



**DERİN ÖĞRENME YÖNTEMLERİ İLE TÜRK İŞARET DİLİNİN
ÖĞRENİLMESİ**

Hazırlayan: Melek ALAFTEKİN

Danışman: Doç.Dr. Kenan ÇİÇEK

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI

Yüksek Lisans Tezi

IĞDIR/2022

Her Hakkı Saklıdır

T.C.
İĞDIR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

DERİN ÖĞRENME YÖNTEMLERİ İLE TÜRK İŞARET DİLİNİN
ÖĞRENİLMESİ

Melek ALAFTEKİN

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI

İĞDIR/2022

Her Hakkı Saklıdır

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını beyan ederim.

Melek ALAFTEKİN



Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ÖZET

DERİN ÖĞRENME YÖNTEMLERİ ile TÜRK İŞARET DİLİNİN ÖĞRENİLMESİ

ALAFTEKİN, Melek

Yüksek Lisans Tezi

Elektrik Elektronik Mühendisliği Ana Bilim Dalı

Tez Danışmanı: Doç. Dr. Kenan ÇİÇEK

Haziran 2022, 75 Sayfa

İşaret dili, sağır ve dilsiz bireylerin duygu ve düşüncelerini çevrelerine daha rahat bir şekilde iletmek için kullandıkları bir iletişim aracıdır. Sesli iletişimden farklı olarak işaret dili, parmak, el, kol, vücut duruşu, jest ve mimik gibi vücudun farklı bölgelerinin kullanıldığı ve her birinin belirli bir anlam ifade ettiği sözsüz bir iletişim şekline sahiptir. İşaret dili, sağır ve dilsiz bireyler ile toplumun geri kalan bireyleri arasındaki iletişimde kilit bir role sahiptir. Normal insanlar arasında işaret dili çok yaygın bilinen bir iletişim dili değildir. Bu yüzden, engelli bireylerin günümüz dünyasına ayak uydurmaları için genellikle bir tercümana ihtiyaçları vardır. Günümüzün hızla gelişen teknoloji dünyasında insan-bilgisayar etkileşiminin gelişmesiyle birlikte, böyle bir gereksinime olan ihtiyacı karşılamak adına araştırmacılar tarafından birçok işaret dili çeviri sistemleri geliştirilmiştir. Bu çalışmada, Türk işaret dili çeviri sisteminde gerçek zamanlı el hareketi tanıma üzerine odaklanılmıştır. Gerçek zamanlı ve yüksek performanslı bir algılama sağlamak için son teknoloji nesne algılama algoritması olan Evrişimli Sinir Ağı (ESA) tabanlı YOLOv4-CSP kullanılmıştır. YOLOv4-CSP algoritması, ağın performansını iyileştirmek için orijinal YOLOv4'ün boyun kısmına CSPNet eklenmesiyle oluşturulmuştur. YOLOv3, YOLOv3-SPP ve YOLOv4-CSP modelleri, Türk İşaret dilindeki rakamlardan oluşan etiketli veri seti ile eğitilerek yapılan el işaretlerinin algılanması üzerindeki performansları karşılaştırılmıştır. Önerilen yöntem ile 9.8 milisaniyede kesinlik %98,84, geri çağırma %97,27, F1-skor puanı 98,04 ve mAP %99,32 sonuçlar elde edilmiştir. Türk işaret dilinde önerilen yöntem, arka plandan bağımsız olarak hem gerçek zamanlı performans hem de doğru el işaretini tahmin etmesi ile diğer algoritmalarından daha iyi performans göstermiştir.

Anahtar kelimeler: Türk İşaret Dili, Evrişimli Sinir Ağları (ESA), YOLOv4, YOLOv4-CSP

ABSTRACT

TURKISH SIGN LANGUAGE RECOGNITION with DEEP LEARNING METHODS

ALAFTEKIN, Melek

Master's Thesis

Electrical and Electronics Engineering Department

Thesis Advisor: Assoc. Dr. Kenan ÇIÇEK

June 2022, 75 Pages

Sign language is a communication tool that deaf and dumb people use to convey their feelings, thoughts, desires, etc. to the outside of their world. Unlike verbal communication, sign language has a form of nonverbal communication in which different parts of the body such as fingers, hands, arms, body postures, gestures and facial expressions are used and each of them has a certain meaning. Sign language has a key role in communication between deaf and dumb individuals and the rest of the society. Sign language is not a widely known communication language among normal people. Therefore, deaf and dumb people often need an interpreter to interact with today's world. With the development of human-computer interaction in today's rapidly developing technology world, many sign language recognition systems have been developed by researchers in order to meet the need for such a requirement. This study focuses on real-time hand gesture recognition in Turkish sign language recognition system. YOLOv4-CSP based on Convolutional Neural Network (CNN), a state-of-the-art object detection algorithm, was used to provide real-time and a high-performance detection. The YOLOv4-CSP algorithm was created by adding CSPNet to the neck of the original YOLOv4 to improve network performance. YOLOv3, YOLOv3-SPP and YOLOv4-CSP models were trained with a labeled data set consisting of numbers in Turkish Sign language, and their performances on the hand signals recognition were compared. With the proposed method, 98,84% accuracy, 97.27% recall, 98.04 F1-score score and 99,32% mAP results were obtained in 9.8 milliseconds. The proposed method for detecting numbers in Turkish sign language outperformed other algorithms with both real-time performance and accurate hand sign prediction, regardless of background.

Key words: Turkish Sign Language, Convolutional Neural Network (CNN), YOLOv4, YOLOv4-CSP

TEŞEKKÜR

Yüksek lisans tezim boyunca desteklerini hiç eksik etmeyen ve değerli zamanlarını ayırarak bana yol gösteren çok değerli danışman hocam Sayın Doç. Dr. Kenan ÇİÇEK'e ve Arş. Gör. İshak PAÇAL'a en içten teşekkürlerimi sunarım.

Ayrıca bu tez çalışmasında kullandığım veri setini hazırlamamda bana yardımcı olan Iğdır Üniversitesi Mühendislik Fakültesi değerli hocalarına ve eğitim hayatım boyunca bana maddi manevi destek veren annem Nahide ALAFTEKİN ve babam Mehmet ALAFTEKİN'e saygı ve teşekkürlerimi sunuyorum.

Melek ALAFTEKİN

IĞDIR/2022

İÇİNDEKİLER

Sayfa No

ÖZET.....	i
ABSTRACT	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER.....	iv
KISALTMALAR DİZİNİ.....	vi
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ	ix
1. GİRİŞ.....	1
2. LİTERATÜR TARAMASI	3
3. MATERYAL ve METOT	11
3.1. Türk İşaret Dili (TİD).....	11
3.2. Derin Öğrenme	13
3.3. Evrişimli Sinir Ağları (ESA)	19
3.3.1. Evrişimli sinir ağları katmanları	20
3.3.1.1. Evrişim katmanı (convolution layer)	20
3.3.1.2. Aktivasyon katmanı (activation layer).....	23
3.3.1.3. Havuzlama katmanı (pooling layer).....	24
3.3.1.4. Tam bağlı katman (fully connected layer-FC).....	25
3.4. Nesne Algılama	26
3.4.1. Nesne algılamanın tarihsel gelişimi.....	27
3.4.2. Evrişimli sinir ağı tabanlı nesne algılama algoritmaları	29
3.4.2.1. Çift aşamalı nesne algılama	31
3.4.2.2. Tek aşamalı nesne algılama	32
3.5. YOLO - You Only Look Once	32
3.5.1. YOLOv4 (YOLO versiyon 4).....	35
3.5.1.1. Aktivasyon Fonksiyonu	41
3.5.1.2. Kayıp Fonksiyonu	43

3.6. Türk İşaret Dilinin Makine Öğrenmesi ile Tespiti	46
3.6.1. Veri seti.....	46
3.6.2. Veri etiketleme.....	47
3.6.3. Veri arttırma.....	50
3.6.4. Değerlendirme metrikleri ve uygulama detayları	52
3.6.4.1. Deneysel kurulum	52
3.6.4.2. El işareti algılama ve metrikleri değerlendirme.....	52
4. BULGULAR ve TARTIŞMA.....	55
4.1. Deneysel Sonuçlar	55
5. SONUÇ ve ÖNERİLER.....	64
KAYNAKÇA	65
ÖZGEÇMİŞ.....	76

KISALTMALAR DİZİNİ

2B	İki Boyutlu
3B	Üç Boyutlu
ANFIS	Adaptive Neuro-Fuzzy Inference System
CIoU	Complete Intersection over Union
CNN	Convolutional Neural Network
sEGM	Surface Electromyography
HMM	Hidden Markov Model
GPU	Graphics Processing Units
IMU	Inertial Measurement Unit
IoU	Intersection over Union
k-NN	k-Nearest Neighbors
mAP	Mean Average Precision
R-CNN	Regions with Convolutional Neural Networks
RoI	Region of Interest
GMM	Gaussian mixture model
RGB	Red-Green-Blue

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 3.1. Hamidiye Ticaret Mektebi önünde Türk İşaret Diliyle “Padişahım Çok Yaşa” diyen çocuklar (Abdullah Frères, Servet-i Fünun dergisinin 129. sayısı).....	12
Şekil 3.2. Hareket ifadesi ile kelime ‘VERMEMEK’	13
Şekil 3.3. Parmak alfabesi ile 13 sayısının yazımı.....	13
Şekil 3.4. Derin öğrenmenin tarihsel gelişimi	14
Şekil 3.5. Geleneksel makine öğrenimi ve derin öğrenme	16
Şekil 3.6. a) Eksik öğrenme, b) İstenilen öğrenme ve c) Aşırı öğrenme grafikleri	18
Şekil 3.7. Evrişimli sinir ağı mimarisi (Mathwork).....	19
Şekil 3.8. Evrişim katmanı.....	21
Şekil 3.9. 5x5 boyutta 2 boyutlu bir giriş görüntüsüne 3x3’lük filtrenin uygulanması ..	21
Şekil 3.10. ReLu aktivasyon grafiği	24
Şekil 3.11. 2x2 filtre ile iki adım kaymalı maksimum havuzlama işlemi.....	25
Şekil 3.12. Tam bağlantılı katman	25
Şekil 3.13. Evrişimli sinir ağları çeşitleri.....	26
Şekil 3.14. Nesne algılama ile otomobil tespiti	27
Şekil 3.15. Nesne algılamanın tarihsel gelişimi.....	28
Şekil 3.16. Geleneksel nesne tespiti akış diyagramı	29
Şekil 3.17. Nesne algılama algoritmalarının genel mimarisi.....	30
Şekil 3.18. YOLO Birleşik Algılama Modeli	33
Şekil 3.19. YOLO mimarisi	33
Şekil 3.20. YOLOv4 Mimarisi	36
Şekil 3.21. CSPn yapısı.....	36
Şekil 3.22. YOLOv4-CSP (Scaled YOLOv4)	41
Şekil 3.23. Mish ve Sızdıran ReLu aktivasyon fonksiyon grafikleri.....	42
Şekil 3.24. Sistem akış diyagramı.....	46
Şekil 3.25. Türk işaret dili rakamları	47
Şekil 3.26. Türk işaret dilinde 9 rakamına ait farklı el işaretleri	47
Şekil 3.27. LabelImg ara yüzü	48

Şekil 3.28. LabelImg ara yüzündeki görüntü.....	48
Şekil 3.29. LabelImg'de verinin etiketlenmesi.....	49
Şekil 3.30. Görüntülerin .txt dosyaları.....	50
Şekil 3.31. Veri artırma teknikleri	51
Şekil 3.32. Cutmix veri artırma tekniği	51
Şekil 3.33. Mozaik Veri Artırma Tekniği.....	52
Şekil 4.1.. Örnek etiketli veri seti	56
Şekil 4.2. Tahmin değerleri verilen örnek veri seti.....	57
Şekil 4.3. YOLOv4-CSP kesinlik-geri çağırma grafiği	59

ÇİZELGELER DİZİNİ

	Sayfa No
Çizelge 3.1. R-CNN serisi ve özellikleri	31
Çizelge 3.2. BoF ve BoS içerikleri	38
Çizelge 4.1. Türk işaret dili veri seti için sıfırdan eğitilmiş modeller ve sonuçları	58
Çizelge 4.2. İşaret dili ile ilgili yapılmış güncel çalışmalar	60



1. GİRİŞ

İşaret dili, sağır ve dilsiz bireylerin duygu ve düşüncelerini çevrelerine daha rahat bir şekilde iletmek için kullandıkları bir iletişim aracıdır. Normal bireylerin kullandığı sesli iletişimden farklı olarak işaret dili parmak, el, kol, vücut duruşu, jest ve mimik gibi vücudun farklı bölgelerinin kullanıldığı ve her birinin belirli bir anlam ifade ettiği sözsüz bir iletişim şekline sahiptir. Dünya sağlık örgütünün 2021 yılı verilerine göre, 34 milyonu çocuk olmak üzere dünya genelinde yaklaşık 430 milyon işitme kaybı yaşayan insan bulunmaktadır. Bu sayının 2050 yılına kadar 700 milyonu aşacağı tahmin edilmektedir (Word Health Organization, 2021). Ülkemizde ise işitme ve konuşma yetersizliği yaşayan bireylerin sayısı 228 bin 589'dır (Aile ve Sosyal Politikalar Bakanlığı, 2021). Bu oranlar, dünyada yaşayan sağır ve dilsiz bireylerin yaşadıkları sıkıntıların ne kadar ciddi boyutta olduğunu gözler önüne seriyor. İşaret dili, işitme engelliler ve toplumun geri kalanı için iletişimde kilit bir role sahiptir. Normal insanlar arasında işaret dili yaygın olmasa da yorumlayabilen çok az insan vardır. Bu durum sağır ve dilsiz bireyler ile normal bireyler arasında iletişim engeli oluşturarak alışveriş yapma, resmi işlemlerde, okulda, kişisel ihtiyaçlarını giderme gibi sosyal yaşamda kendilerini ifade etmede zorluklar yaşamalarına sebep olur. Bu yüzden, sağır ve dilsiz bireyler ile dünyanın geri kalanı arasında ortak bir iletişim tabanı oluşturmak amacıyla, işaret dilindeki işaretleri takip eden, tanımlayan ve bunları anlamlı harf, sayı, kelime ve ifadelere dönüştüren işaret dili tanıma platformları geliştirilmiştir.

İşaret dili tanıma, görme tabanlı ve eldiven tabanlı olmak üzere iki temel algılama tekniğine dayanmaktadır (Elakkiya, 2020). Eldiven tabanlı tekniklerde, el ve parmakların hareketini, konumunu, yönünü ve hızını yakalamak için kullanıcıya bağlanan bir dizi sensor kullanılır. Bu teknikte araştırmacılar veri eldivenleri (Shukor *et al.*, 2015), microsoft kinect (Ren *et al.*, 2013), sıçrama hareket kontrolörü (Naglot and Kulkarni, 2016), derinlik kamerası (Almeida *et al.*, 2014), yüzey elektromiyografisi (sEMG) sinyal kol halkaları (Khomami and Shamekhi, 2021; Tateno *et al.*, 2020) ve atalet ölçüm birimi (IMU) (Khomami and Shamekhi, 2021) gibi çeşitli sensörlere çalışmalarında yer vermişlerdir. Bu çalışmalar ile mükemmel sonuçlar elde edildiği halde, pahalı olması ve günlük yaşam kullanımına uygun olmadığı için pek tercih

edilmemektedir. Öte yandan, görüntü tabanlı tekniklerde ise tek bir kamera yardımıyla alınan el hareketlerinin görüntüleri kullanılarak işaret tanınması yapılır. Görme tabanlı teknikler daha kolay olmakla birlikte diğer tekniklere göre nispeten daha düşük hesaplama maliyetine sahiptir. Görme tabanlı teknikler ilgili el işaretini belirlemek için segmentasyon, kontur ve sınır modelleme, görüntü özellik vektörü gibi işlemler kullanılarak görüntü işleme ve /veya sinyal işleme gerçekleştirilir. Daha sonra, elde edilen veriler, işaretin yorumlanması için saklanan etiketli veri seti ile karşılaştırılarak belirli olasılıklar hesaplanır. Bu olasılıklara bağlı olarak işaretin sınıflandırılması yapılır.

