiksel işlemler sonrasında sifra yakınsaması problemi ortaya çıkmaktadır.

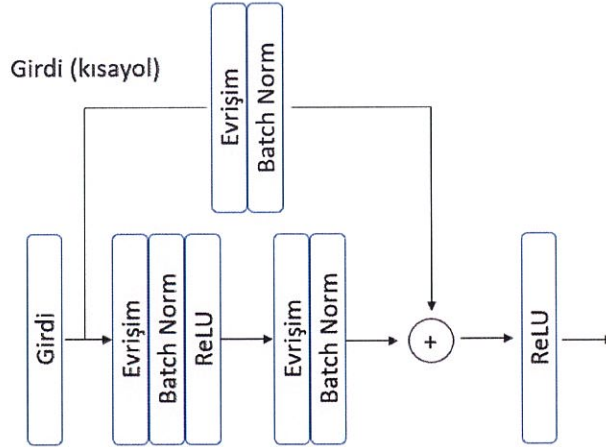
Klasik ESA mimarilerinden farklı olarak artık değerler kısayol bağlantıları sayesinde iki katmanda bir katman sonuna eklenir. Bu sayede sifra yakınsama problemine çözüm getirilmiş, derin ağı eğitme hızının arttığı gözlemlenmiştir.

Şekil 2.29’de Resnet modellerinde kullanılan standart özdeşlik bloğu gösterilmiştir.



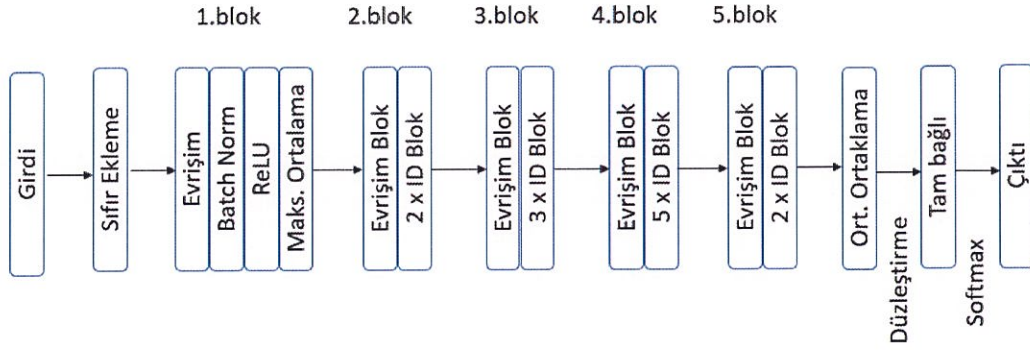
Şekil 2.29. Resnet Özdeşlik bloğu.

Özdeşlik bloğunun dışında Resnet modelinde kullanılan Evrişim bloğu Şekil 2.30’da verilmiştir. Kısayolda yer alan evrişim katmanı girdinin boyutunu azaltarak ana yoldaki çıktının boyutlarına dönüştürülür. Böylece boyutlar sonraki ekleme aşaması için eşleşir.



Şekil 2.30. Resnet Evrişim bloğu.

Şekil 2.31’de ise 34 katmanlı bir Resnet modeli gösterilmektedir.



Şekil 2.31 Resnet-34 model mimarisi.

ILSVRC yarışmasına katılan farklı ekipler 1000 sınıflı toplam 1 milyon imge içeren ImageNet veri kümesi üzerinde nesne sınıflandırma problemine çözümler getirmişlerdir ve elde ettikleri hata oranları Çizelge 2.2’de gösterilmiştir.

Çizelge 2.2. ILSVRC yarışması sonuçları

Yıl	ESA Modeli	Ekip	Sıralama	Hata Oranı	Katman Sayısı	Parametre Sayısı
1998	LeNet	LeCun ve ark.	-	-	5	60 bin
2012	AlexNet	Krizhevsky ve ark.	1.	%15.3	8	60 milyon
2013	ZFNet	Zeiler ve Fergus	1.	%14.8	8	-
2014	GoogleNet	Szegedy ve ekibi	1.	%6.67	22	4 milyon
2014	VGGNet	Simonyan ve Zisserman	2.	%7.3	19	138 milyon
2015	ResNet	He ve ark.	1.	%3.57	152	-

3. MATERYAL VE YÖNTEM

3.1. Veri Kümesi

Trafik işaretlerinin tanınmasını amaçlayan çalışmaların başarımlarını analizlerinde kullanılmak üzere açık erişime sahip olan veri kümeleri bulunmaktadır (Alvaro ve diğ. 2018). Bu çalışmada önerilen SGDS'nin başarımlarını incelemesi yapılırken bu problemin çözümünde Alman trafik işaretleri veri kümesi (GTSRB) kullanılmıştır.

Almanya'daki Institut für Neuroinformatik tarafından hazırlanmış GTSRB veri kümesindeki imgeler 15x15'ten 250x250 piksellik farklı boyutlarda ppm formatında sunulmuştur. Çizelge 3.1'de veri kümesindeki 43 sınıfa ait toplam 51.839 trafik işareti imgesinin detayları verilmiştir. Bu imgelerin 39.209 tanesi eğitim kümesinde yer alırken 12.630 tanesi ise test kümesi için ayrılmıştır.

Veri kümesi hakkında detaylı bilgi Stallkamp ve diğ. (2011)'de bulunabilir. 43 ayrı sınıfa ait trafik işaretlerinin orijinal imgeleri Şekil 3.1'de, veri kümesinde yer alan trafik işareti imgelerine ait örnekler ise Şekil 3.2'de verilmiştir.



Şekil 3.1. Trafik işaretleri sembolleri



Şekil 3.2. GTSRB veri kümesi trafik işareti örnekleri

Çizelge 3.1. GTSRB veri kümesi detayları

	Toplam Veri Sayısı	Eğitim Veri Kümesi	Test Veri Kümesi	Sınıf	Çeşitlilik
GTSRB	51.839	39.209	12.630	43	~30 imge/trafik işareti

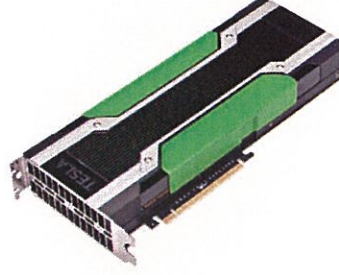
3.2. Donanımlar ve Yazılımlar

Bu çalışmada kullanılan ESA mimarileri farklı işletim sistemleri ile eğitilmiş ve sistemlerin olumlu ve olumsuz tarafları gözlemlenmiştir.

3.2.1. Google Colab

Colab, Google firmasının makine öğrenmesi ve derin öğrenme çalışmalarının geliştirilmesi için son kullanıcıya ücretsiz olarak sunduğu bir bulut servisi. Uzaktan erişim sayesinde Jupyter notebook ortamında GPU, tensör işlem birimi (TPU) ve merkezi işlemci birimi (CPU) desteği ile derin öğrenme modellerinin eğitilmesi mümkündür. Ayrıca Tensorflow, Keras, Pytorch ve OpenCV gibi derin öğrenme kütüphanelerini hazır olarak içerisinde barındırmaktadır.

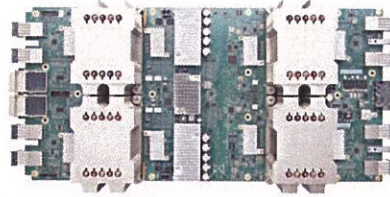
Google Colab servisi geliřtiricilere Nvidia firmasına ait Tesla K80 modeli üzerinde derin öğrenme uygulamaları geliřtirilebilmektedir. Şekil 3.3’de görülen ekran kartlarında iki adet GPU bulunmaktadır. Bu sayede aynı anda iki büyük veri kümesi üzerinde işlem yapabilme kapasitesine sahiptir.



Şekil 3.3. Nvidia Tesla K80 GPU modeli (Anonim 2019a)

4.992 adet CUDA paralel işlem çekirdekleri sayesinde uygulamalar yalnızca CPU içeren sistemlere göre 10 kat daha hızlı çalışmaktadır.

Şekil 3.4’te Google firmasının derin öğrenme algoritmalarının daha hızlı bir şekilde eğitilmesi için geliřtirdiđi Tensör işlem birimi ise dört bağımsız çipten oluşmaktadır.



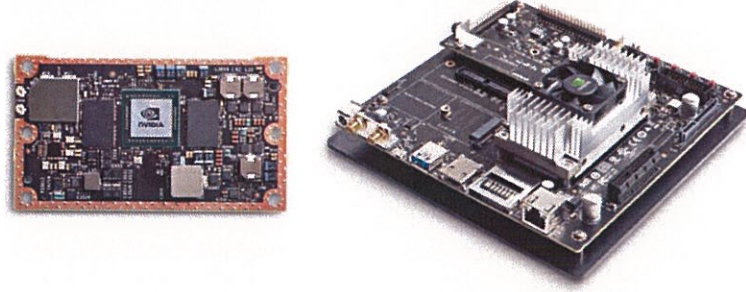
Şekil 3.4. Google TPU modeli (Anonim 2019a)

3.2.2. Nvidia Jetson TX2 Geliřtirme Kartı

Nvidia firmasının tasarladığı Şekil 3.5’te gösterilen Jetson TX2 geliřtirme kartı GPU ve CPU yanında birtakım giriş/çıkış ara birimlerini desteklemektedir.

256 Cuda paralel çekirdekli GPU ve iki çekirdekli 64 bit Denver 2 ve dört çekirdekli ARM A57 CPU'lardan oluşmaktadır.

Drone, insansız hava araçları ve otonom robotic sistemler gibi mobil sistemler için geliştirilen derin öğrenme uygulamaları geliştirme kartı üzerinde bulunan algılayıcılar sayesinde gerçek zamanlı olarak test edilebilmektedir.



Şekil 3.5. Nvidia Jetson Tx2 işlemcisi ve geliştirme kartı (Anonim 2019b)

3.2.3. Kişisel Bilgisayar

Tasarlanan ESA modelleri GPU bulunmayan bir bilgisayar üzerinde eğitilmiştir. Sistem bilgileri Şekil 3.6'te gösterilmiştir.

System	
Processor:	Intel(R) Core(TM) i7-4600U CPU @ 2.10GHz 2.70 GHz
Installed memory (RAM):	8,00 GB
System type:	64-bit Operating System, x64-based processor

Şekil 3.6. PC sistem bilgileri

3.2.4. Yazılımlar

Makine öğrenmesi ve derin öğrenme çalışmaları farklı yazılım dilleri ile geliştirilmektedir. Bu alanda en popüler yazılım dilleri ise Python (%76,3), R (%59,2),

C/C++ (%19,2), Java (%18,3) dir. Özellikle Python sağladığı birçok derin öğrenme kütüphanesi sayesinde sınıflandırma problemlerinde en çok kullanılan programlama dilidir.

Derin öğrenme için çeşitli üniversiteler ve şirketlerce geliştirilmiş farklı özelliklere sahip birçok çatı ve kütüphane bulunmaktadır. Kullanılan popüler kütüphaneler ve özellikleri Çizelge 3.2’de özetlenmiştir.

Çizelge 3.2. Derin öğrenme kütüphaneleri ve özellikleri

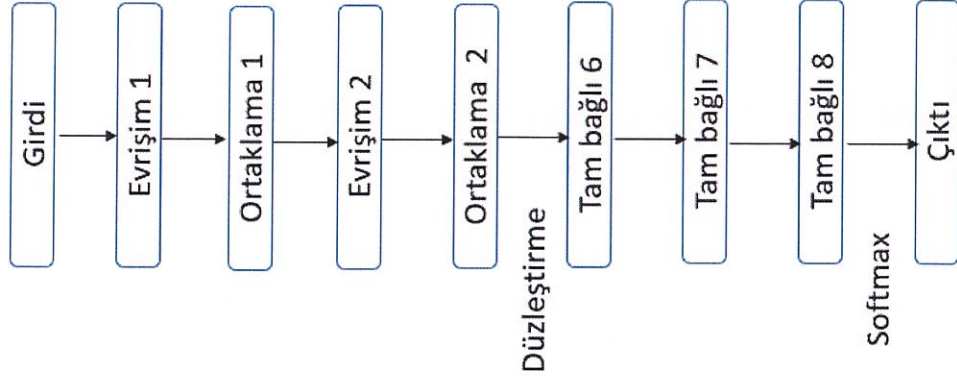
Kütüphane	Geliştirici	Programlama Dili
Tensorflow	Google	Python
Caffe	Berkeley Vision and Learning Center	Python
PyTorch	Google & Facebook	Python
Keras	Google	Python

Tez çalışması kapsamında Tensorflow ve Keras kütüphanelerini destekleyen Python yazılım dili kullanılmıştır.