Son yıllarda, makine öğreniminin bir alt kümesi olan derin öğrenme, sağlam ve gerçek zamanlı bir işlem için görme tabanlı işaret dili tanımada popüler hale gelmiştir (Rastgoo *et al.*, 2021). Derin öğrenme, görüntü sınıflandırma, nesne algılama, görüntü elde etme, anlamsal bölütleme ve insan pozu tahmini gibi alanlarda yüksek performans göstererek işaret dili çalışmalarında büyük bir önem ve güvenilirlik kazanmıştır (Guo *et al.*, 2016). Ayrıca görüntü işlemede oldukça gerekli olan büyük veri kümelerinin işlenebilir olması derin öğrenme tekniğinin bir diğer avantajıdır. Görme tabanlı işaret dili tanımada derin öğrenmenin temel amacı, ilgili el işaretini, yakalanan görüntüdeki el işareti ile karşılaştırarak aralarındaki benzerlik oranına göre doğru bir sınıflandırma yapmaktır (Nimisha and Jacob., 2020). Bu nedenle, işaret dili tanıma sistemlerinde yapılan araştırmaların çoğu el modelinin arka plandan ayrılması üzerine olmuştur.

Bu çalışmada, YOLO (You Only Look Once) (Redmon *et al.*, 2016) nesne algılama algoritmasının 4. Versiyonu olan YOLOv4 (Bochkovskiy *et al.*, 2020) ve CSPNet (Wang *et al.*, 2020) ile oluşturulan YOLOv4-CSP (Wang *et al.*, 2021) kullanarak herhangi bir görüntü ön işlem (segmentasyon, kontur belirleme vb.) ve algılama için tek tip bir arka plana ihtiyaç duyulmadan ilgili el hareketini tanımak için yeni bir yöntem önerilmektedir. Önerdiğimiz bu yöntem, yaptığımız literatür taraması sonucunda Türk işaret dilinde ilk kez kullanıldı. Önerilen yöntemin doğruluğunun test edilmesi için Türk İşaret diline ait rakamlar kullanılmıştır. Bu yöntemde, bir videodan nesne olarak kabul edilen el hareketlerini algılayarak gerçek zamanlı nesne algılama gerçekleştirilir.

2. LİTERATÜR TARAMASI

İletişim dilinin farklı bir biçimi olan işaret dili, normal insanlar arasında çok yaygın olmaması, sadece Türk toplumunda değil dünya genelinde geniş insan toplulukları için de önemli bir sorundur. Son yıllarda ilerleyen teknolojiyle birlikte insan bilgisayar etkileşiminin iyice artması işaret dili tanıma sistemlerini aktif araştırma konusu haline getirmiştir. İşaret dili, görsel hareketler ve işaretler içeren el hareketlerinin ve yüz ifadelerinin yapısal bir biçimi olması bilgisayarlı görme algoritmaları için uygun bir test-yazılım yatağı haline getirmiştir (Wu and Huang, 1999). Derin öğrenmenin popüler olması ile birlikte geliştirilen işaret ve jest tanıma sistemleri ile işaret dili kullanıcılarının iletişim engellerinin kaldırılmasına büyük oranda fayda sağlaması ve sağlıklı, rahat bir iletişim kurmalarına yardımcı olması hedeflenmektedir. Bu alanda, yapılan ilk çalışmalardan biri, bir eldivende manyetik akı sensörleri kullanarak el pozisyonunu ve oryantasyonunu tahmin etmeye çalışan Zimmerman (1987) tarafından önerilen bir el hareketi ara yüzüne dayanmaktadır.

Starner *et al.*, (1998), gizli markov modellerini (Hidden Markov Models-HMMs) kullanarak Amerikan İşaret Dili (AİD) tanıma için iki görüntü tabanlı sistem üzerinde çalışmışlardır. İlk yaklaşımlarında bir masanın üzerine yerleştirdikleri kamera yardımıyla ve ikinci yaklaşımlarında ise kullanıcı tarafından giyilen bir başlık üzerine monte ettikleri bir kamera yardımıyla ilgili el işaretini aldılar. Yaptıkları çalışmalarda 40 işaret test edildi. İlk yaklaşımlarında %92 ve ikinci yaklaşımlarında ise %98'lik bir doğruluk payına ulaşılar.

Bauer and Hienz (2000), Alman İşaret Dili'nde (AlİD) 97 işaretin tanınmasını sağlayan video tabanlı sürekli bir işaret dili tanıma sistemi geliştirdiler. İşaretlerin tek renkli bir video kamera ile alındığı çalışmada, her işaret için sürekli yoğunluklu HMMs kullandılar. Sistem girdileri için manuel işaretlerinin parametrelerini (el şekli, yönelim ve konum gibi) içeren öznitelik vektörleri kullanıldı. Sürekli işaret tanıma için de hesaplama karmaşıklığını ortadan kaldırmak için ışın arama algoritması kullanarak geliştirdikleri sistemde %91,7'lik bir tanıma doğruluğu elde ettiler.

Al-Jarrah and Halawani (2001), Arapça işaret dilinde manuel alfabelerin hareketlerinin otomatik çevirisine yönelik bir çalışma önermişlerdir. Geliştirdikleri

sistemde, herhangi bir eldiven ya da cihaz kullanmasına ihtiyaç duyulmadan bir kamera yardımıyla alınan el görüntülerini ön işlemden geçirdikten sonra parmak uçları bölgesini kapsayacak şekilde alınan hareket alan merkezi ile hareket sınırı arasındaki hesaplanan vektör uzunluklarından oluşan öznitelik şemasından yararlandılar. El hareketlerinin konumu, boyutu ve/veya yönündeki değişikliklerinden etkilenmeyen öznitelik vektörleri, tanıma işlemini gerçekleştirmek için uyarlanabilir nöro-bulanık çıkarım sisteminde (Adaptive Neuro-Fuzzy Inference System-ANFIS) eğitim modelinde kullandılar. Belirli bir bulanık çıkarım sisteminin parametrelerini optimize etmek için kullanılan eğitim modelinde çıkarmalı kümeleme algoritması ve en küçük kareler tahmincisi kullanıldı. Yapılan deneyler de 30 Arapça manuel alfabesini %93,55 doğrulukla tanıyabildiğini ortaya koydular.

Wang *et al.*, (2002), fonemleri kullanarak geniş bir kelime dağarcığı ile Çince işaret dili tanımaya yönelik bir yaklaşım sunmuşlardır. Yaptıkları çalışmada yaklaşık 2.400 ses birimi kullandılar. Her ses birimi için HMM tabanlı eğitim modeli ve Çince işaretleri bu ses birimlerine göre kodlayan bir kodlayıcı kullandılar. Sistemde Gauss kümelenmesi, Biagram dil modeli, iyi bir tanıma doğruluğu için n-best pass algoritması ve arama alanını azaltmak için ağaç yapılı bir ağ kullanan bir kod çözücü sundular. Sundukları yaklaşım ile 5.119 işaret sözlüğü üzerinde etkileyici sonuçlar elde etmiş olsalar da farklı ses birimleri için ayrı bir uyarlama isteyen işaretleyiciden bağımsız olmayan bir yaklaşım elde etmişlerdir.

Chen *et al.*, (2003), tek el hareketlerinin tanınmasını sağlayan bir sistem geliştirdiler. Durağan bir arka plan kullanarak geliştirdikleri bu sistem, gerçek zamanlı el izleme ve segmentasyonu, öznitelik çıkarma, HMM eğitimi ve ilgili el hareketinin tanımlanması olmak üzere art arda gerçekleşen dört bölümden oluşmaktadır. Sistem, ilgili el bölgesinin arka plandan ayrılmasını sağlayan bir segmentasyon ve bunun takibini sağlayan gerçek zamanlı bir izleme algoritmasına sahiptir. Fourier tanımlayıcısı (Fourier Descriptor) kullanarak uzaysal özelliklerini, hareket analizini kullanarak da zamansal özelliklerini karakterize ederek görüntünün özneteliğini elde ettiler. Yapılan el hareketinin ilgili el hareketi ile benzerliğinin bulunması ve sınıflandırılması için HMM eğitim modelini sundular. Farklı HMM'ler uygulayarak her bir hareket için bir puanlama sırası elde edilerek en yüksek puanlama değerine sahip olan model, ilgili el

hareketinin aranan cevabı olarak değerlendirilmiştir. Geliştirdikleri bu sistem ile 20 farklı el hareketinin tanıtılmasını sağlayarak %90'ın üzerinde bir başarı elde ettiler.

Brashear *et al.*, (2006), küçük sağır çocukların Amerikan İşaret Dili (AİD) becerilerini geliştirmeye ve dil gelişimlerini teşvik etmeye yönelik jest tanıma teknolojisi kullanarak CopyCat adında bir bilgisayar oyunu geliştirdiler. Oyunun veri seti tabanı, Atlanta Bölgesi Sağır Okulu'ndaki sağır çocukların Oz Büyücüsü oyunun versiyonunu oynayan kullanıcı çalışmalarından toplanmıştır. Sürekli işaretleme, giyim ve cilt tonları dahil olmak üzere çeşitli özelliklerden oluşturulan veri seti, beş çocuk kullanıcı tarafından 22 kelimeliden oluşan oyun cümleleri 541 cümle örneği ve 1.959 bireysel işaret örneğinden oluşmaktadır. Oyunun çalışma prensibi, sağlam el segmentasyonu ve takibi için renk histogramına dayanmaktadır. Çocukların bileklerinin arkasına monte edilmiş kablosuz ivme ölçerli küçük renkli eldiven ile elde edilen el hareketleri bilgisini tanımak için HMM kullandılar. Deneye katılan dört çocuğun verilerini sistem üzerinden eğitiler ve beşinci çocuğun verilerini de test ederek kullanıcıdan bağımsız modeller oluşturdular. Yapılan çalışmalar sonucunda çocuk başına ortalama %91,75 ile %73,73 arasında değişen kelime doğruluğu elde ettiler.

Fang *et al.*, (2007), Çin İşaret Dili'nde (ÇİD) 5.113 işaretten oluşan geniş kelime dağarcığına sahip bir sözlükte iki bitişik işaret arasındaki geçişleri işlemek için sürekli işaret dili tanımada Geçiş Hareketi Modellerini (GHM'ler) önerdiler. Geniş bir kelime dağarcığından kaynaklanan kitlesel geçiş hareketlerinin üstesinden gelmek için dinamik zaman atlama kullanılarak k-ortalamalardan geliştirilmiş bir zamansal kümeleme algoritması önerdiler. Tekrarlanan bir bölütleme algoritması ile sürekli cümlelerin geçiş parçalarının otomatik bölütlenmesini sağlayarak Geçiş Hareketi Modelini önyükleme işlemi ile eğittiler. Kümelenmiş GHM'ler yapıları iyi genellemeler sayesinde geniş kelime dağarcığına sahip sürekli işaret dilini tanımada kullanıma uygun ve işlevsel olarak bir Viberto algoritması yerini alabileceğini öne sürdüler. Yapılan deneyler sonucunda sürekli işaret dilinde, 5.113 Çince işaretten oluşan bir sözlükten 1.500 test cümlesi örneğinde ortalama %91,9'luk bir doğruluk oranı elde ettiler.

Han *et al.*, (2009), sürekli el işareti hareketlerinin bir alt birim segmentasyon algoritmasını geliştirerek bir işaret dili tanıma yaklaşımı sundular. Segmentlerin sınırlarını belirlemek için el hareketi zamansal süreksizliğini kullandılar. Son olarak,

benzer alt birimdeki segmentlerin birleştirilmesini sağlayıp daha doğru sonuçlar elde etmek için Dinamik Zaman Atlama (Dynamic Timewarping - DTW) mesafe ölçümleri ve uzaysal-zamansal kümeleme kullanmışlardır. El hareketlerinin alt birimlerine ayırma fikrinin işaret dili biliminde heceleme modellerinde kullanılarak daha doğru sonuçlara ulaşılabileceğini savunulmuşlardır.

Zafrulla *et al.*, (2011) Amerikan İşaret Dili'nde (AİD) işitme engelli çocuklar için geliştirdikleri eğitici oyunlarda kullanılmak üzere bir yaklaşım önerdiler. Kinect derinlik kamerası kullanarak mevcut kullandıkları CopyCat sistemlerinde eldiven ve sensör ihtiyacını azaltacak yönde karşılaştırmalı bir çalışma sundular. Çalışmalarını ayakta durma ve oturma olmak üzere iki farklı pozisyonda değerlendirdiler. 1.000 kelimelik Amerikan işaret dilinde kinect kamerada kullanıcılar (yetişkin) otururken ve ayakta dururken sırasıyla %51,5 ve %76,12 doğrulama, CopyCat sisteminde ise oturarak %74,82'lik bir doğrulama elde ettiler. Kinect kamera sistemi oturarak kullanım için daha fazla geliştirilmesi gerekse de elde ettikleri sonuçlar Kinect kameranın işaret doğrulama için uygun bir seçenek olabileceği kanısına vardılar.

Memiş ve Albayrak (2013), Türk İşaret Dili'nde (TİD) kinect RGB video dizilerindeki uzamsal-zamansal özelliklerinin kullanıldığı bir tanıma sistem sundular. Önerilen sistemde, ardışık videolardaki dinamik işaretlerin zamansal özelliklerini ifade etmek için kümülatif hareketli görüntüler yaklaşımı, hareketlerin uzamsal özellikleri elde etmek içinde kümülatif işaret görüntülerine 2-B ayrık kosinüs dönüşümü (Discrete Cosine Transform-DCT) uyguladılar. Manhattan mesafeli k-NN sınıflandırıcısı kullanarak tanıma işlemini gerçekleştirdiler. Türk İşaret Dilinin üç farklı kategorisinde 111 kelimeye ait 1.002 işaretin yer aldığı bir işaret veri tabanında önerdikleri sistem yaklaşık %90 bir tanıma oranına sahiptir.

Huang *et al.*, (2015), işaret dili tanıma sistemlerinin karmaşık ve büyük olmasından dolayı kullanılan mevcut yöntemlerin işaret dilinin doğru ve güvenilir bir şekilde çevrilmesinde yaşanan zorlukların üstesinden gelmek için yeni bir 3B Evrişimli Sinir Ağı (ESA) önerdiler. Evrişimli Sinir Ağı, ham videodan hem uzamsal hem de zamansal özellikleri herhangi bir ön bilgi olmadan otomatik olarak çıkarmak için kullandılar. Sistem performansını artırmak için renk bilgisi, derinlik ve vücut eklem konumları dahil olmak üzere çok kanallı video akışları renk, derinlik ve yörünge bilgileri

birleştirilerek 3B ESA'ya girdi olarak kullanarak nihai sonuca ulaşmaya çalıştılar. Önerilen model microsoft kinect ile derlenen bir veri kümesi hem 3D ESA hem de GMM-HMM'yi (Gaussian Mixture Model-Hidden Markov Model) değerlendirilerek yöntemin etkinliği gösterdiler.

Mao *et al.*, (2017), evrişimli sinir ağı ve tekrarlayan sinir ağına dayalı uzun-kısa süreli bellek (Long Short Term Memory- LSTM) yöntemi içeren ve bir diziden diziye çeviren kodlayıcı-kod çözücü tabanlı bir çerçeve önererek Çince İşaret Dili'nde (ÇİD) bir çalışma sundular. Önerilen çerçeve, özellik çıkarma, kodlayıcı-kod çözücü modelleme ve model birleştirme olmak üzere üç modülden oluşmaktadır. ESA, görüntü dizisi özelliklerini çıkarmak için kullanılırken uçtan uca zamansal modeli oluşturmak için bir kodlayıcı-kod çözücü ağı için iki LSTM katmanı basamaklandırılarak kullanılmıştır. Kodlayıcı-kod çözücü, giriş özelliklerinin zamansal bilgilerini ve işaret dili sözcüklerinin bağlam modelini de öğrenmeye yardımcı olur. Microsoft Kinect 2.0 veri seti ile kodlayıcı-kod çözücü çerçevesinde girdi olarak iskelet koordinatlarını kullanarak oluşturdukları başka bir model ile kombinasyon yaparak nihai tahmin sonucuna ulaşmaya çalışarak Çince İşaret Dilinde zamansal bilgi ve bağlam ilişkisini öğrenmeyi hedefleyen bir yöntem önerdiler.

Moghaddam *et al.*, (2020), hibrit çekirdek temel bileşen analizi ve hibrit çekirdek diskriminant analizinden oluşan değiştirilmiş doğrusal olmayan çekirdek tabanlı hızlı özellik çıkarma yöntemlerine dayalı bir yöntem önererek Farsça İşaret Dili'ndeki (FİD) statik ve dinamik işaretlerin tanınmasına amaçladılar. Dinamik işaretler, öznelik çıkarma yöntemi ve uzay-zaman yaklaşımı birlikte kullanılarak tanınmıştır. Önerdikleri yöntemin daha iyi performans gösterdiğini ve yüksek tanıma oranları elde ederken hesaplama süresini azalttığını mevcut birkaç özellik çıkarma yöntemiyle karşılaştırılarak sundular. Yaptıkları deneyler sonucunda dinamik ve statik işaretlerde sırasıyla %96,99 ve %96,78 sınıflandırma doğruluğu elde ettiler.

Khomami and Shamekhi (2021), yüzey elektromiyografisi ve atalet ölçüm birimi sensörlerini kullanarak Farsça İşaret Dili'ni (FİD) tercüme edecek giyilebilir ve düşük maliyetli bir cihaz önerisinde bulundular. Önerdikleri sisteme belirli zaman aralıklarında sensörlerin yardımıyla on gönüllüden alınan 20 Farsça işareti girdi olarak kullandılar. Sistemin yakalama doğruluğunu arttırmak için her iki sensörün en yüksek değerli 25

özelliği ve k-NN sınıflandırıcısına dayanan bir çalışma sundular. Yaptıkları çalışmalar %96,13 ortalama doğruluk elde ederek önerdikleri çalışmanın Farsça işaret dilinde uygulanabilirliğini gösterdiler.

Rezende *et al.*, (2021) işaret dili tanıma sistemlerinde jest tanımayla birlikte yaşanan görsel karışıklık, makine öğrenimi ve bilgisayarla görme alanlarında yaşanan zorluklar göz önünde bulundurularak Brezilya İşaret Dili (BİD) kamu veri tabanını geliştirmeye ve doğrulamaya yönelik bir çalışma sundular. İşaretlerin kayıt protokolü, işaretlerin seçimi, işareti yapan kişinin karakteristik özelliği, video çekimi için kullanılan sensörler ve yazılımlar, video kayıt durumu ve veri yapısı olmak üzere beş aşamadan oluşmaktadır. Veri tabanı oluşturulurken RGB-D sensörü ve RGB kamera kullanılarak kayıt protokolü aşamalarına dikkat edilerek on iki farklı kişiden BİD'den 20 işaretin 1000'den fazla videosu çekerek oluşturdular. Her bir işaret, her kişi tarafından beş kez kaydedilerek 1200 veri örneği içeren veri tabanı RGB video kareleri, derinlik, gövde noktaları ve yüz bilgilerini içermektedir. 3B ve 2B Evrişimli Sinir Ağlarına dayalı derin öğrenme yaklaşımlarının uygulanmasıyla geliştirilmiş veri tabanı ile ortalama %93,3'lük bir doğrulama elde ettiler. Yaptıkları çalışmayla halka açık bir geniş bir veri tabanı oluşturarak araştırma topluluğu için önemli bir katkıda bulundular.

Her işaret dilinin birbirinden farklı olması iletişim kanallarının da farklı olmasına sebep olmaktadır. Gündüz ve Polat (2021), Türk İşaret Dilinde kullanılan iletişim kanallarını temsil etmek için 4 RGB, 3 poz, 1 optik akış olmak üzere toplam 8 farklı veri akışı analiz ettiler. RGB ve optik akış için inception 3D ve poz veri akışları için LSTM-RNN kullanıldı. Önerilen çalışmada her birinin tek başına kullanıldığı testlerde sırasıyla RGB akışı için %28 ile %79, poz akışı için %9 ile %50 ve optik akışı için %78,5 doğrulama elde ettiler. Çok akışlı bir veri birleştirme mekanizması ile üç akışın birlikte kullanılması Boğaziçi işaret genel veri setinde %89,3 doğruluk elde ettiler.

Aksoy ve ark., (2021), derin öğrenme modellerini kullanarak Türk İşaret Dili (TİD) için kapsamlı bir çalışma sundular. TİD alfabesindeki 29 harf için 10.223 görüntü alınarak bir veri tabanı oluşturdular. CapsNet, AlexNet ve ResNet-50, DenseNet, VGG16, Xception, InceptionV3, NasNet, EfficientNet, Hitnet, SqueezeNet gibi Evrişimli Sinirli Ağ (ESA) tabanlı mimariler ile tasarladıkları TSLnet'i (Turkish Sign Language-

Net) kullanılarak görüntüler üzerinde sınıflandırma işlemi yaptılar. Yaptıkları karşılaştırmalı deneylerde en yüksek doğruluk oranlarını CapsNet ve TSLNet modeller için sırayla %99,7 ve %99,6 olduğunu tespit ettiler.

Görüldüğü üzere birçok araştırmacı konuşma dillerine özgü olan işaret dillerini tamamlayan dinamik ve statik işaretlerin tanınması ile ilgili ciddi anlamda çalışmalar yapmışlardır. Yukarda bahsedilen çalışmalar dışında oyunlar, akıllı ev sistemleri, çeviri mobil uygulamaları gibi birçok alanda işaret dili ile ilgili literatür çalışmaları da mevcuttur. Son yıllarda makine öğrenimi, derin öğrenme gibi alanların gelişmesiyle birlikte bilgisayarla görüntü tanıma alanında işaret dili tanıma popüler hale geldi. Bilgisayarla görme alanındaki en büyük yeniliklerden biri olan ve nesne algılama alanında sıklıkla tercih edilen YOLO - You Only Look Once (v1, v2, v3, v4 ve son olarak v5) işaret dilinde de son yıllarda kullanılmaya başlanılmıştır. İşaret dilinde YOLO'nun kullanıldığı birkaç literatür çalışmasına değinilecektir.

Geleneksel işaret dili öğrenme sistemlerinde en çok karşılaşılan iki zorluk eğitim verilerinde el alanın küçük olması ve jest-mimiklerin eklenmesiyle oluşturulan veri setlerinin çok büyük olmasıdır. Bunun üstesinden gelmek için Kim *et al.*, (2018) ilgili el bölgesini segmentasyon yöntemi ile ön işlemden geçirerek YOLO nesne algılama ağını kullanarak bunun üstesinden gelmeyi hedeflemiştir. İlgili el bölgesi tüm görüntü içerisinde bir nesne olarak görülmüş ve ESA uygulanarak nihai sonuca ulaşılmıştır. 12 statik işaretin test edildiği sistemde ilgili bölge segmentasyon işlemi ile doğruluğun %12 (%86'dan %98'e) arttığını ve eğitim süresinin %50'den fazla azaldığını gördüler. Yapılan bu çalışmayla önceden eğitilmiş el özellikleri sayesinde, öğrenmek için daha fazla işaret hareketi ekleme kolaylığı sağlandığı görülmüştür.

Bilgisayarla görme alanında işaret dili algılamada yaşanan karmaşık arka planda elin algılanmasına Nakjai *et al.*, (2019) Tay Parmak Yazımında (TPY) bir öneride bulundular. TPY tek el şemasına odaklanılan çalışmada 25 parmak işareti için yerleştirme ve sınıflandırma için evrişimli sinir ağı mimarisinin kullanıldığı YOLO tabanlı bir Tay Parmak Yazımını (YTPY) önerdiler. 15.000 görüntüden oluşan eğitim veri kümesi ve 15.000 görüntüden oluşan test veri kümesi üzerinde yaptıkları çalışmada çeşitli karmaşık arka plan koşullarında %82,06 ve tek düze bir arka planda %84,99 mAP'ye ulaştılar.

Ye *et al.*, (2020), doppler radar sensörler yardımıyla yakaladıkları işaretleri Yolov3-tiny yardımıyla sürekli işaret dili tanıma üzerinde bir çalışma sundular. Çince İşaret Dili (ÇİD) ve Amerikan İşaret Dili'ndeki (AİD) dört işaret ile yaptıkları çalışmada karmaşık ampirik mod ayrıştırması (Complex Empirical Mode Decomposition- CEMD) kullanarak işaretlerin spektrogramlarını elde ettiler. İşaretlerin mikro-doppler işaretleri için görüntü keskinleştirme kullanarak işaretlerin sınıflandırılması için çerçevesi ResNet ile değiştirilerek geliştirilmiş bir YOLOv3-minik ağ kullanıldılar. Önerilen yöntem ile sürekli işaret tanımada 0,924 hassasiyet, 0,993 geri çağırma, 0,957 F1-skor değeri (pozitif tahmin değeri) ve 0,99 mAP elde ettiler. Yaptıkları çalışma ile üç günlük bir konuşma senaryosunda 0,235 ortalama kelime hata oranı ile diğer çalışmalara göre %10 daha düşük hata payı elde ettiler.

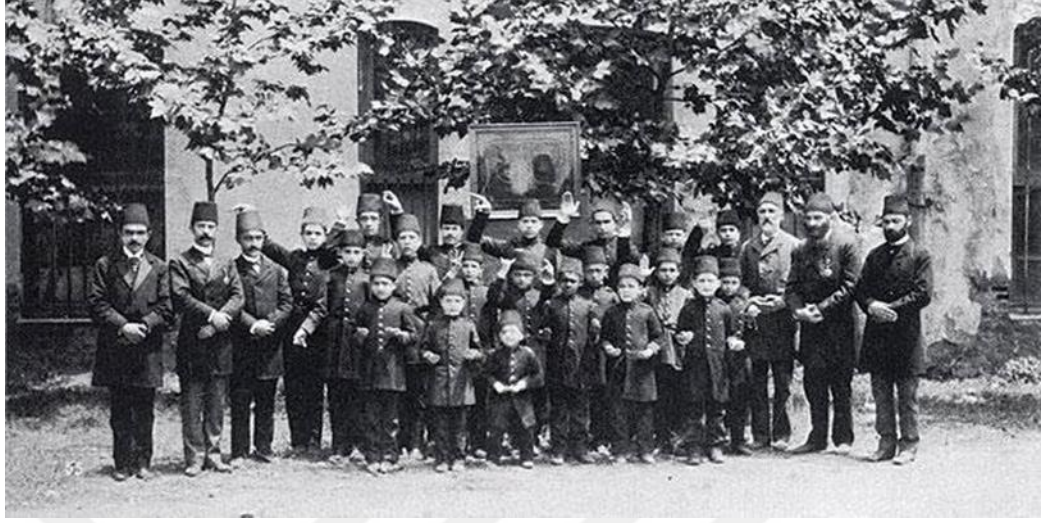
Özetlenen çalışmalarda görüldüğü üzere günümüze kadar çeşitli işaret dilleri için araştırmacılar tarafından farklı yöntemler önerilmiştir. Bunlarla birlikte, Türk işaret dili için literatüre baktığımızda sınırlı sayıda çalışmanın mevcut olduğunu görüyoruz. Bu çalışmalar içerisinde de derin öğrenme tabanlı olan YOLO ile herhangi bir çalışma bulunmamaktadır. Bu noktadan yola çıkarak bu çalışmamızda YOLOv4-CSP nesne algılama ağı Türk işaret dilinde tanıma alanında ilk defa kullanılmıştır. Bu amaçla Türk işaret dilinde bulunan on adet rakam (0,1,2,3,4,5,6,7,8,9) analiz ederek YOLOv4-CSP nesne algılama potansiyeli araştırılmıştır. Bu yöntem ile herhangi bir ön işleme ihtiyaç duyulmadan (filtreleme, segmentasyon vb.) ve uygularken de tek renk bir arka plan gerektirmeyen bir teknik sunulmuştur. Bu tez çalışmasıyla önerdiğimiz yöntem Türk işaret dili için yapılan çalışmalar arasında bir ilk olacak ve ileriki çalışmalar için de güçlü bir referans niteliğindedir.

3. MATERİYAL ve METOT

3.1. Türk İşaret Dili (TİD)

Türk İşaret Dili (TİD), Türkiye'de yaşayan işitme engelli insanların el, kol ve vücudun belli kısımlarını kullandığı bir iletişim dilidir. Dünyada ne kadar işaret dili olduğu bilinmemekle beraber her ülkenin kendine has işaret dilleri mevcuttur. Örneğin; Çince İşaret Dili (ÇİD), Amerikan İşaret Dili (AİD) Arap İşaret Dili (ArİD), Hintçe İşaret Dili (HİD) gibi. İşaret dilleri, bulunduğu ülkenin konuşma dilinden her ne kadar etkilense de kendine özgü bir dil yapıları vardır.

Türkiye'de işaret dilinin varlığı Osmanlı İmparatorluğu'nun kuruluş yıllarına kadar dayanmaktadır. O dönemde işaret diline ait pek fazla yazılı kaynak bulunmasa da eğitim konusundaki ilk çalışma Sultan II. Abdülhamit döneminde yapılmıştır. 1889 yılında Avusturya vatandaşı olan Ferdinand Grati'nin Sultan II. Abdülhamit'e ülkede sağır ve dilsizlere ait bir eğitim kurumunun bulunmadığını, sağır ve dilsizlerin sosyal hayatta yaşadıkları zorları dile getiren bir rapor sunmasıyla birlikte ilk sağır ve dilsizler okulu açılmıştır (Demirel, 2013). Yıldız Sağırlar Okulu ismiyle açılan okulda din ve cinsiyet ayrımı gözetmeksizin sağır ve dilsiz öğrencilerin davet edilmesiyle eğitim hayatına başlanılmıştır. Sultan II. Abdülhamit döneminde birçok kez yeri değiştirilen ve sonra kapatılan sağır ve dilsizler okulu Cumhuriyet Dönemi ile birlikte İzmir'de sağır, dilsiz ve körler okulu olarak eğitim hayatına devam etmiştir (Günay ve Görür, 2013). İzmir'de açılan okul düzenli ve sürekli eğitim faaliyeti göstererek bir süre Türkiye'nin tek sağır, dilsiz ve körler okulu olarak kalmıştır (Erten ve Arıcı, 2022).



Şekil 3.1. Hamidiye Ticaret Mektebi önünde Türk İşaret Diliyle “Padişahım Çok Yaşa” diyen çocuklar (Abdullah Frères, Servet-i Fünun dergisinin 129. sayısı)

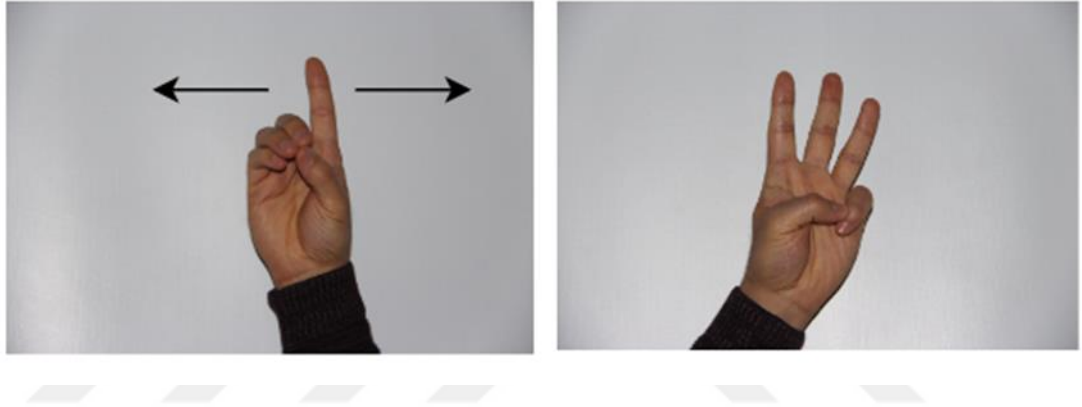
Cumhuriyetin ilerleyen yıllarında kısmen konuşma yeteneği ve sağır olan Süleyman Gök, kendi imkanlarıyla 1944 yılında İstanbul’da sağırılarla birlikte körlere de eğitim veren bir okul kurmuştur. 1950’lerde ilköğretim okulu olarak hizmet veren İstanbul, İzmir, Ankara ve Diyarbakır illerinde toplam dört okul vardı (Gök, 1958).

Ülkemizde, 1953 yılında işaret dili eğitimi yasaklanarak eğitimde “sözel yöntem” politikası benimsenmiştir. 52 yıl süren bu yasaklama politikası 2005 yılında çıkarılan ‘5378 sayılı Özürlüler Kanunu’yla Türk Milli Eğitim Sistemi’nde Türk işaret dili serbest bırakılmıştır. Bununla birlikte, Türkiye’de ilk defa Türk Dil Kurumu 2007 tarihinde “Birinci Türk İşaret Dili Çalıştayı” düzenlemiştir.