3.3.Önerilen Modeller

Bu çalışmada trafik işaretlerini tanıyan ESA temelli bir sürücü güvenlik destek sistemi önerilmiştir. Dört farklı ESA mimarisi kullanılmıştır ve modellerin başarımları karşılaştırılmıştır.

Birinci ESA modeli Lenet-5 modelinden ilham alınarak oluşturulmuştur. Modelin katman yapısı Girdi-Evrişim-Aktivasyon-Ortaklama-Evrişim-Aktivasyon-Ortaklama-Tam bağlı-Tam bağlı-Tam bağlı-Çıktı şeklinde sıralanması ile oluşturulmuştur.



Şekil 3.7. İki katmanlı ESA katman yapısı

Önerilen sistemde kullanılan ESA açık kaynak kodlu TensorFlow kütüphanesinden faydalanılarak Python programlama dili ile gerçekleştirilmiştir (Anonim 2018). Sistemin başarımlarını incelemesi GTSRB veri kümesinde yer alan trafik işaretleri kullanılarak yapılmıştır. Önerilen SGDS’de kullanılan ve yapılan deneyler sonucunda karar verilen ESA katman yapısı Şekil 3.7’te sunulmuştur.

Kullanılan veri kümelerindeki imgelerin boyutları 15x15 ile 250x250 boyutlarında farklılık göstermektedir. Bu nedenle verilerin giriş katmanına uygulanması öncesinde 32x32x3 boyutuna indirgenmesi gerçekleştirilmiştir. Daha sonra veri kümesinde yer alan imgeler (32x32x3 RGB) gri ölçekli (32x32x1) biçime dönüştürüldükten sonra normalize edilerek kullanılmıştır.

Katman yapısını detaylı incelendiğinde Çizelge 3.3’te gösterildiği gibi birinci evrişim katmanında 6 adet filtre kullanılmıştır. Filtrelerin boyutları 5x5 büyüklüğündedir. Sıfır ekleme $p1=0$ ve kaydırma miktarı $s1=1$ olarak belirlenmiştir. 32x32x1 boyutundaki imgeler ile 5x5 boyutundaki 6 adet filtre evrişim işlemine tabi tutulduğunda $n_{y,g} = (32 - 5 + 0/1) + 1 = 28$ olarak belirlenir. n_c ise filtre sayısına göre değişiklik göstermektedir. Bu durumda $n_c = 6$ olarak belirlenmiştir. Evrişim işlemi sonunda çıktı 28x28x6 boyutuna erişmiştir ve Relu aktivasyon fonksiyonu uygulanmıştır. Aktivasyon fonksiyonu sonrası

2x2 boyutunda maksimum ortaklama katmanı uygulanmıştır. Kaydırma miktarı $s_1=2$ olarak belirlenmiştir. Sıfır ekleme ortaklama katmanında uygulanmamaktadır. Ortaklama katman çıktısının boyutları $n_{y,g}=(28-2)/2+1=14$ olarak belirlenmiş ve 14x14x6 boyutlarında çıktı elde edilmiştir. Böylece birinci evrişim katmanındaki işlemler tamamlanmıştır.

14x14x6 boyutlarındaki girdi 5x5 boyutlarında 16 adet filtre ile ikinci evrişim katmanına tabi tutulmuştur. Sıfır ekleme $p_2=0$ ve kaydırma miktarı $s_2=1$ olarak belirlenmiştir. Bu durumda $n_{y_2,g_2}=(14-5+0)/1+1=10$ elde edilmiş ve ikinci evrişim katmanının çıktısı 10x10x16 olarak elde edilmiştir ve Relu aktivasyon fonksiyonu uygulanmıştır. Aktivasyon fonksiyonu sonrası 2x2 boyutundaki filtre ve 2 olarak belirlenmiş kaydırma miktarı ile ikinci maksimum ortaklama katmanı uygulanmış ve $n_{y,g}=(10-2+0/2)+1=5$ ve $n_c=16$ olarak 5x5x16 boyutlarında bir çıktı elde edilmiştir. Böylece ikinci evrişim katmanı da tamamlanmıştır.

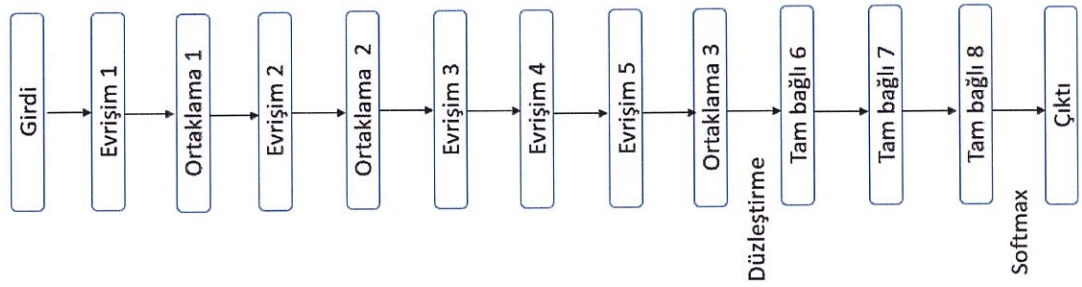
Çizelge 3.3. İki katmanlı ESA modeli katman yapısı

Katman	Tanım	Girdi	Çıktı
Evrişim	filtre: 5x5; kaydırma:1x1; sıfır ekleme: 0	32x32x3	28x28x6
Maksimum Ortaklama	filtre: 2x2; kaydırma:2x2;	28x28x6	14x14x6
Evrişim	filtre: 5x5; kaydırma:1x1; sıfır ekleme: 0	14x14x6	10x10x16
Maksimum Ortaklama	filtre: 2x2; kaydırma:2x2;	10x10x16	5x5x16
Düzleştirme	Input 5x5x16 -> çıktı 400	5x5x16	400
Tam bağlı	Her nöron bir sonraki katmana bağlı	400	120
Tam bağlı	Her nöron bir sonraki katmana bağlı	120	80
Tam bağlı	Her nöron bir sonraki katmana bağlı	80	43