2012 yılında Millî Eğitim Bakanlığı tarafından Türk işaret dili sözlüğü hazırlanarak işaret dilindeki ihtiyacın karşılanması hedeflenmiştir. Türk işaret dili sözlüğü, günümüzdeki mevcut haliyle sık kullanılan 2.000 sözcük/kavramdan oluşmaktadır. Türk işaret dilindeki her hareket konuşma dilindeki bir kelimeye, bir harfe ya da bir rakama denk gelmektedir. El, jest, mimik ve baş, omuz gibi diğer vücut hareketlerinin bütünüyle geniş bir anlam dağarcığına sahiptir. Şekil 3.2’de gösterildiği gibi sık kullanılan anlamsal kavramlar için bir hareket bütünlüğü varken, Şekil 3.3’te ise özel isimler ve rakamlar için parmak alfabesi kullanılmaktadır.



Şekil 3.2. Hareket ifadesi ile kelime ‘VERMEMEK’ (TİD sözlüğü, 2015)

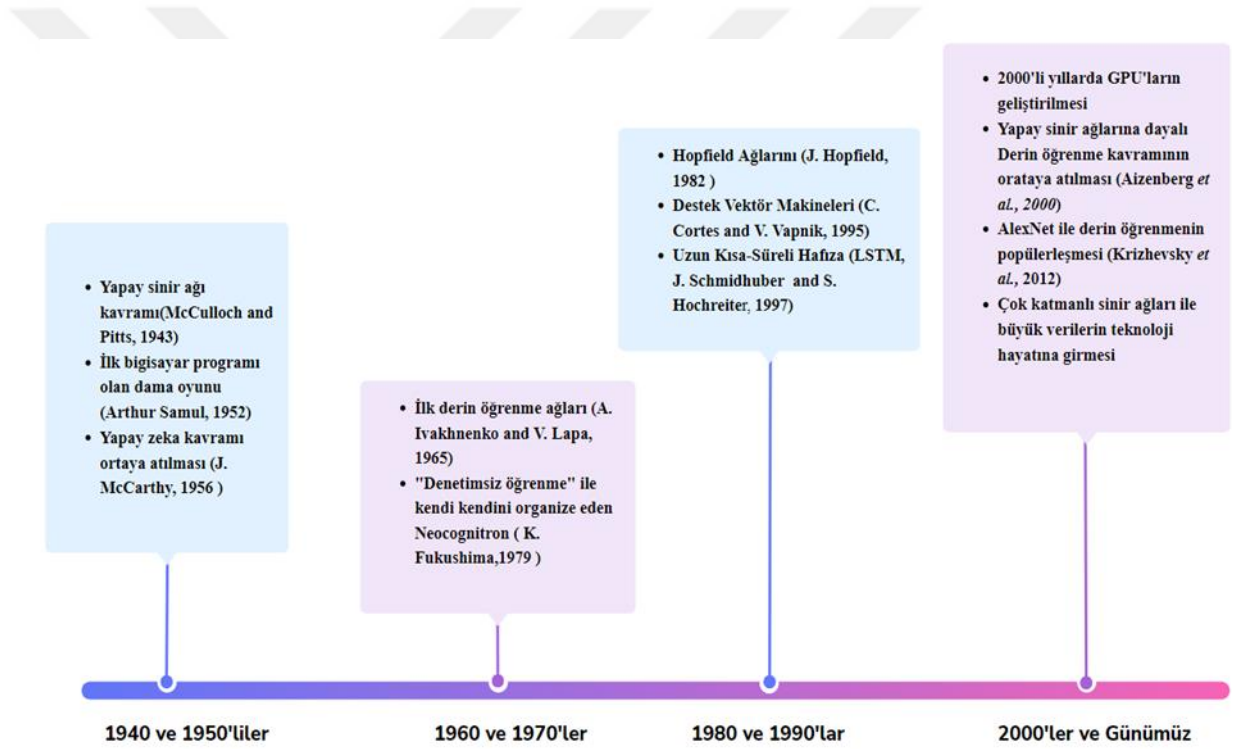


Şekil 3.3. Parmak alfabesi ile 13 sayısının yazımı

3.2. Derin Öğrenme

Bilgisayar teknolojilerinin hızlı gelişimi bilim dünyasında bilginin hızla yayılmasına ve beraberinde veri çokluğunu meydana getirmiştir. Bu veri çokluğu içerisinde yararlı bilgilerin hızlı ve doğru bir şekilde elde edilmesini amaçlayan birçok önemli çalışma yapılmıştır. Bu çalışmaların ürünlerinden biri olan derin öğrenmenin temeli 1943 yılına kadar dayanmaktadır. Şekil 3.4’te derin öğrenmenin tarihsel gelişiminde yaşanan önemli dönüm noktaları verilmiştir. 1943 yılında Pitts ve McCulloch tarafından insanın düşünsel sürecini taklit etmeyi amaçlayan yapay sinir ağlarını sundular. Bu çalışmayla birlikte, 1948 yılında Alan Turing tarafından paylaşılan bir makalede yer alan ‘düşünen makineler yaratma olasılığı’ düşüncesi ile günümüzde insanoğlunun hayatını kolaylaştırmak için yapılacak olan çalışmalar için bir dönüm noktası oluşturdu. Alan Turing’in yaptığı çalışma, tahminlerinin manuel olarak girildiği dünyanın ilk satranç oynayan bilgisayar programıydı (Shinde and Shah, 2018). İlk defa, 1956 yılında John McCarthy’n yaptığı bir konferansta literatürde "Artificial Intelligence

(AI)" olarak adlandırılan yapay zeka terimini ortaya attı. Yapay zeka, insanlar gibi düşünmeye ve hareketlerini taklit etmeye programlanmış makinelerde, insan zekasının simülasyonunu ifade eder. Bu terim ile birlikte öğrenme ve problem çözme gibi insan zihniyle ilişkili özellikler sergileyen herhangi bir makineye uygulanabilen ve yapay zekanın alt birimi olan makine öğrenimi geliştirildi. Böylece, makine öğrenimi ile modern toplumları birçok yönden destekleyerek teknolojiye ilerlemesini sağladılar. Son 10 yılda makine öğreniminin birçok pratik uygulama kullanılması ve bununla birlikte genel olarak yapay zekanın uygulama alanını genişleten Derin Öğrenme (Deep Learning - DL) kavramı kullanılmaya başlandı.



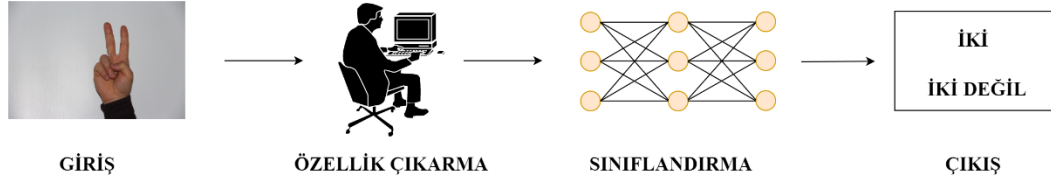
Şekil 3.4. Derin öğrenmenin tarihsel gelişimi

Derin öğrenme, yapay zekanın en yaygın kullanılan araçlarından biridir ve ayrıca makine öğrenmesinin bir alt dalıdır. Derin öğrenme, son yıllarda, birçok bilimsel disiplinde kullanılan makine öğrenimi alanında en yaygın kullanılan hesaplama yaklaşımı haline gelerek muazzam sonuçlar elde etti (Alzubaidi *et al.*, 2021). Geleneksel sinir ağlarından türetilen derin öğrenme, çoklu bir mimaride veriler arasındaki karmaşık ilişkileri modellerken, verilerdeki üst düzey soyutlamaları öğrenmeye çalışır ve özellik çıkarma, dönüştürme, model analizi ve sınıflandırma için

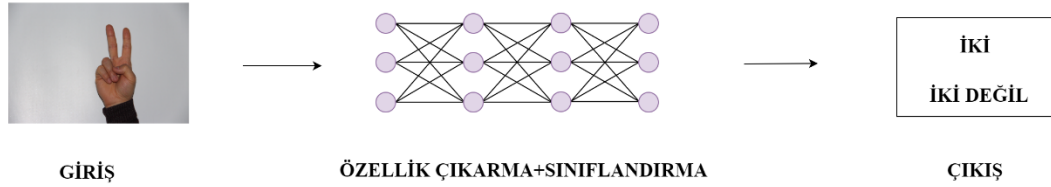
çok sayıda doğrusal olmayan bilgi işlem birimi katmanlarından yararlanır (Deng and Yu, 2014). Derin öğrenme, genel olarak derin mimaride hiyerarşik mimariler kullanarak alt düzey özelliklerinden üst düzey özellikler elde ederek ham verilerden otomatik özellik çıkaran zengin bir sinir ağı yapısına sahiptir (Ongsulee, 2017).

Derin öğrenme gelişmekte olan bir yaklaşımdır ve son yıllarda derin öğrenmeye olan ilginin artmasıyla birlikte anlamsal ayrıştırma (Bordes *et al.*, 2012), görüntü tanıma (He *et al.*, 2016), oyun oynama (Justesen *et al.*, 2020), doğal dil işleme (Mikolov *et al.*, 2013), tıbbi görüntü analizi (Pacal ve ark., 2020) gibi çeşitli geleneksel yapay zeka alanlarında yaygın olarak tercih edilmektedir. Derin öğrenme, bir görüntü tanıma işleminde, piksel başına yoğunluk değerleri vektörü veya bir dizi kenar, belirli şekle sahip bölge vb. özellikleri daha soyut bir hale getirmesi öğrenme görevini diğer algoritmalarından daha basit hale getirir (Ongsulee, 2017). Günlük hayatımızda Siri ve Google Asistan gibi sanal asistanlarda, Netflix ve Youtube gibi kişiselleştirilmiş eğlence platformları gibi birçok alanda derin öğrenme karşımıza çıkar. Derin öğrenmenin gerçek dünya uygulamalarında bu kadar popüler olmasının sebebi hem devasa veri kümelerinin eğitime izin vermesi hem de teknolojinin ilerlemesiyle birlikte geliştirilen grafik işleme birimlerinin (GPU'lar) muazzam bir paralel hesaplama gücüne sahip olması ile birlikte derin öğrenmede işlem hızının artmasıdır. Derin öğrenme ile artan hesaplama gücü, işlem hızının da artırılmasıyla, derin ağların ön-eğitim (pre-training) olmaksızın eğitilebilmesini mümkün kılarak karmaşık ve büyük verilerde hızlı ve doğru sonuca ulaşılmasına imkan sağlamıştır (Fan *et al.*, 2019).

GELENEKSEL MAKİNE ÖĞRENİMİ



DERİN ÖĞRENME

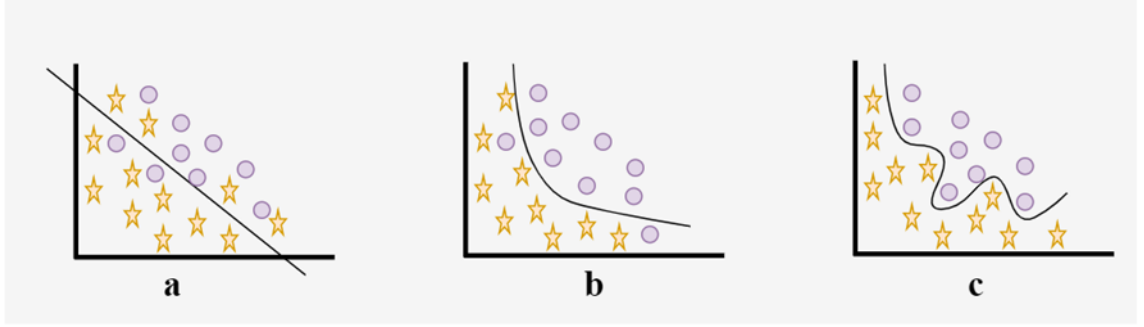


Şekil 3.5. Geleneksel makine öğrenimi ve derin öğrenme

Şekil 3.5'te geleneksel makine öğrenimi ve derin öğrenmenin karar mekanizması gösterilmiştir. Geleneksel makine öğrenimi ve derin öğrenme arasındaki temel fark algoritmaların nasıl öğrendiğidir. Geleneksel makine öğrenmesi sürecinde, denetimli öğrenme algoritmaları kullanılarak veri kümesindeki ilgili özellikleri içeren etiketli verileri olarak adlandırılan işlenmiş veriler ile eğitim yapılır. Makine öğreniminde bu özellik çıkarma adımı kullanıcı tarafından yapılır. Derin öğrenmede ise özellik çıkarma adımı, etiketlenmemiş veriler üzerinden, yani ham veriler kullanılarak eğitim sırasında çok katmanlı sinir ağı tarafından optimize edilerek otomatik olarak gerçekleştirilir. Eğitim sırasında etiketlenmemiş verileri kullanabilmesi derin öğrenme algoritmalarının benzersiz özelliklerinden biridir (Chaudhuri *et al.*, 2011). Çok katmanlı sinir ağları ile hiyerarşik yapılar kullanarak gerekli bilgiler doğrudan veriler üzerinden sağlanması daha doğru bir sonuca götürür. Günümüzde, sağır ve dilsiz insan sayısının atması ve bu insanların modern hayata hızlı bir şekilde ayak uydurması açısından işaret dili tanıma sistemleri büyük bir önem teşkil etmektedir. Bu yüzden bu çalışmada daha doğru sonuçlara ulaşılması adına derin öğrenme algoritmaları kullanılmıştır.

Derin öğrenme, çok sayıda özellik katmanına sahip ve en gelişmiş sonuçları elde etmeyi sağlayan etkili bir makine öğrenimi yöntemi olarak geliştirilmiştir. Derin öğrenme algoritmalarını büyük veriler üzerinde kendi kendine öğrenebilme ve özellik çıkarma işlemini zahmetsiz bir şekilde yapma özelliğine sahiptir. Bu sebeple, son yıllarda, derin öğrenme algoritmalarının bilgisayarla görme alanında kapsamlı bir şekilde çalışılmasına izin vererek nesne algılamada ilham verici bir performans sergilemiştir. Derin öğrenme, yapay sinir ağı, evrişimli sinir ağı, tekrarlayan sinir ağı gibi yaklaşımlara sahiptir. Bu yaklaşımlardan, insan beyin sinir yapısı ilham alınarak oluşturulan evrişimli sinir ağları, 2012 yılında yapılan bir nesne algılama yarışmasında, nesnelerin sınıflandırılmasında daha az hata payı bırakarak mevcut nesne algılama algoritmalarından daha iyi olduğu kanıtlayarak nesne algılama alanında dikkatleri üzerine çekmeyi başarmıştır (Krizhevsky *et al.*, 2012). (3.3.'te ESA detaylı bir şekilde açıklanmıştır). Bu teknolojik atılımla birlikte Nesne algılama algoritmaları derin öğrenme ile yeni bir döneme girmiş oldu.

Günümüzün önde gelen Facebook, Google, Netflix gibi şirketlerin müşteri davranış analizi, iş akış modellerindeki eğilimlerini öğrenmek ve geliştirmek için kullandıkları teknolojinin temeli büyüyen veri biliminin bir parçası olan makine öğrenimine dayanmaktadır. Günümüz teknolojisinin temeli her ne kadar buna dayansa da makine öğreniminde ortaya çıkan ve makine öğrenimi modellerinin performansını düşüren iki ana sorun vardır. Makine öğrenimi algoritmalarının eğitim veri kümeleri üzerinde gereğinden fazla çalışıp sonuçları ezberlemesi ile oluşan aşırı öğrenme (overfitting) sorunu ve yine kullanılan algoritmaların verileri yeterince yakalayamamasından dolayı ortaya çıkan eksik öğrenme (underfitting) sorunudur.



Şekil 3.6. a) Eksik öğrenme, b) İstenilen öğrenme ve c) Aşırı öğrenme grafikleri

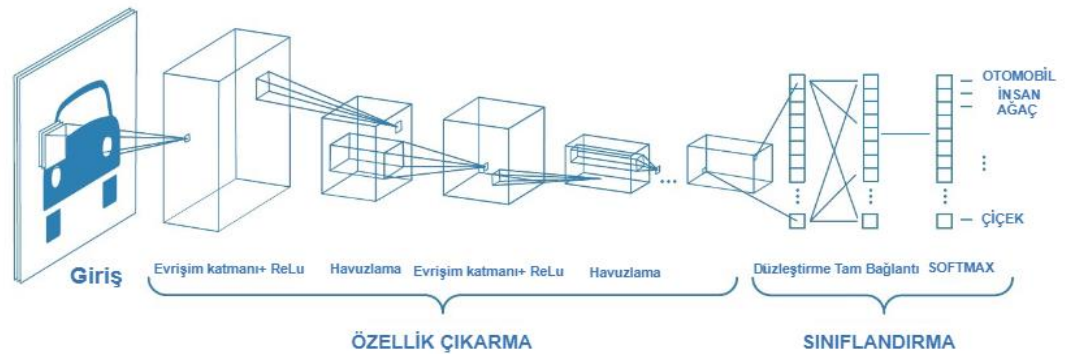
Şekil 3.6’da makine öğrenimi eğitim modeline ait sınıflandırma için basit lineer regresyon grafikleri verilmiştir. Şekil 3.6.a bir eksik öğrenme grafiğidir. Eksik öğrenme, modelinin eğitim ve test verileri arasındaki ilişkiyi doğru bir şekilde yakalayamadığı için bu veriler üzerinde iyi bir genelleme yapamayarak sınıflandırmayı iyi bir şekilde öğrenemediği durumlarda ortaya çıkar. Şekil 3.6.a’da görüldüğü gibi lineer bir eğilim göstererek tüm çıkış verilerini kapsamadığı görülmektedir. Bu durum hem eğitim hem de test verileri üzerinde çok yüksek hatalar vererek modelin düşük performans göstermesine yol açar. Şekil 3.6.c, bir aşırı öğrenme grafiğidir. Aşırı öğrenme, modelin eğitim verileri üzerinde fazla çalışarak veriler üzerindeki fazla ayrıntı ve onlardan gelen gürültüleri öğrenmeye çalıştığı zaman ortaya çıkar. Şekil 3.6.c’ye baktığımız zaman modelin eğitim verilerini doğru bir şekilde tahmin ettiğini ve mükemmel bir uyum gösterdiğini görebiliriz fakat bu istenmeyen bir durumdur. Çünkü, aşırı öğrenme durumu sadece eğitim verileri üzerinde iyi bir genelleme yaparak doğru tahminler yapmasını sağlar ve ağıın tüm veriler üzerinde genelleme yapmasının önüne geçerek test verileri üzerinde başarısız sonuçlar vermesine sebep olur. Regresyona dayalı olan nesne algılama algoritmaları istatistik bir uyum aranır. Şekil 3.6.b, eğitim verileriyle bir makine öğrenimi modelinin zaman içindeki performansı değerlendirilerek elde edilmesi beklenen en uygun ağ modelini gösteren iyi bir regresyon grafiğidir. İstenilen ideal öğrenmede hem eğitim hem de test verileri üzerinde minimum kayıplar göstererek veri setleri üzerinde birbirine yakın ve yüksek değerde tahmin sonuçları verir. Derin öğrenme sahip olduğu çok katmanlı yapısı ile daha büyük veri kümeleri ile eğitim yapılmasına izin vermesi ve ağıın fazla parametrelere sahip olması yukarıda

bahsettiğimiz durumların meydana gelmesini engelleyecek tekniklerin kullanılmasına izin verir.

3.3. Evrişimli Sinir Ağları (ESA)

Derin öğrenme, çok sayıda özellik katmanına sahip etkili bir makine öğrenimi yöntemi olarak geliştirilmiştir. Derin öğrenme mimarisi olan Evrişimli Sinir Ağı (ESA) mimarisi ise 20 yılı aşkın süredir kullanılmaktadır. Birden fazla girdi alma ve yerel alanlar içinde evrişim operatörleri kullanarak girdiler üzerinde işlem yapabilme yeteneğine sahiptir. Bu nedenle, son yıllarda, bilgisayarla görme alanında büyük ilgi görmüştür (Yu *et al.*, 2017). Evrişimli sinir ağları, hiyerarşik öğrenme özelliğine sahiptir ve uçtan uca bir öğrenme modeli sunarak görüntü özelliklerini eksiksiz bir şekilde öğrenebilme, görüntülerden iyi özellik çıkarabilme ve eğitim için parametre sayısını azaltabilme olanakları ile nesne algılama için uygun bir yapı sunar (Zhiqiang and Jun, 2017).

Çok katmanlı bir mimariye sahip olan ESA, en önemli özelliklerinden biri eğitim parametrelerini büyük ölçüde azaltmasıdır. Her katmandaki özellikler, kendinden bir önceki katmanın yerel alanının evrişimli çekirdek ağırlıkları paylaşarak üretilir (Zhiqiang and Jun, 2017). Katmanlarda bulunan yerel alanlardaki özellikler paylaşarak görüntü analizi gibi konularda ESA'yı diğer sinir ağlarına kıyasla daha iyi hale getirir. Şekil 3.7'de ESA'nın genel mimarisi verilmiştir.



Şekil 3.7. Evrişimli sinir ağı mimarisi (Mathwork)

Genel olarak, evriřimli sinir aęları mimarisi, birkaç evriřim (convolution) katman ve havuzlama (pooling) katmanın bulunduęu özellik çıkarma katmanı ve çıktı katmanına yakın bulunan tam baęlı (fully connected) katmanından oluřmaktadır (Guo *et al.*, 2016). Bir girdi verisi; görüntü, ses ve video gibi sinyallerden oluřabilir. Girdiler, öncelikle, her bir evriřim ve havuzlama katmanlarında iřlenerek özellik haritaları çıkartılır, bu özellik haritalarının çekirdek aęlıkları sırayla kendinden bir sonraki katmana aktarılır ve son olarak tüm katmanlarla yerel baęlantısı olan tam baęlantılı katmanda en iyi etiket tahmin edilerek nihai sonuç için sınıflandırmanın yapılacaęı bir eęitim sürecinden geçer.

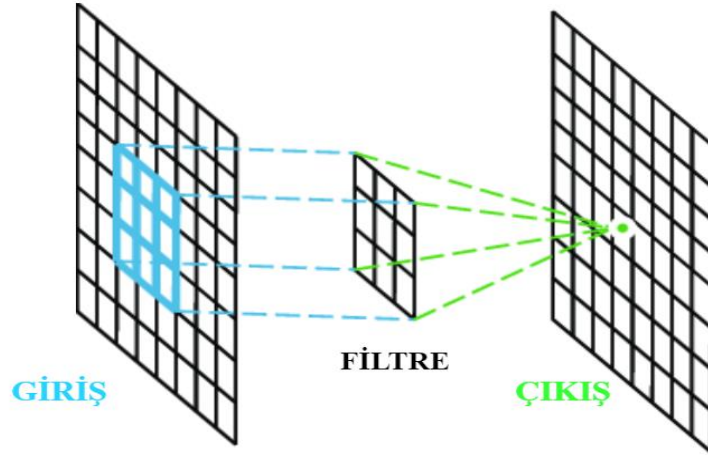
Evriřimli sinir aęları, katman baęlantı aęlıkları ile bunlara eklenen hata sapması oranı (bias) gibi öğrenilen parametreleri aktarmak için geri yayılım (Back Propagation-BP) algoritması kullanılır. Ayrıca, aę parametrelerinin optimize edilerek giriş ve çıkış arasındaki sapma miktarı azaltılıp denetimli bir öğrenme süreci sunar.

3.3.1. Evriřimli sinir aęları katmanları

Evriřimli Sinir Aęlarının özellik çıkarma ve sınıflandırma olmak üzere iki kısımdan oluřtuęuna değinilmiřti. Bu kısımlar sırasıyla evriřim katman, aktivasyon katmanı, havuzlama katmanı ve tam baęlantılı katmandan oluřmaktadırlar.

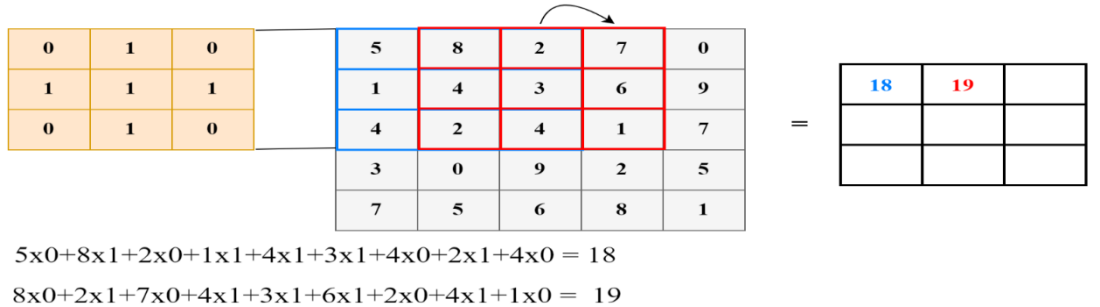
3.3.1.1. Evriřim katmanı (convolution layer)

Evriřimli sinir aęının ilk katmanı olan evriřim katmanı, girdilerin çıktısını belirleyen ve sinir aęlarının temel yapı taşıdır. Bu katmanda, giriş görüntüsünün temel özelliklerini çıkarmak için çeřitli filtreler kullanılır. Evriřim katmanın iřleyiři Şekil 3.8’de gösterilmiřtir. Burada, görüldüęü üzere bir giriş görüntüsü ile bir filtrenin birleřmesiyle elde edilen sonuç bir sonraki katman için giriş olarak yerini alır. Böylece eęitim boyunca giriş görüntüsünden filtreler yardımıyla öğrenilen yerel özellikler bir sonraki katmana iletilir.



Şekil 3.8. Evrişim katmanı (Yakura *et al.*, 2018)

Bilgisayarda, görüntüler NxM (N:Satır, M:Sütun) boyutunda büyük matrisler olarak algılanır. Evrişim katmanlarda giriş görüntüsünden küçük ve genellikle 3x3, 5x5, 7x7 gibi boyutlarında farklı filtreler kullanılır. Her eğitim görüntüsünün farklı özelliklerini algılamayı öğrenmeye çalışan bu filtreler, tüm görüntü üzerinde kaydırılarak uygulanır. Böylece, filtrelerin yardımıyla parlaklık, kenarlar bilgisi, keskinlik gibi çok basit ya da karmaşık işlemler yapılarak nesneler benzersiz şekilde tanımlanır. Her bir filtre görüntü ile birleşerek bir özellik haritası oluşturur. Özellik haritaları, her bir filtreye özgü özelliklerin keşfedildiği bölgelerdir. Kullanılan filtre sayısı kadar özellik haritası elde edilir.



Şekil 3.9. 5x5 boyutta 2 boyutlu bir giriş görüntüsüne 3x3'lük filtrenin uygulanması

Şekil 3.9’da, 5x5 boyutundaki giriş görüntü için 3x3 boyutunda bir filtrenin nasıl uygulandığı gösterilmektedir. Filtre maskesi, belirli yönlerde adım adım kaydırılarak

$$\begin{aligned}\text{Özellik haritası}(i, j) &= \text{giriş} * \text{filtre}(i, j) \\ &= \sum_m \sum_n \text{giriş}(i, j) \text{filtre}(i - m, j - n)\end{aligned}\quad (3.1)$$

tüm görüntü üzerinde dolaştırılır. Bu dolaşma esnasında matris sınırına gelindiğinde ise bir basamak aşağı kaydırılarak tekrar aynı işlemlere devam edilir. Filtre boyutu görüntü boyutu ile uyumlu olmadığı zaman sıfır dolgu yöntemiyle giriş matrisinin dışında kalan tüm öğeleri sıfır eklenerek uyumlu hale getirilebilir. Evrişim katmanında denklem

$$\begin{aligned}\text{Özellik haritası}(i, j) &= \text{giriş} * \text{filtre}(i, j) \\ &= \sum_m \sum_n \text{giriş}(m, n) \text{filtre}(i - m, j - n)\end{aligned}\quad (3.1)$$

(3.1)’de verilen konvülyasyon işlemi gerçekleştirilir (Goodfellow *et al.*, 2017).

Denklem (3.1)’de verilen i ve j terimleri, konvülyasyon işlemi sonucunda elde edilecek özellik haritalarının konumlarını ve m ve n terimleri filtrenin her bir konumunu ifade etmektedir. Evrişim katmanında konvülyasyon işlemi yerine birçok sinir ağı kitaplığında daha basit bir işlem olan çapraz korelasyon işlemi uygulanır (Goodfellow *et*

$$\begin{aligned}\text{Özellik haritası}(i, j) &= \text{giriş} * \text{filtre}(i, j) \\ &= \sum_m \sum_n \text{giriş}(i + m, j + n) \text{filtre}(m, n)\end{aligned}\quad (3.2)$$

al., 2017). Çapraz korelasyon işlem denklem (3.2) ile ifade edilir :

Özellik haritalarının her bileşeni giriş görüntüsündeki küçük bir yerel bölgeye bağlıdır. Evrişim katmandaki yerel bağlanabilirlik, sinir ağının, giriş görüntüsündeki

yerel bölgede maksimum düzeyde yanıt veren filtreleri öğrenmesine izin verir (Ke *et al.*, 2018). Böylece, konvolüsyon işlemi ile elde edilen özellik haritası ve filtre parametreleri tüm yerel konumlarda paylaşılır. Parametrelerinin paylaşımı sinir ağında kullanılacak parametre sayısını azaltır. Bu da evrişimli sinir ağlarının eğitilmeleri kolaylaştırır ve iyi bir genelleme yaptırarak performanslarını iyileştirir.

3.3.1.2. Aktivasyon katmanı (activation layer)

Evrişim katmanından gelen doğrusal özellik haritaları, ağı giriş görüntülerinden doğrusal olmayan özelliklere uyum sağlaması için bir aktivasyon işlevi kullanılır. Aktivasyon işlevi evrişim katmanından hemen sonra gelir. Aktivasyon katmanı olarak da adlandırılan bu katmanda, sigmoid, tanh, ReLu gibi aktivasyon fonksiyonları kullanılır. En yaygın olarak doğrusal olmayan ReLu aktivasyon fonksiyonu kullanılır (Agarap, 2018). Doğrusal olmayan ReLu aktivasyon fonksiyonun matematiksel denklemi aşağıdaki

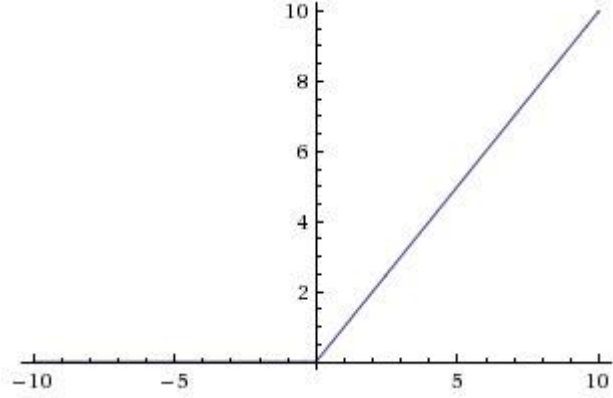
gibidir:

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

(3.3)

ReLu

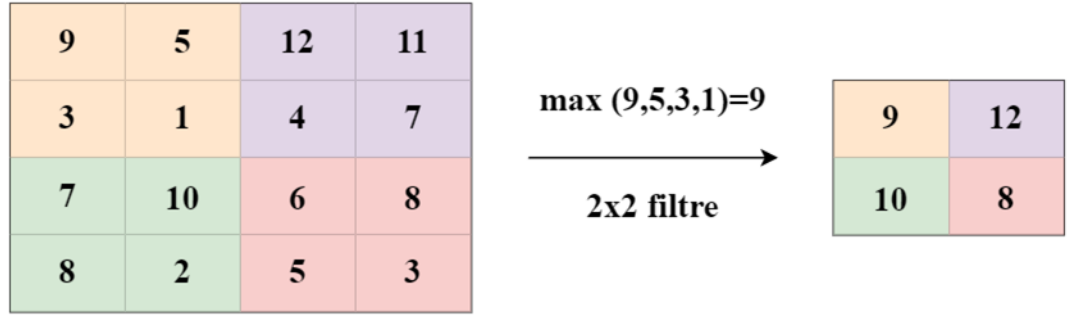
aktivasyon fonksiyonu, evrişim katmanından gelen negatif özellik harita ağırlıklarını sıfıra çekip ağla olan bağlantısını keserek giriş görüntüsünü uyumlu hale getirir. Böylece, ağı doğrusal olmayan özellikleri artırılarak ağı değişkenleri arasındaki her türlü sürekli ve karmaşık ilişki öğrenilir ve tahmin değeri yükseltilmiş olur. Doğrusal bir etkinleştirme işlevine sahip bir ağ iyi eğitilemez ve ağı tahmin doğruluğu azalır. Bu nedenle sinir ağlarının tamamen doğrusal olması istenmeyen bir durumdur. Şekil 3.10'da ReLu aktivasyon katmanının çıkış verileri üzerindeki etkisi gösterilmiştir. Şekilde de görüldüğü gibi negatif değerlerde daima sıfır değerini verirken pozitif değerleri ise aynen verir. ReLu aktivasyon katmanın hesap karmaşıklığının çok az olması oldukça hızlı sonuçlar üretmesini sağlar ve bu durum algoritmalarda tercih edilmesini sağlamaktadır.