Tam bağlı katmanın bir YSA özelliği taşıdığı ve matrisler yerine vektör girdi aldığını belirtilmişti. Düzleştirme işlemi ile 5x5x16 boyutundaki matris 400x1 boyutunda bir

vektöre dönüştürülmüştür. Tam bağlı katmanda 400 birimlik girdi 120 birime bağlanmıştır. Bu katman sonrasında Relu aktivasyon fonksiyonu uygulanmıştır. İkinci tam bağlı katmanda girdi sayısı 120 birimdir ve 84 birime bağlanmıştır. Bu katman sonrasında Relu aktivasyon fonksiyonu uygulanmıştır. Üçüncü tam bağlı katmanda 84 birimlik girdiler 43 birimlik çıktıya bağlanmıştır. Daha sonra Softmax sınıflandırıcı yardımıyla trafik işaretlerinin sınıflandırılması gerçekleştirilmiştir.

İkinci ESA modeli Alexnet modelinden ilham alınarak oluşturulmuştur. Şekil 3.8’de verilen modelin katman yapısı Girdi-Evrişim-Ortaklama-Evrişim-Ortaklama-Evrişim-Evrişim-Evrişim-Ortaklama-Tam bağlı-Tam bağlı-Tam bağlı-Çıktı şeklinde sıralanması ile oluşturulmuştur.



Şekil 3.8 AlexNet temelli ESA mimarisi

Katman yapısı incelendiğinde Çizelge 3.4’te gösterildiği gibi birinci evrişim katmanında $32 \times 32 \times 1$ boyutundaki girdi 9 adet 5×5 boyutundaki filtre ile evrişime tabi tutulmuştur ve $28 \times 28 \times 9$ boyutunda çıktı elde edilmiştir. 2×2 boyutundaki maksimum ortaklama işlemi ile yeni boyutlar $14 \times 14 \times 9$ olarak güncellenmiştir.

Bir sonraki evrişim katmanında 3×3 boyutundaki 32 adet filtre ile evrişim işlemi uygulanmış ve $12 \times 12 \times 32$ boyutunda çıktı elde edilmiştir. 2×2 boyutundaki maksimum ortaklama işlemi ile $6 \times 6 \times 32$ boyutlarında yeni çıktı edilmiştir.

Sonrasında 3×3 boyutundaki 48 adet, 64 adet ve 96 adet filtre ile 3 kez sırasıyla evrişim işlemi uygulanmış ve $6 \times 6 \times 96$ boyutunda çıktı elde edilmiştir. 2×2 boyutundaki

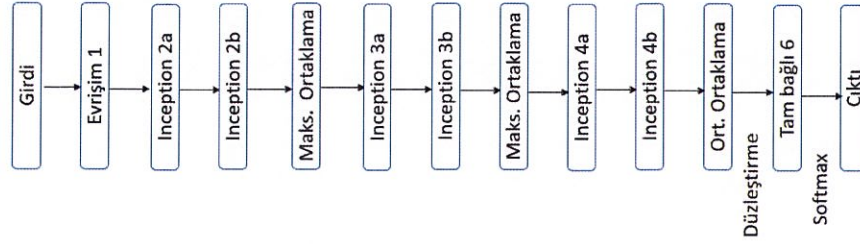
maksimum ortaklama katmanı ile çıktı 3x3x96 olarak güncellenmiştir ve softmax ile sınıflara ayrılmıştır.

Çizelge 3.4. AlexNet temelli ESA modeli katman yapısı

Katman	Tanım	Girdi	Çıktı
Evrişim	filtre: 5x5; kaydırma:1x1; sıfır ekleme: 0	32x32x1	28x28x9
Maksimum Ortaklama	filtre: 2x2; kaydırma:2x2;	28x28x9	14x14x9
Evrişim	filtre: 3x3; kaydırma:1x1; sıfır ekleme: 0	14x14x9	12x12x32
Maksimum Ortaklama	filtre: 2x2; kaydırma:2x2;	12x12x32	6x6x32
Evrişim	filtre: 3x3; kaydırma:1x1; sıfır ekleme: 1	6x6x32	6x6x48
Evrişim	filtre: 3x3; kaydırma:1x1; sıfır ekleme: 1	6x6x48	6x6x64
Evrişim	filtre: 3x3; kaydırma:1x1; sıfır ekleme: 1	6x6x64	6x6x96
Maksimum Ortaklama	filtre: 2x2; kaydırma:2x2;	6x6x96	3x3x96
Düzleştirme	Girdi 3x3x96 -> Çıktı 864	3x3x96	864
Tam bağlı	Her nöron bir sonraki katmana bağlı	864	400
Tam bağlı	Her nöron bir sonraki katmana bağlı	400	160
Tam bağlı	Her nöron bir sonraki katmana bağlı	160	43

Üçüncü ESA modeli GoogleNet modelinden ilham alınarak oluşturulmuştur. Şekil 3.9’da verilen modelin katman yapısı Evrişim-Inception-Inception-Maksimum Ortaklama-Inception-Inception-Maksimum Ortaklama-Inception-Inception-Ortalama Ortaklama-Düzleştirme-Tam bağlı-Softmax şeklinde sıralanması ile oluşturulmuştur. Her inception modülü Şekil 2.29’daki katmanları içermektedir. Basit inception modeli yerine 1x1 evrişim katmanları içeren paralel işlem yapma yeteneği sağlayan bu yapı kullanılmıştır. [1x1] evrişim katmanı eklenerek işlem karmaşıklığının önüne geçilmiştir. Dolayısıyla çıkış boyutunun büyüklüğü ve parametre sayısı çok büyümektedir.

GoogleNet modeli 22 katmandan oluşmaktadır ve ekte model mimarisi gösterilmiştir. Kullanılan Google Net modeli 14 katmandan oluşmaktadır. Ortaklama katmanlarında derinlik değişmediği için katman olarak sayılmamaktadır. Her inception modülünde 2 katman yer almaktadır.