Şekil 3.10. ReLu aktivasyon grafiği

3.3.1.3. Havuzlama katmanı (pooling layer)

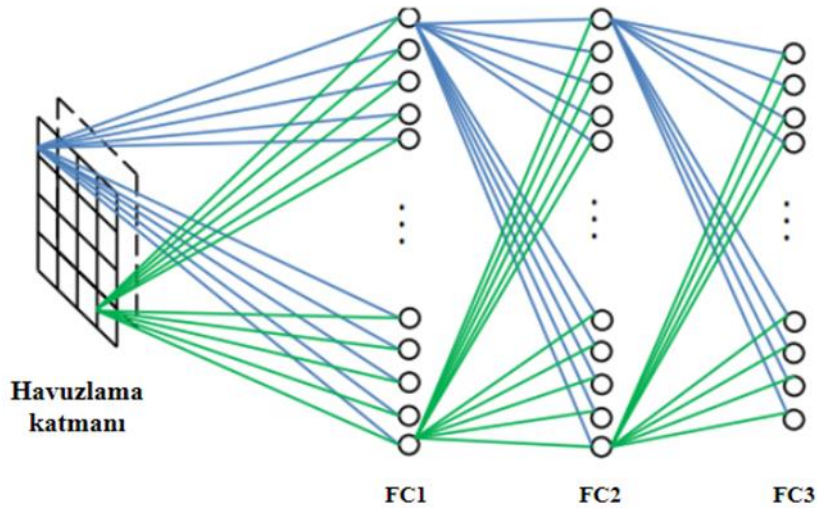
Genellikle bir havuzlama katmanı iki ardışık evrişim katmanı arasına dahil edilir. Havuzlama katmanı, sonraki katmanlar için eğitilebilir parametre sayıları azaltmak için kullanılır. Girişin derinliği değiştirilmeden alt örnekleme yapılarak bir ağıdaki hesaplama yükü azaltılıp fazla uydurmaların önüne geçilir. Alt örnekleme, ortalama ya da maksimum işlem kullanılarak oluşturulabilir. Havuzlama katmanında da belirli adımlarla filtreler uygulanarak piksellerin maksimum değerleri veya ortalama değerleri alınır. Bu iki havuzlama işleminden maksimum havuzlama daha iyi sonuç verdiği için yaygın olarak ESA mimarisinde kullanılır. Şekil 3.11’de maksimum havuzlama gösterilmiştir. Burada 4x4 boyutunda bir giriş görüntüsü üzerinde 2x2 boyutundaki max-pool filtresi iki adımda bir uygulanarak hesaplama yerine en büyük değere sahip değer döndürür. Böylece alt örnekleme yapılarak giriş görüntüsünün uzamsal boyutu küçülmüş olur.



Şekil 3.11. 2x2 filtre ile iki adım kaymalı maksimum havuzlama işlemi

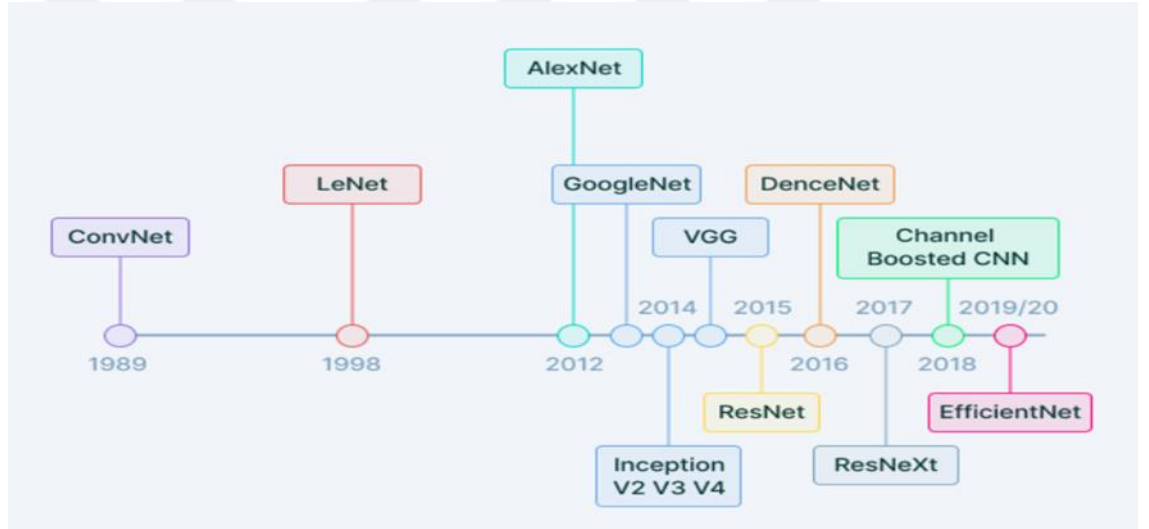
3.3.1.4. Tam bağlı katman (fully connected layer-FC)

Evrişim katmanları tarafından çıkarılan ve havuz katmanları tarafından alt örneklenen özellik haritaları bir veya daha fazla tam bağlantılı katmanına girdi olarak bağlanır. Şekil 3.12’de verilen tam bağlantılı katmanda havuzlama katmanından gelen iki boyutlu özellik haritaları tek boyutlu vektöre dönüştürülür. Tam bağlantı katmanı kendinden önceki katmanların hepsi ile bağlantılıdır. Tam bağlantılı katmandan sonra bir aktivasyon işlevi fonksiyonu kullanılır ve sınıflandırma yapılarak nihai sonuç elde edilir.



Şekil 3.12. Tam bağlantılı katman (Guo *et al.*, 2016)

Nesne algılama alanındaki son gelişmelerle birlikte evrişimli sinir ağıları modellerinde önemli gelişmeler olmuştur. Şekil 3.13'te nesne algılama algoritmalarında popüler olarak kullanılan modellerin tarihsel gelişimi verilmektedir. Şekilde görüldüğü üzere 2012 yılından sonra ESA tabanlı modellerin gelişiminde önemli bir hızla artış görülmektedir. 2012 yılında Krizhevsky ve ark. tarafından AlexNet'in nesne algılamada kullanmaları ile birlikte ESA tabanlı algoritmalar araştırmacıların ilgisini büyük ölçüde çekmiş ve bu alanında yeni birçok ESA tabanlı modeller hızla geliştirilmiştir. ESA'nın nesne algılama alanında önemli bir yere sahip olması, gelişmiş bilgi işlem gücü ve nesne algılama alanında sürekli yeni fikirlerin ortaya atılmasıyla birlikte gelişim süreci hızlı bir şekilde devam etmektedir.

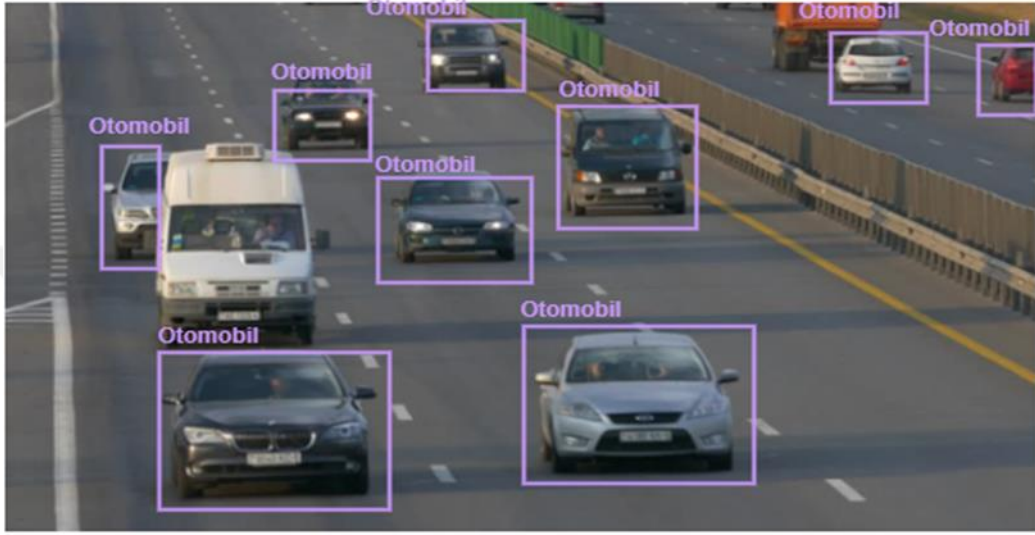


Şekil 3.13. Evrişimli sinir ağıları çeşitleri (Baheti, 2022)

3.4. Nesne Algılama

Günümüz teknolojisinde, nesne algılama, bilgisayar ve robot görüş sistemleri için gerekli olan temel yeteneklerden biridir. Hızla artan görüntü veri kümeleri içerisinde doğru bilgiye hızlı ve doğru bir şekilde ulaşmak çok önemlidir. Bu yüzden, görüntü işleme teknolojisinde bir görüntünün kapsamlı bir şekilde incelenip anlaşılması ve görüntüdeki hedef nesnenin doğru tanımlanması nesne algılama işlemini önemli bir noktaya getirmektedir. Nesne algılama, dijital bir görüntü ya da videoda önceden tanımlanmış kategorilerdeki (insan, ağaç, otomobil gibi) belirli sınıflardaki bir veya birden fazla anlamsal nesneyi hızlı ve doğru bir şekilde tanıma ve bunları yerleştirme görevini üstlenen bir bilgisayarla görme teknolojisidir (Deng *et al.*, 2020). Geniş bir

uygulama yelpazesine sahip olan nesne algılama, görsel arama, yüz algılama, sanal gerçeklik, video gözetimi, davranış analizi, robotik görme ve drone sahne analizi gibi karmaşık veya üst düzey görme görevlerinin hem akademik çalışmaların hem de çeşitli gerçek dünya uygulamalarının temel yapı taşı olarak benimsenmiştir.



Şekil 3.14. Nesne algılama ile otomobil tespiti

Nesne algılama, görüntüdeki her nesnenin her örneğinin ölçeğini gösteren eksen hizalı bir sınırlayıcı kutuyla birlikte nesnelerin görüntü içindeki konumunun yerelleşmesini ve görüntü hakkında anlamsal bilgiler elde edilmesini sağlar (Russakovsky *et al.*, 2015). İlgili görüntüdeki karmaşık arka plan, gürültü bozukluğu, düşük çözünürlük gibi etkenler hedef nesnenin bulunmasını bazen zorlaştırır da bu zorlukların üstesinden gelmek için literatürde birçok nesne algılama tekniği tartışılmıştır (Zou *et al.*, 2019). Nesne algılamaya yönelik mevcut yaklaşımlara baktığımızda makine öğrenimi teknikleri temel olarak kullanılmıştır. Evişimli sinir ağı, destek vektör makinesi (Support-Vector Machines-SVM), lojistik regresyon, k-en yakın komşu (k-NN), naive bayes sınıflandırıcı gibi makine öğrenimi algoritmalar kullanılarak giriş veri kümeleri üzerinde özellik haritalarını çıkarma, sınıflandırma, etiketlendirme vb. işlemlerin yapılmasını sağlar.

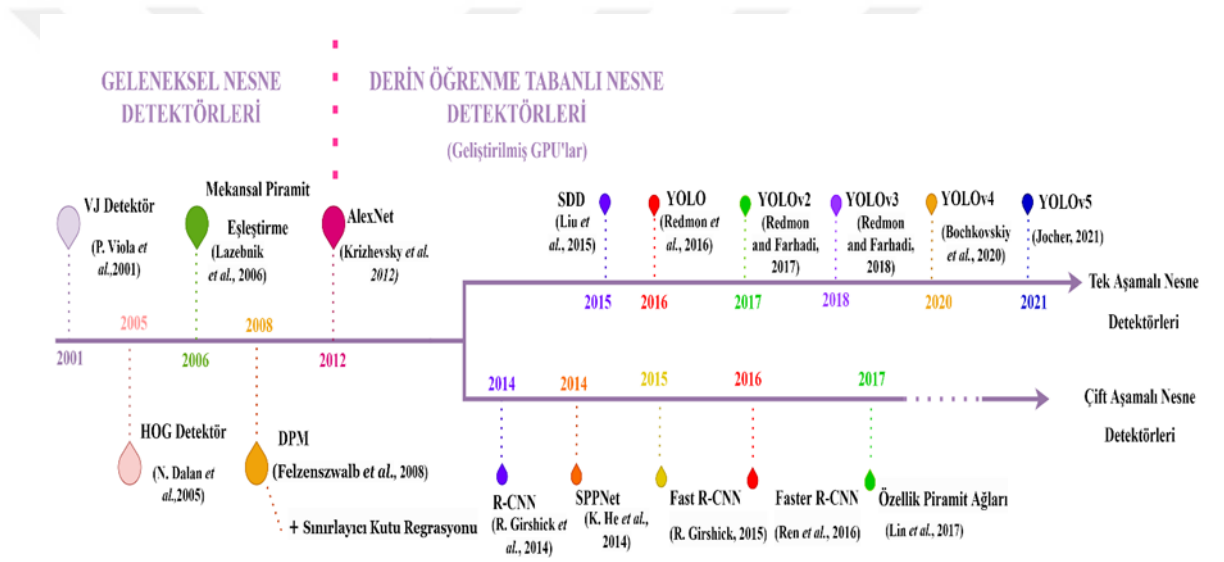
3.4.1. Nesne algılamanın tarihsel gelişimi

Bilgisayarla görmenin temel taşı olan nesne algılama, segmentasyon, sahne anlama, nesne izleme, görüntü altyazısı oluşturma gibi üst düzey görme konularının

temelini oluşturur. Geniş bir uygulama yelpazesine sahip olan nesne algılama, kapsamlı bir algılama işlemi için çok zengin ve rafine nesne bilgisine sahip olmalıdır. Bu yüzden, yaşanan son teknolojik atılımlar ile birlikte araştırmacıların artan miktar da bir ilgi çekmiştir. Nesne algılama tekniklerinin tarihsel gelişimi 2000’li yılların başlarına dayanmaktadır ve bu süreçte iki önemli gelişme dönemi yaşadı.

- Geleneksel Nesne Algılama Dönemi (2014 öncesi)
- ESA Dayalı Nesne Algılama Dönemi (2014 sonrası)

Şekil 3.15’te gösterildiği gibi iki ayrı döneme ayrılabilir.



Şekil 3.15. Nesne algılamanın tarihsel gelişimi

Nesne algılamanın ilk teknik gelişimi 2000’li yıllarının başında başladı. 2001 yılında, P. Viola ve M. Jones tarafından bulunan Viola-Jones (VJ) dedektörü ile ten rengi segmentasyonu yaparak ilk kez insan yüzünün gerçek zamanlı tespitini başardılar (Viola and Jones, 2001). VJ dedektörü, kayan pencereler yardımıyla herhangi bir pencerenin insan yüzü içerip içermediğini görmek için bir görüntüdeki tüm olası konumları ve ölçekleri gözden geçirme yöntemine dayanıyordu. Bu çalışmayı, birçok nesne algılama tekniğinin temelini oluşturan başka bir özellik tanımlayıcısı olan yönlendirilmiş gradyanların histogramı (Histograms of Oriented Gradient-HOG) (Dalal and Triggs, 2005) ve deforme edilebilir parça tabanlı model (Deformable Part Model-DPM) (Felzenszwalb *et al.*, 2010) nesne tespit çalışmaları izledi.

Makine öğrenimine dayanan geleneksel nesne algılama algoritmaları, kayan pencereler yardımıyla bilgilendirici bölge seçimi, özellik çıkarma ve son aşamada sınıflandırmayı temel alarak nesne algılamayı işlemini gerçekleştirir. Şekil 3.16'da geleneksel nesne algılama akış diyagramı gösterilmiştir. Geleneksel nesne algılama yöntemlerinin kendine özgü birtakım eksiklikleri vardı. Kayan pencere tabanlı bölge seçimi, yüksek bilgi işlem karmaşıklığı, yüksek kayan pencere fazlalığına sahip olması, morfolojik çeşitliliği, aydınlatma değişikliklerinin çeşitliliği ve arka planın çeşitliliğinin manuel olarak kontrol etmesi nesne tespit algoritmalarının tasarlanmasını zorlaştırır.



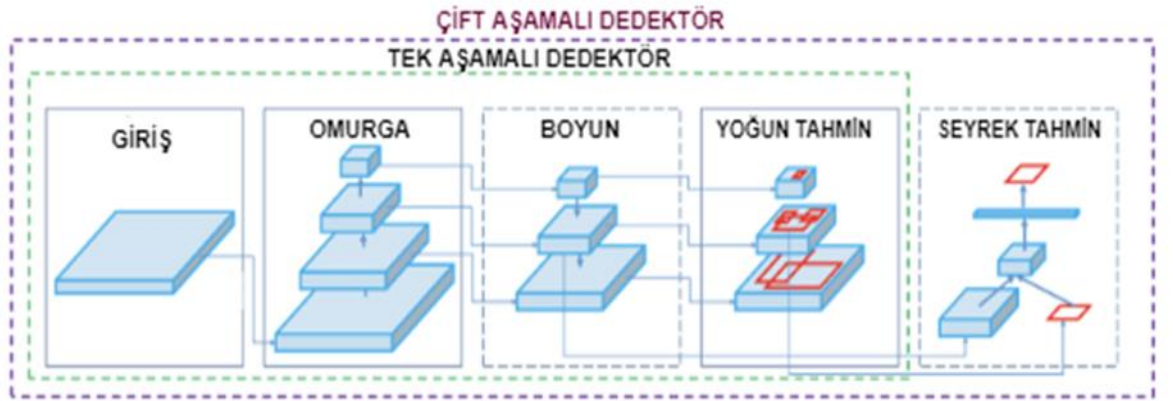
Şekil 3.16. Geleneksel nesne tespiti akış diyagramı

Manuel özelliklere dayanan geleneksel nesne algılama teknikleri, makine öğreniminin alt birimi olan derin öğrenmenin keşfedilmesiyle önemli bir dönüm noktası yaşadı. 2012 yılının sonlarında, Krizhevsky ve ark. tarafından derin öğrenmeye dayalı evrişimli sinir ağını nesne algılama tekniklerine uygulamayı önerildiler. Yaptıkları çalışma ile dünya çapında en iyi bilinen bilgisayarla görme yarışması olan ImageNet Classification'da nesne sınıflandırmada daha az bir hata payı bırakarak mevcut nesne algılama algoritmalarına karşı üstün bir başarı kazandılar (Krizhevsky *et al.*, 2012). Bu gelişmeyle birlikte nesne algılama tekniklerinde yeni bir dönem açılmış oldu. Böylelikle, evrişimli sinir ağının geliştirilmesi ve bilgi işlem gücünün sürekli olarak iyileştirilmesi ile genel nesne algılamanın bel kemiği haline geldi (Jiao *et al.*, 2019). Evrişimsel sinir ağı tabanlı nesne algılama teknikleri iki kategoriye ayrılabilir. Bu kategoriler; iki aşamalı nesne algılama (aday bölgeler) yöntemi ve tek aşamalı nesne algılama (regresyon) yöntemidir.

3.4.2. Evrişimli sinir ağı tabanlı nesne algılama algoritmaları

Derin öğrenmenin klasik makine öğrenimi algoritmaları dahil olmak üzere birçok bilgisayarla görme algoritmalarının sınırlamalarının üstesinden gelerek önemli ölçüde geliştirilmiş bir nesne algılama performansına sahip olduğuna daha önce

değinişmişti. Evrişimli sinir ağı ise nesne algılamada derin öğrenme algoritmalarının bel kemiğidir. Şekil 3.17’de nesne algılama algoritmalarının genel mimarisi gösterilmektedir. Şekilde de gösterildiğı gibi evrişimli sinir ağı tabanlı nesne algılama algoritmaları iki kategoriye ayrılabilir: Çift aşamalı nesne algılama (aday bölgeler) yöntemi ve tek aşamalı nesne algılama (regresyon) yöntemleridir. Modern bir nesne detektörü, giriş görüntüsünün özelliklerini çıkararak özellik haritasının oluşturan omurga (bachbone) ve nesne sınıflarını ve sınırlayıcı kutularını tahmin etmek için kullanılan algılama kafası (head) adı verilen yapılardan oluşur. Genellikle nesne algılama detektörleri kafa yapılarına göre çift aşamalı nesne detektörü ve tek aşamalı nesne detektörü diye ayrılır. Bunlara ek olarak son yıllardaki nesne algılama modellerinde boyun ve algılama kafası arasında bulunan farklı aşamalardan özellik haritalarını toplamaya yardımcı olan boyun (neck) adı verilen bir ek bağlantıya sahipler.



Şekil 3.17. Nesne algılama algoritmalarının genel mimarisi (Bochkovskiy *et al.*, 2020)

3.4.2.1. Çift aşamalı nesne algılama

Çift aşamalı nesne algılama algoritmalarında, eğitim sırasında, hedef nesne bilgilerini çıkarmak için bölge önerisi (Region of Interest – ROI) kullanılır. Nesne algılama süreci, özellik haritalarına ilk aşamada bölge önerisi ağı uygulanır ve ikinci aşamada her bölge için tahminler yapılarak sınıflandırma işlemi gerçekleştirilir. Bu yöntem, ağıın hesaplama doğruluğu yükseltir ancak hesaplama miktarını artırarak gerçek zamanlı algılamasını zorlaştırır. Çift aşamalı nesne algılama algoritmalarına R-CNN serisi örnek olarak verilebilir.

Çizelge 3.1. R-CNN serisi ve özellikleri

R-CNN serisi	VOC 2007 (%mAP)	VOC 2012 (%mAP)	Tahmin süresi (sn)	Özellikler
R-CNN (Girshick <i>et al.</i> , 2014)	58,5	53,3	47	2000 bölge önerisi için seçici arama kullanılır, fazla depolama alanı gerektirir
Hızlı R-CNN (Girshick, 2015).	70,0	68,4	2	Seçici arama ile çıkarılan özellik haritalarına bölge önerisi havuzlaması uygulanır
Daha hızlı R-CNN (Ren <i>et al.</i> , 2016).	78,8	75,9	0,2	Bölge öneri ağı kullanılarak özellik haritaları çıkartılır

Çizelge 3.1’de VOC 2007 ve VOC 2012 veri setleri üzerinde birkaç R-CNN serinin uygulanarak elde edilen % mAP değerleri, tahmin süreleri ve özellikleri karşılaştırılmalı olarak verilmiştir. VOC veri setleri, birçok nesne kategorisi içeren nesne algılama algoritmaları için kullanılan veri seti örnekleridir. Verilen R-CNN seri içinde en iyi sonuç daha hızlı R-CNN ile elde edilmiştir. Daha hızlı R-CNN nesne

algılama algoritmasında çoklu bölge seçmek yerine bölge öneri ağıının kullanılması hem ağıın daha hızlı olmasına hem de daha doğru sonuçlar vermesini sağlamaktadır.

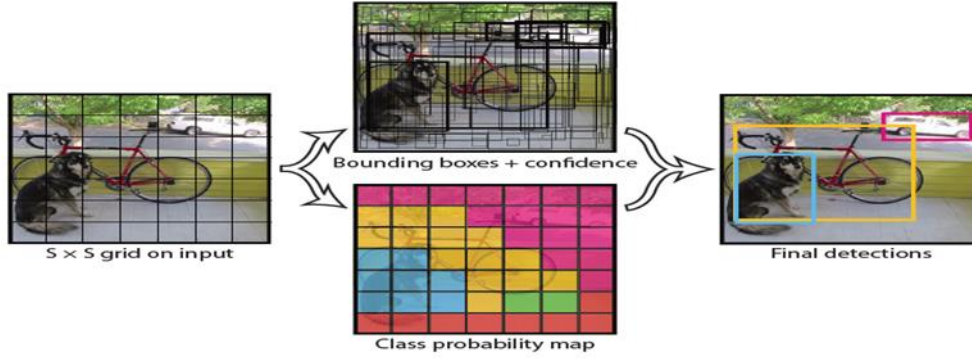
3.4.2.2. Tek aşamalı nesne algılama

Tek aşamalı nesne algılama algoritmaları, ilgili bölgeleri ayrı ayrı önermek yerine, tam bir regresyon problemi kullanarak doğrudan görüntüden sınırlayıcı kutu ve nesne sınıfını elde eden bir nesne algılama işlemi önerir. İki aşamalı nesne algoritmaları, önemli bölge tespiti için bir ön aşamayı ve ardından bu alanlarda bir nesne tespiti için bir sınıflandırma kullanır. Bu, nesne tespiti ve yerelleşme hassasiyeti açısından iki aşamalı nesne algoritmaların çoğunun daha iyi bir performans göstermesini sağlasa da tek aşamalı nesne algılama algoritmalarından hızı açısından değerlendirildiğinde oldukça yavaştır. Bu nedenle esnek yapı ve gerçek zamanlı algılama için tek aşamalı nesne algılama algoritmaları daha çok tercih edilmektedir. Bu ağlara örnek olarak, Sadece bir kez bakarsın (You only look once - YOLO) serisi verilebilir.

Bir sonra ki bölümlerde YOLO serisi ayrıntılı olarak anlatılacaktır.

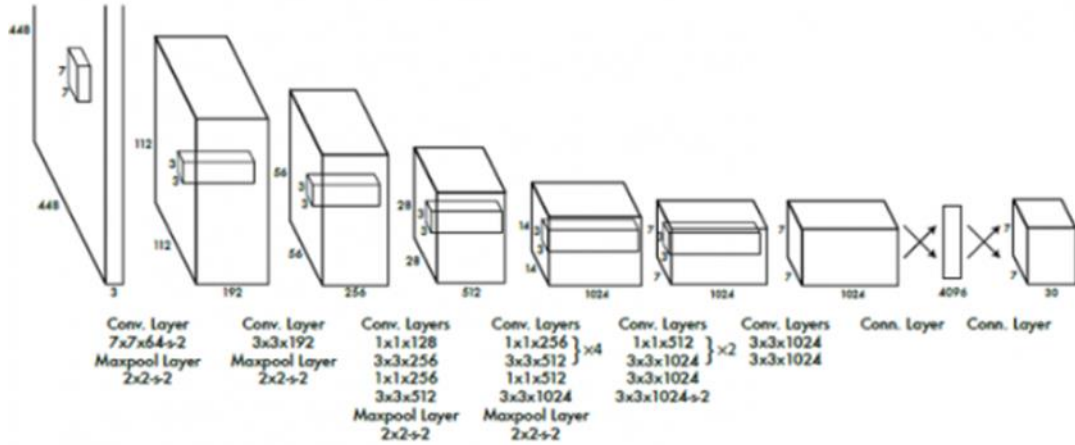
3.5. YOLO - You Only Look Once

Redmon ve ark. tarafından önerilen nesne algoritması olan YOLO, ilk regresyon tabanlı tek aşamalı nesne algılama algoritmasıdır. Araştırmacılara gerçek zamanlı bir nesne tespiti sunan YOLO serisinin ilki 2016 yılında piyasaya sunulmuştur. Redmon, daha önceki nesne algoritmalarındaki bölge tespiti+sınıflandırma mantığını tamamen terk ederek bunun yerine tüm görüntüye tek bir sinir ağı uygulamıştır. Birleşik ağ mimarisi olan YOLO, uçtan uca bir algılama yaparak sınırlayıcı kutular ve sınıf olasılıklarının tahminlerini doğrudan görüntüden alır. Basit bir regresyon analiz modeli ile görüntüleri gerçek zamanlı olarak saniyede 45 kare hızında işler ve gerçek zamanlı algılamayı sunar (Redmon *et al.*, 2016).



Şekil 3.18. YOLO Birleşik Algılama Modeli (Redmon *et al.*, 2016)

Şekil 3.18’de görüldüğü gibi, birleşik ağ modeli verilen YOLO, görüntüleri $S \times S$ boyutunda ızgaralara böler. Her ızgara hücresi, kendisine denk gelen nesne merkezini algılar. Daha sonra, her ızgara hücresinde, her biri bir güven puanına sahip B sınırlayıcı kutular tanımlanır. Güven puanı, sınırlayıcı kutu içerisi nesnenin bulunma olasılığını ifade eder. Güven puanı belirlenirken iki sınırlayıcı kutu arasındaki örtüşme oranı olan birleşim üzerinde kesişim (Intersection Over Union-IOU) oranı dikkate alınır.



Şekil 3.19. YOLO mimarisi (Redmon *et al.*, 2016)

Şekil 3.19’da verilen YOLO mimarisi, 24 evrişim katmanı, 4 maksimum havuzlama katmanı ve ardından 2 tam bağlantılı katman vardır. Katman sayılarını azaltmak için 1x1 evrişim ve ardından 3x3 evrişim katmanı gelir. Küçük boyutlu görsellerin algılanması için giriş görüntüler 448x448 boyutunda eğitilir. Ağın, son

tahmin çıktısı $7 \times 7 \times 30$ boyutunda olur. Aktivasyon fonksiyonu olarak tüm katmanlar sızdıran doğrultulmuş doğrusal aktivasyonu (Sızdıran Relu) kullanılır. YOLO nesne algılama ağı, eğitim ve test sırasında tam görüntüyü kullanılması ve sınıflandırmasıyla ilgili bağlamsal bilgilerin etkin bir şekilde yakalaması, daha hızlı R-CNN gibi iki aşamalı nesne algılama algoritmalarına kıyasla daha fazla arka plan hatası verir. YOLO, gerçek zamanlı nesne algılamayı başarmak için kapıyı açsa da küçük boyutlu nesneleri algılamada ve olağandışı en boy oranlarına sahip nesneleri algılamada önemli sayıda yerleştirme hatası verir. YOLO nesne algılama ağında gerekli iyileştirilmeler bir sonraki YOLO versiyonlarında yapılmıştır.

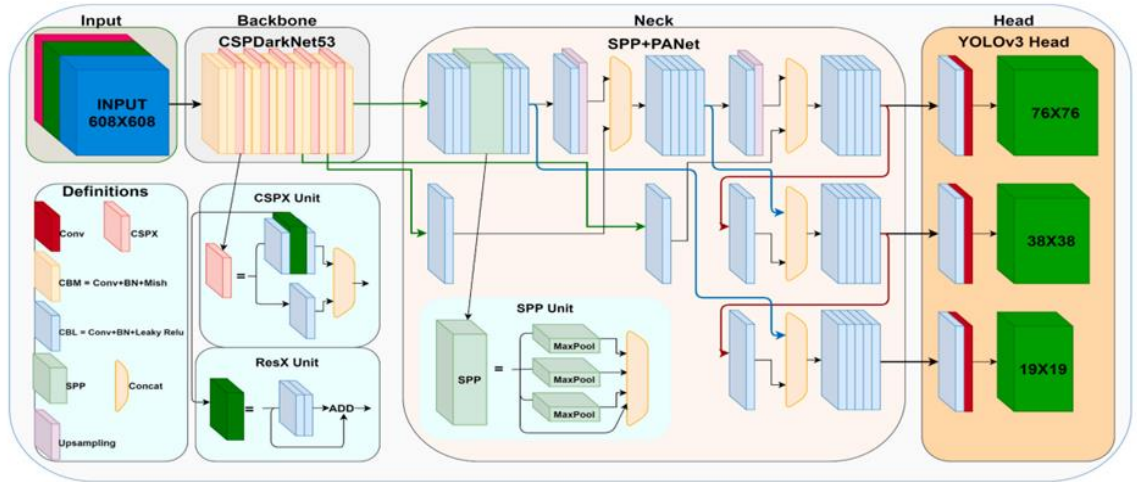
2017 yılında, ilk versiyonundaki sınırlamaların üstesinden gelmek için, Redmon ve Farhadi YOLO'da bazı iyileştirmeler yaparak YOLOv2 (Redmon and Farhadi, 2017)'yi sundular. YOLOv2'deki temel amaçları sınıflandırma doğruluğunu koruyarak geri çağırma ve yerleştirmede iyileştirmeler yapmaktır. Bunun üzerine ağ mimarisinde 19 evrişim katmana sahip Darknet19 ve eğitim sırasında katman çıktılarının dağılımlarının değişmesini önleyerek ağı öğrenme hızını arttırmak için toplu normleştirme (BN) (Ioffe and Szegedy, 2015) önermişlerdir. Ayrıca, çok ölçekli bir eğitim, bağlantı kutuları ve boyut kümesi de ağ mimarisinde kullanılan diğer yapılardır. Bütün bu yeniliklere rağmen YOLOv2 sahip olduğu omurga yapısından dolayı nesne algılama doğruluğunda sınırlamalara neden olmaktadır. Bu sınırlamaların üstesinden gelmek için 2018 yılında omurga yapısında Darknet mimarisinin bir versiyonu olan Darknet-53 omurga yapısına sahip YOLOv3 (Redmon and Farhadi, 2018) piyasaya sürüldü. DarkNet53(Redmon and Farhadi, 2018) yapısında artık modüller ve kısa yol bağlantıları kullanılarak daha geniş katmanlarda öznetelik bilgileri elde edildi. YOLOv3 mimarisini önceki versiyonlardan ayıran en önemli özelliklerinden biri, bir görüntünün içindeki farklı boyutlardaki nesneleri, özellikle küçük nesneleri algılamak için çok ölçekli tahmin yapan Özellik Piramit Ağlarını (Feature Pyramid Network-FPN) (Lin *et al.*, 2017) kullanılmasıdır. FPN ağının kullanılması ve DarkNet53 omurgası YOLOv3'ü hem hız hem de performans açısından oldukça başarılı hale getirmiştir. YOLOv3 mimarisinde zengin özellik haritaları çıkar nesnelerin bilgilerini hem düşük hem de yüksek düzeyde yakalamak için ağda üç nesne algılama kafası bulunmaktadır. Ağ yapısında üç farklı yerde 32, 16 ve 8 adımda alt örneklemeler yapılarak üç farklı ölçekte

algılama sağlanır. Bununla birlikte YOLOv3'te lojistik regresyon kullanılarak farklı en-boy oranlarını hesaplayan sınırlayıcı kutular için nesne güven puanı ve sınıf tahmini yapılır.