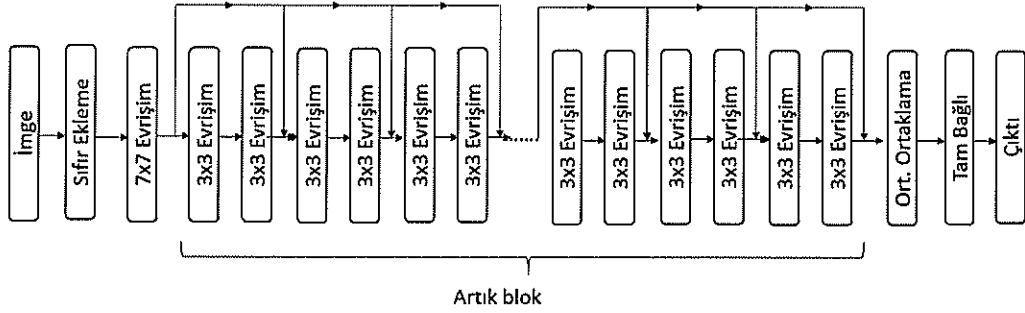
Şekil 3.9 GoogleNet temelli ESA mimarisi

Çizelge 3.5'te görüldüğü üzere ağ başlangıcındaki evrişim katmanında 3×3 boyutundaki 64 farklı filtre ve 2×2 'lik kaydırma oranı ile $32 \times 32 \times 1$ boyutundaki imgeler evrişim işlemine tabi tutularak $16 \times 16 \times 64$ boyutunda çıktı elde edilmiştir. Sırasıyla Inception 2a ve Inception 2b modülüne giren $16 \times 16 \times 64$ boyutundaki yapı $16 \times 16 \times 480$ boyutuna erişmiştir. 3×3 boyutundaki filtre ve 2×2 'lik kaydırma oranı ile $7 \times 7 \times 480$ boyutu elde edilmiştir ve boyutlar küçülmüştür. Inception(3a) ve Inception(3b) modülü ile evrişim ve ortaklama işlemlerinden geçirilen $7 \times 7 \times 480$ boyutundaki girdi $7 \times 7 \times 512$ boyutuna erişmiş ve bir sonraki maksimum ortaklama katmanı sonrasında $3 \times 3 \times 512$ boyutuna erişmiştir. Inception(4a) ve Inception(4b) katmanlarında evrişim ve ortaklama işlemleri uygulandığında $3 \times 3 \times 1024$ boyutunda çıktı elde edilmiştir. 3×3 boyutundaki ortalama ortaklama katmanı sonrasında $1 \times 1 \times 1024$ boyutunda çıktı elde edilmiş ve düzleştirme fonksiyonu sayesinde vektör elde edilmiştir. Son katman olan tam bağlı katman ve softmax olasılık fonksiyonu kullanılarak trafik işaretleri 43 farklı sınıfa ayrılmıştır.

Dördüncü ESA modeli ResNet modelinden ilham alınarak oluşturulmuştur. Şekil 3.10'da gösterilen model 34 katmandan oluşmaktadır ve birbirini takip eden her iki evrişim katmanının çıktısına giriş eklenerek sıralanmıştır. Bu sayede ID ve evrişim bloklarını kapsayan ResNet modeli oluşturulmuştur.

Çizelge 3.5. GoogleNet temelli ESA modeli katman yapısı

Katman	Tanım	Girdi	Çıktı
Evrişim	filtre: 3x3; kaydırma:2x2; sıfır ekleme: 0	32x32x1	16x16x64
Inception(2a)		16x16x64	16x16x256
Inception(2b)		16x16x256	16x16x480
Maksimum Ortaklama	filtre: 3x3; kaydırma:2x2	16x16x480	7x7x480
Inception(3a)		7x7x480	7x7x512
Inception(3b)		7x7x512	7x7x512
Maksimum Ortaklama	filtre: 3x3; kaydırma:2x2	7x7x512	3x3x512
Inception(4a)		3x3x512	3x3x832
Inception(4b)		3x3x832	3x3x1024
Ortalama Ortaklama	filtre: 3x3; kaydırma:1x1	3x3x1024	1x1x1024
Tam bağlı	Her nöron bir sonraki katmana bağlı	1024	43

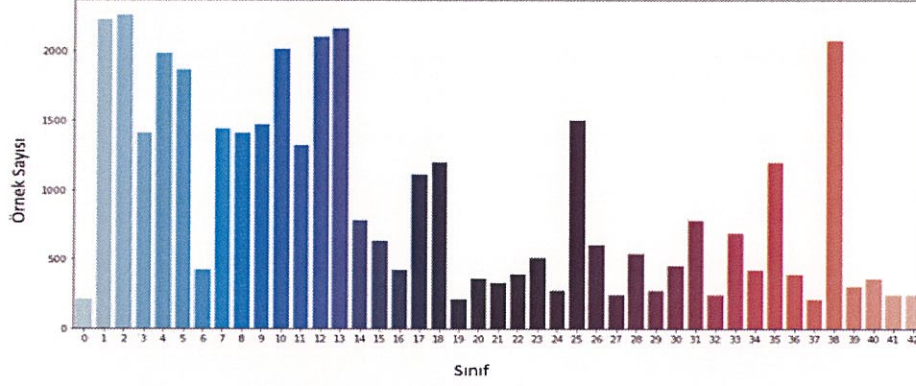


Şekil 3.10 Resnet temelli ESA mimarisi

Şimdiye kadar ele alınan LeNet, AlexNet, GoogleNet ve Resnet modellerinin eğitilmesine ait hassas parametrelerin belirlenmesi bir sonraki bölümde anlatılacaktır.

4. BULGULAR ve TARTIŞMA

GTSRB eğitilme veri kümesinde yer alan imge örneklerinin sınıflar bazındaki dağılımı Şekil 4.1’de verilmiştir. İlgili dağılım incelendiğinde sınıf bazındaki örnek dağılımlarının dengesiz olduğu görülmektedir.

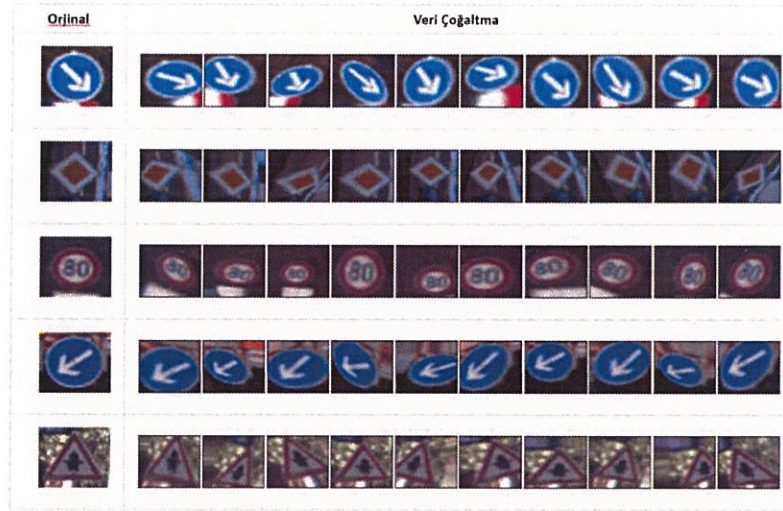


Şekil 4.1 GTSRB veri kümesi için sınıf bazında imge örneği sayıları

Bu dengesizliğin giderilmesi için az sayıda örneğe sahip olan sınıflar için yeni imge örnekleri türetilmiştir. İmge türetme işlemi var olan örneklerin kopyalanması ve daha sonra rastgele döndürme, parlaklık değiştirme ve %20 oranında yakınlaştırma işlemlerine tabii tutulması yoluyla gerçekleştirilmiştir. Şekil 4.2’de imge türetme işlemi sonrası elde edilen bazı örnekler gösterilmektedir.

Böylece eğitim kümesinde yer alan her bir sınıf 3.644 örneğe ulaşmıştır. Sonuç olarak eğitim kümesi 39.209 imge örneğinden 156.692 adet örneğe arttırılmıştır. Test kümesinde ise herhangi bir değişiklik yapılmamıştır.

GTSRB verisinde eğitilme veri kümesinde yer alan imge örneği sayısının başarımlar üzerindeki etkisi Çizelge 4.1’de verilmiştir. Düşük epok sayılarında bile sınıflandırma doğruluğunda yaklaşık %2’lik bir iyileşme görülmektedir.



Şekil 4.2. Arttırılmış imge örnekleri

Çizelge 4.1. Orjinal ve arttırılmış veri kümesi ile başarımları

Veri Kümesi	Örnek Sayısı	Epok	Sınıflandırma Doğruluğu
Orjinal Veri Kümesi	39209	10	%88,30
Arttırılmış Veri Kümesi	156692		%91,00
Orjinal Veri Kümesi	39209	20	%90,50
Arttırılmış Veri Kümesi	156692		%92,10

Önerilen SGDS’de kullanılan ESA’nın eğitime parametrelerinin (yığın boyutu, öğrenme oranı ve Epok sayısı) seçimi ızgara arama yöntemi ile gerçekleştirilmiştir. İlgili seçim için yapılan denemeler Çizelge 4.2, Çizelge 4.3 ve Çizelge 4.4’te sunulmuştur.

Çizelge 4.2. Yığın boyutuna bağlı başarımları

Deneme	Epok	Yığın Boyutu	Öğrenme Oranı	Sınıflandırma Doğruluğu
1	5	128	0.001	%88,30
2		500		%81,90
3		1000		%78,60
4		5000		%40,60

Çizelge 4.3. Öğrenme oranına bağlı başarımların değişimi

Deneme	Epok	Yığın Boyutu	Öğrenme Oranı	Sınıflandırma Doğruluğu
1	5	128	0.001	%91,60
2			0.006	%90,00
3			0.01	%88,60
4			0.1	%52,00

Yapılan ızgara arama sonucunda yığın boyutu ve öğrenme oranı için en uygun değerler olarak 128 ve 0.001 seçilmiştir.

Çizelge 4.4. Epok'a bağlı başarımların değişimi

Deneme	Epok	Yığın Boyutu	Öğrenme Oranı	Sınıflandırma Doğruluğu
1	5	128	0.001	%88,30
2	10			%91,00
3	20			%92,10

Kullanılan 4 ayrı ESA mimarisi için deneysel olarak kabul edilen hassas parametreler kullanılmıştır.

Çizelge 4.5'te kullanılan donanımlara bağlı modellerin eğitim süreleri gösterilmiştir.

Çizelge 4.5. ESA modellerinin donanımlara göre öğrenme süreleri

Donanımlar	LeNet	AlexNet	GoogleNet	ResNet
Colab GPU	45 dk	30 dk	20 dk	20 dk
Colab TCU	50 dk	35 dk	22 dk	21 dk
Jetson Tx2	80 dk	60 dk	45 dk	35 dk
PC	120 dk	100 dk	80 dk	60 dk

Kullanılan 4 ayrı ESA mimarisi için deneysel olarak kabul edilen hassas parametreler kullanılmıştır.

Yığın boyutu ve öğrenme oranı için en uygun değerlerin seçimi yapıldıktan ESA'nın eğitimi örnek sayısı artırılmış GTSRB eğitilme veri kümesi kullanılarak gerçekleştirilmiştir ve GTSRB başarımı Çizelge 4.6'da gösterilmiştir.

Çizelge 4.6. Önerilen sistemin GTSRB için başarımı

Arttırılmış Eğitim Kümesi	Arttırılmış Test Kümesi	Epok	Yığın Boyutu	Öğrenme Oranı	ESA Mimarisi	Sınıflandırma Doğruluğu
156.692	12.630	200	128	0.001	LeNet	%94,40
					AlexNet	%95,20
					GoogleNet	%97,40
					Resnet34	%98,10

Literatürde GTSRB veri kümesini kullanan bazı çalışmalar ile bu çalışmada önerilen modelin bir karşılaştırması Çizelge 4.7'de sunulmuştur.

Çizelge 4.7. GTSRB kullanan çalışmalarda elde edilen başarımların oranları

Takım	Method	Sınıflandırma Doğruluğu
Yin ve diğ.(2017)	Resnet	%99,67
Alvaro ve diğ. (2018)	ESA ile Uzaysal Dönüşüm ağıları	%99,21
Sermanet ve LeCun (2011)	Çok-Ölçekli ESA	%99,17
Xie ve diğ. (2016)	Seri bağlı ESA	%97,94
Sürücü ve Doğan (2018)	Hiyerarşik ESA	%97,80
Canyong (2018)	VGG-16	%96,00

GTSRB kullanılarak eğitilen SGDS'nin başarımı Türkiye karayollarından toplanmış Şekil 4.3'te verilen 18 trafik işaretleri üzerinde incelenmiştir. Başarım sonuçları Çizelge 4.8'de verilmiştir.



Şekil 4.3 Türkiye karayollarından toplanan trafik işaret örnekleri



Şekil 4.4 Kasisli yol işareti ve sistem tahmini

Test örnekleri içerisinde yer alan Şekil 4.4'teki kasisli yol trafik işareti 2 model tarafından doğru tahmin edilememiştir. GTSRB veri kümesinde yer alan “Bisiklet geçebilir” trafik işareti olduğu tahmin edilmiştir.

GTSRB veri kümesinde Şekil 4.5'te görüldüğü gibi metin içeriğine sahip STOP trafik işaretine ait imge örnekleri yer almaktadır.



Şekil 4.5. GTSRB veri kümesi ve Türkiye'deki STOP işareti

Türk karayollarında kullanılan DUR trafik işareti de STOP ile aynı geometrik şekle sahip ve metin içeren bir işarettir. Önerilen sistemin DUR işaretini doğru olarak sınıflandırdığı görülmektedir.

Şekil 4.6’da okul geçidi işareti incelendiğinde Türk karayollarında kullanılan işarettaki çocuk simgeleri ile GTSRB veri kümesinde yer alan işarettaki simgelerde farklılık olması ve test için kullanılan imgede yer alan trafik işaretinin tamamı olmamasına rağmen tüm modeller bu işareti doğru sınıflandırmışlardır.



Şekil 4.6. Türkiye’deki okul geçidi işareti ve GTSRB tahmini

Ayrıca Türk karayollarında bölgesel olarak farklılık gösteren Şekil 4.7 ve Şekil 4.8’de gösterilen hız sınırı değerleri bulunmaktadır. Test imgeleri içerisinde yer alan 45 ve 82 km/s hız limitleri GTSRB veri kümesinde yer almamaktadır. Tüm modeller bu trafik ışıklarını yüksek doğruluk oranıyla tahmin ettiğini belirtmektedir. Tahmin edilen Id etiketleri incelendiğinde 45 km/sa için 40km/sa ve 82 km/sa yerine 80 km/sa olarak tahmin ettiği gözlemlenmiştir.



Şekil 4.7. Türkiye’deki hız sınırı işareti (82km/sa) ve model tahminleri



Şekil 4.8 Türkiye’deki hız sınırı işareti (45km/sa) ve model tahminleri

Çizelge 4.8. Türkiye karayollarından toplanan trafik işaretleri için başarımların sonucu

Trafik işaretleri	Lenet	AlexNet	GoogleNet	ResNet	Tahmin edilen işaretler
Aşırı hız sınırlaması (30km/sa)	✓	✓	✓	✓	Aşırı hız sınırlaması (30km/sa)
İleri ve sola mecburi yön	✓	✓	✓	✓	İleri ve sola mecburi yön
Sola tehlikeli viraj	✓	✓	✓	✓	Sola tehlikeli viraj
Işıklı işaret cihazı	✓	✓	✓	✓	Işıklı işaret cihazı
Kasisli Yol	✗	✗	✓	✓	Bisiklet geçebilir
Kamyon giremez	✓	✓	✓	✓	Kamyon giremez
Ana yol-tali yol kavşağı	✓	✓	✓	✓	Ana yol-tali yol kavşağı
Dur	✓	✓	✓	✓	Dur
Aşırı hız sınırlaması (45km/sa)	✗	✗	✗	✗	Aşırı hız sınırlaması (40km/sa)
Ada etrafında dönünüz	✓	✓	✓	✓	Ada etrafında dönünüz
Aşırı hız sınırlaması (82km/sa)	✓	✓	✓	✓	Aşırı hız sınırlaması (80km/sa)
Yüksekliği 5 metreden fazla olan taşıt giremez	✗	✗	✗	✗	Aşırı hız sınırlaması (50km/sa)
Anayol	✓	✓	✓	✓	Anayol
Sağa dönülmez	✓	✓	✓	✓	Sağa dönülmez
Ana yol-tali yol kavşağı	✓	✓	✓	✓	Ana yol-tali yol kavşağı
Dikkat	✓	✓	✓	✓	Dikkat

5. SONUÇ

Bu tezde 4 farklı ESA mimarisi ile trafik işaretlerinin tanınması gerçekleştirilmiş ve mimarilerin başarımları oranları karşılaştırılarak en yüksek başarıma sahip ESA katman yapısına sahip bir SGDS önerilmiştir.

Önerilen SGDS’de kullanılan en yüksek başarıma sahip ESA katman yapısına deneysel olarak karar verilmiştir. ESA’nın eğitime parametreleri için en uygun değerler ızgara arama yöntemi ile belirlenmiştir.

Önerilen SGDS’nin başarımları GTSRB veri kümesi kullanılarak incelenmiştir. GTSRB eğitime veri kümesinde yer alan sınıflara ait imge örneklerindeki dengesizlik orijinal imgeler kullanılarak yeni imgeler türetilmesi yoluyla giderilmiştir.

Kullanılan ESA mimarilerinin katman yapılarındaki farklılıklar, kullanılan hassas parametreler modelin eğitime hızını ve başarımlarını büyük oranda etkilemiştir.

Yapılan deneyler sonucunda imge sınıflandırma problemlerinde kullanılan GPU’lerin CPU’lara göre çok hızlı işlem yapma kapasitesinin olduğu gözlemlenmiştir.

Yapılan deneyler sonucunda ResNet mimarisinden oluşturulmuş ESA modeli başarımları oranı en yüksek model olarak seçilmiştir ve önerilen SGDS’nin GTSRB test veri kümesi üzerinde %98,10 sınıflandırma doğruluğuna sahip olduğu gösterilmiştir. Ayrıca, GTSRB kullanılarak eğitilen SGDS’nin Türkiye karayollarında kullanılan 18 farklı trafik işareti üzerinde de başarılı sonuç verdiği gözlemlenmiştir.

Gelecek çalışmalar kapsamında Türk karayollarında kullanılan ve tüm trafik işaretlerini içeren bir imge kümesinin oluşturulması ve ESA temelli farklı mimarilerin bu küme üzerindeki başarımlarını incelemelerinin yapılması planlanmaktadır.

KAYNAKLAR

- Alvaro, A.G., Juan, A.A.G., Luis M. S.M. 2018.** Deep Neural Network for Traffic Sign Recognition Systems: An analysis of Spatial Transformers and Stochastic Optimization Methods, *Neural Networks*, 99:158-165.
- Anonim, 2017.** Trafik Kazaları Özeti 2017, Trafik Güvenliği Dairesi Başkanlığı Karayolları Genel Müdürlüğü, Ankara.
- Anonim, 2018.** Tensorflow kütüphanesi. <https://www.tensorflow.org/overview>, Erişim Tarihi: 10.05.2019,
- Anonim, 2019a.** Colab ortamının incelenmesi. <https://colab.research.google.com>, Erişim Tarihi: 11.06.2019.
- Anonim, 2019b.** Jetson Tx2 geliştirme kartı incelenmesi <https://nvidia.com>, Erişim Tarihi: 17.02.2019.
- Canyong, W. 2018.** Research and Application of Traffic Sign Detection and Recognition Based on Deep Learning. *International Conference on Robots & Intelligent Systems (ICRIS)*, 150-152.
- Canziani, A., Paszke, A., Culurciello, E. 2017.** An Analysis of Deep Neural Network Models for Practical Applications. ArXiv, 1-7
- Goodfellow, I., Bengio, Y., Courville, A. 2016.** Deep Learning. *MIT Press*, 330 pp.
- He, K., Zhang, X., Ren, S., Sun, J. 2016.** Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.
- Hossain, S., Hyder, Z. 2015.** Traffic Road Sign Detection and Recognition for Automotive Vehicles. *International Journal of Computer Applications*, 120(24),0975-8887.
- Hussain, S., Abualkibash, M., Tout, S. 2018.** A Survey of Traffic Sign Recognition Systems Based on Convolutional Neural Networks. *2018 IEEE International Conference on Electro/Information Technology*, 0570-0573.
- İnik, Ö., Ülker, E. 2017.** Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri. *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6(3): 85-104.
- Kılıkçier, Ç., Yılmaz, E. 2018.** Trafik Işığı Tespiti Yapan Bir Sürücü Güvenlik Destek Sistemi. *Politeknik Dergisi*, 21(2): 419-426.
- Kırmızıoğlu, E., Yaman, H. T. 2011.** Analysis of Comprehension of Traffic Signs: A Pilot Study In Ankara, Turkey. *MSc Thesis*, Civil Engineering Department, Middle East Technical University, Ankara
- Kingma, D., Ba, J. 2015.** Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*
- Kiran C. G., Lekhesh V. P., Abdu Rahiman V., Rajeev K. 2009.** Traffic Sign Detection and Pattern Recognition Using Support Vector Machine, *Seventh International Conference on Advances in Pattern Recognition*, 87-90.
- Krizhevsky, A., Sutskever, I., Hinton, E., Geoffrey. 2012.** ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*, 1:1097-1105.
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. 1998.** Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

- Liu, S., Deng, W. 2015.** Very deep convolutional neural network based image classification using small training sample size. *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, 730-734
- Murat, Y.S, Cakici, Z. 2017.** Trafik İşaretlerinin Bilinirliği Üzerine Bir Araştırma: Denizli Örneği. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*
- Ramadhan, S., Ergen, B. 2017** Traffic Sign Detection and Recognition. *International Journal of Advanced Research in Electrical, Electronics, Instrumentation Engineering*, 957-963.
- Sermanet, P., LeCun, Y., 2011.** Traffic Sign Recognition with Multi-Scale Convolutional Networks. *Neural Networks(IJCNN) The International Joint Conference on*, 2809–2813.
- Stallkamp J., Schlipsing, M., Salmen, J., Igel, C. 2011.** The German Traffic Sign Recognition Benchmark: A multi-class classification competition. *In International Joint Conference on Neural Networks*, 139:1453-1460.
- Stallkamp, J., Schlipsinga, M, Salmen, J., Igel, C. 2012** Man vs. Computer: Benchmarking Machine Learning Algorithms for Traffic Sign Recognition. *Neural Networks*, 32:323-32.
- Sürücü, E., Doğan, H. 2018.** Traffic Sign Recognition with Hierarchical Convolutional Neural Network. *Signal Processing and Communications Applications Conference (SIU)*, 26(1):1-4.
- Szegedy,C., Liu,W. Jia, Y. , Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. 2015.** Going Deeper with Convolutions, *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*
- Wang, Y., Shi, M., Wu, T. 2009** A Method of Fast and Robust for Traffic Sign Recognition. *Fifth International Conference on Image and Graphics*, 891 – 895.
- Xie, K., Ge, S., Ye, Q., Luo, Z. 2016.** Traffic Sign Recognition Based on Attribute-Refinement Cascaded Convolutional Neural Networks. *17th Pacific-Rim Conference on Multimedia*, 201-210.
- Yin, S., Deng, J., Zhang, D. 2017.** Traffic sign recognition based on deep convolutional neural network. *Optoelectronics Letters*, 13(6):476-480.

ÖZGEÇMİŞ

Adı Soyadı : Mehmet ZAM
Doğum Yeri ve Tarihi : Konak, 1990
Yabancı Dil : İngilizce, İtalyanca

Eğitim Durumu
Lise : İzmir 60.Yıl Anadolu Lisesi, 2008
Lisans : Uludağ Üniversitesi Elektronik Mühendisliği, 2013

Çalıştığı Kurum/Kurumlar : TOFAŞ Türk Otomobil Fabrikası A.Ş.

İletişim (e-posta) : mehmetzam@outlook.com

Yayınları :

Zam, M., Yılmaz, E. 2019. Trafik İşaretlerini Tanıyan Evrişimsel Sinir Ağları Temelli Bir Sürücü Güvenlik Destek Sistemi. *Uludağ University Journal of The Faculty of Engineering.* (Gönderildi.)