2020 yılında YOLOv3'ün geliştirilmiş bir versiyonu olan ve mAP değerinde %10'a kadar ve saniyedeki kare sayısında (fsp) ise %12'ye kadar bir iyileşme sağlayan YOLOv4 algoritması geliştirildi. YOLOv4 algoritması, sinir ağının öğrenme yeteneğini geliştirirken hesaplama ve bellek maliyetlerini azaltmak için birden fazla tekniğin bir araya getirilmesi ile optimal ve doğru bir nesne algılama ağı olarak sunuldu. YOLOv4 algoritması CSPDarknet53 omurgasına, bir uzamsal piramit havuzu (Spatial Pyramid Pool-SPP) ve değiştirilmiş bir yol toplama ağı (Path Aggregation Network-PANet) içeren boyun ve YOLOv3'te bulunan üç algılama kafasına sahiptir. Bölüm 3.4.1'de YOLOv4 algoritması ayrıntılı olarak açıklanmıştır. 2021 yılında YOLO ailesinin en son geliştirilmiş versiyonu olan YOLOv5 piyasaya sürüldü. Açık kaynak olarak sunulan YOLOv5 (Jocher, 2021) henüz bir makale olarak sunulmamıştır. Bu tez çalışmasına başladığımızda nesne algılama algoritmalarında YOLOv4-CSP algoritması mevcuttu. Bu yüzden, bu tez çalışmasında daha hızlı ve iyi sonuçlar almak için orijinal YOLOv4 mimarisi optimize edilmesi ile oluşturulan YOLOv4-CSP (Wang *et al.*, 2021) kullanılmıştır.

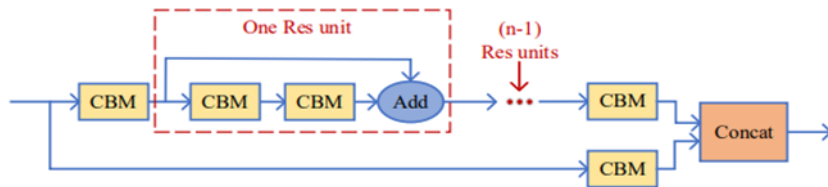
3.5.1. YOLOv4 (YOLO versiyon 4)

YOLO ailesinin en son teknoloji ürünü YOLOv4 (Bochkovskiy *et al.*, 2020), mevcut diğer derin öğrenme tabanlı nesne algılama tekniklerine kıyasla gerçek zamanlı olarak yüksek algılama hızı ve yüksek algılama doğruluğu ile geliştirilmiş bir performansla 2020 yılında piyasaya sürüldü. YOLOv4'ün yapısı diğerlerinden farklı olarak öznitelik çıkarımı için kullanılan bir omurga, nesne sınıflandırması ve sınırlayıcı kutu tahmini yapan bir kafa ve öznitelik haritaları toplayarak zengin anlamsal özellikler sağlayan ek yol bağlantıları içeren boyun adı verilen yapılardan oluşmaktadır.



Şekil 3.20. YOLOv4 Mimarisi (Pacal ve Karaboğa, 2021)

YOLOv4 algoritmasının omurgasında gradyan sorunlarını azaltıp ağırlık öğrenme yeteneğini artırmak için CSPNet (Wang *et al.*, 2020) ve DarkNet53'ün (Redmon and Farhadi, 2018) birleştirilmesiyle oluşturulan CSPDarkNet53 kullanılmaktadır. CSPDarkNet53 evrişim yapısında, 5 blok CSP ve 11 CBM modülünden oluşur. Her bir CSP bloğu, evrişim katman, toplu normelleştirme (BN) ve Sızdıran ReLU (Xu *et al.*, 2015)'dan oluşmaktadır. CBL modülleri ise CSP bloğu, evrişim katman, BN ve Mish aktivasyon işlevinin birleştirilmesi ile oluşur. Nesne algılama ağında bir yenilik olan CSP blokları, ağırlık derinleşmesini ve daha fazla özellik çıkarmasını sağlayan residual (Res) birimleri içerir. Şekil 3.20'de detaylı olarak gösterilen YOLOv4 mimarisinde CSP blokları CSP1, CSP2, CSP8, CSP8, CSP4 bloklarını içerir. Bloklar içerdikleri Res birimleri sayılarına göre adlandırılır. Şekil 3.21'de bir CSP yapısı gösterilmektedir.



Şekil 3.21. CSPn yapısı (Fu *et al.*, 2020)

YOLOv4 nesne algılama ağının boyun kısmında Mish aktivasyon işlevi (Misra, 2019) kullanılırken ağın geri kalan kısmında Leaky aktivasyon işlevi kullanılır. Mish aktivasyon işlevinin ortalama performansı diğer aktivasyon işlevlerine oranla daha iyidir. Bu da ağın genelleme kabiliyetini güçlendirerek fazla uydurmanın önüne geçer.

Ağın boyun kısmında kullanılan SPP (*Spatial Pyramid Pooling*) modülü (He *et al.*, 2015), ağın uzamsal boyutunu koruyarak ağın alıcı alan boyutunu iyileştirir. SPP modülünde, 1x1, 5x5, 9x9, 13x13 boyutlarındaki kayan pencereler ile maksimum bir havuzlama uygulanır. Son evrişim katmanına bağlanan SPP modülü, ağın çalışma hızında neredeyse hiç azalmaya sebep olmadan ağın doğruluğunu artırır. YOLOv4 algoritmasında yerelleşmeyi sağlamak için değiştirilmiş Yol Toplama Ağları (Path Aggregation Network-PAN) kullanılır. Değiştirilmiş PAN, alt katmanlardaki özellikleri, birleştirilmiş kısa yollar yardımıyla üst katmanlara ulaştırır. Böylece, ağın farklı katmanlardan bilgi toplanmasını kolaylaştırır. Son olarak, ağın boyun kısmında değiştirilmiş bir Uzamsal Dikkat Modülü (Spatial Attention Module-SAM) kullanılır. SAM modülünde iki özellik haritası oluşturmak için giriş özellik haritalarına ayrı ayrı maksimum ve ortalama havuzlama uygulanır. YOLOv4'te bu havuzlama katmanları kaldırılarak yerlerine evrişim katmanı içeren değiştirilmiş bir SAM modülü kullanılır.

YOLOv4 algoritmasının kafa kısmı, çıkarılan özellikleri kullanarak nihai sonucu veren kısımdır. YOLOv3'ün detektör kısmında bulunan üç YOLO kafası benimsenmiştir. YOLOv4 detektör başlıklarının girdileri, boyun kısmından gelen zengin uzamsal bilgilerden oluşur. Buradaki ana işlev, en iyi sınırlayıcı kutuyu bulmak ve sınıflandırma yapmaktır. Detektörde 19x19, 38x38 ve 76x76 boyutlarında ayrı ayrı özellik haritaları oluşturulur. Farklı boyutlardaki nesneleri bulmak için kullanılan bu başlıklar, hedef nesnenin koordinatlarını (x,y, genişlik(w), yükseklik (h)), bir güven puanı ve bir sınıf tahmin puanı içerir. Sonuç kısmı bu verilere dayandırılarak neticelendirilir.

YOLOv4 mimarisi oluşturulurken birçok nesne algılama algoritması test edilerek en uygun optimizasyon sağlanmaya çalışılmıştır. Bochkovskiy ve ark. bir dizi özel modül içeren iki yöntem önermişlerdir. Bunlar; modelin tespit doğruluğunu artırmak için modele farklı eğitim stratejileri sunan ya da sadece eğitim maliyetini

arttıran Bag of Freebies (BoF), ve daha az hesaplama maliyeti ile nesne algılama doğruluğunu önemli ölçüde artıran Bag of Specials (BoS) yöntemleridir. Bu yöntemlerin içerikleri, farklı amaçlar için tasarlanmış detektörlere göre farklılık gösterebilir. BoF ve BoS modül sahip oldukları içerikler ile araştırmacılara geniş bir çalışma yelpazesi sunarak farklı amaçlar için kullanılabilecek birçok yeni nesne algılama algoritmalarının oluşturulmasına olanak tanır. YOLOv4'ün verimliliğini önemli ölçüde artırmak için hem omurga hem de detektör kısmında (yani kafa ve boyun) BoF ve BoS kullanılmıştır. YOLOv4'te kullanılan BoF ve BoS modül içerikleri Çizelge 3.2'de verilmiştir.

Çizelge 3.2. BoF ve BoS içerikleri

Yolov4 mimarisi	Eğitimde kullanılan BoF	Sonuç çıkarma için kullanılan BoS
Omurga	• CutMix and Mosaic veri artırma • DropBlock regularization • Class label smoothing	• Mish activation • Cross-stage partial connections (CSP) • Multi-input weighted residual connections (MiWRC)
Detektör (Boyun + Kafa)	• CIoU-loss • CmBN • DropBlock regularization • Mosaic data augmentation • Self-Adversarial Training(SAT) • Cosine annealing scheduler • Optimal hyperparameters	• Mish activation • SPP-block • SAM-block • PAN path-aggregation block • DIOU-nms

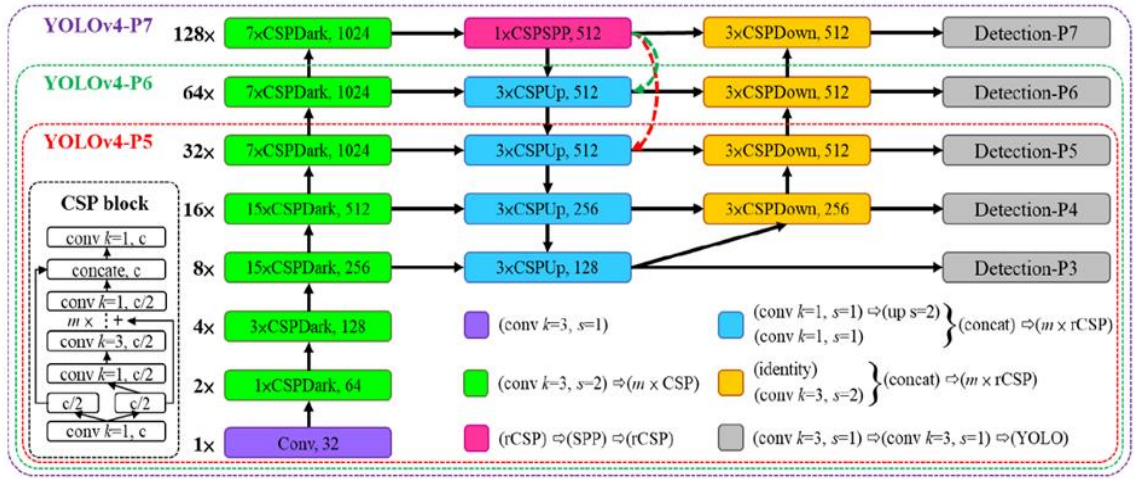
Nesne algılama detektörünün çevrim dışı eğitilmesi ve eğitim sırasında çıkarım maliyetini etkilemeden detektörden daha doğru sonuçlar elde etmek tercih edilen bir eğitim metodolojisidir. Bu nedenle, BoF yöntemine uyan ve nesne algılama detektörlerinde sıklıkla tercih edilen tekniklerden biri olan veri artırma tekniği bu eğitim

stratejisini sunar. Veri artırmanın temel amacı, nesne algılama detektörünün farklı ortamlarda elde edilen görüntülere karşı daha yüksek hassasiyete sahip olmasını sağlamak için mevcut eğitim verilerini değiştirerek yeni eğitim verileri elde eder. Tek görüntülerde parlaklık, renk tonu gibi fotometrik bozulmalar ve döndürme, rastgele ölçekleme gibi geometrik bozulmalar veri artırma tekniği olarak kullanılırken birden fazla görüntü için mixup (Zhang *et al.*, 2018) ve cutmix (Yun *et al.*, 2019) teknikleri kullanılır. Mixup tekniği, rastgele alınan iki görüntüyü farklı katsayılar ile çarparak bu iki görüntüyü birleştirir ve oluşan görüntünün etiket ayarlamasını yaparken cutmix tekniği ise bir görüntüden kırılan parçayı başka bir görüntünün kırılmış kısmına ekleyerek iki görüntüyü birleştirir. Bu veri artırma teknikleri dışında rastgele silme, cutout, ızgara maske, dropout, dropconnect ve dropblock gibi veri artırma teknikleri de mevcuttur. YOLOv4, veri artırma tekniklerinden dört farklı görüntüden yeni bir çoklu görüntü oluşturan mozaik ve cutmix veri artırma tekniklerini benimser. Ayrıca, BoF yöntemi, modeli daha iyi hale getirebilmek için farklı sınıflar arasındaki ilişki derecesini öğrenmek için yumuşak etiket (label smooting) tekniğini ve sınırlayıcı kutu regresyonu amaç fonksiyonunu içerir. Sınırlayıcı kutu (Bounding Box-BBox) görüntünün içindeki ilgili nesnenin ana hatlarını içeren x,y koordinatları, yükseklik ve genişliğe sahip dikdörtgen bir sınıflandırma kutusudur. Nesne algılama algoritmalarında görüntü içindeki nesnenin bütünlüğünü dikkate almak ve daha hassas bir sınıflandırma yapmak için sınırlayıcı kutular için kayıp fonksiyonları kullanılır. Geleneksel nesne algılama yöntemlerinden farklı olarak tahmin edilen sınırlayıcı kutu ile gerçek sınırlayıcı kutu arasındaki örtüşme oranını veren IoU kaybı (Yu *et al.*, 2016) önerilmiştir. Bununla birlikte IoU kayıp fonksiyonundan yola çıkarak sınırlayıcı kutular arasında örtüşmeyen kısımları da hesaba katmak için GIoU (Generalized Intersection over Union), iki sınırlayıcı kutu arasındaki merkez nokta mesafesini hesaplayan DIOU (Distance-IoU) ve sınırlayıcı kutuların örtüşen alanını, merkez noktaları arasındaki mesafeyi ve en boy oranını hesaba katan CIOU (Complete IOU) araştırmacılar tarafından ileri sürülmüştür.

Nesne algılama ağının hassasiyetini önemli ölçüde artıran BoS yöntemleri birçok eklenti modülleri içerir. Bu eklenti modülleri, ağın alıcı alanı genişlenmek için SPP, dikkat mekanizması için SAM, farklı katmanlardaki özellik haritalarını toplamak için PAN, ağın hassasiyetini artırmak aktivasyon fonksiyonu olarak Mish aktivasyon

fonksiyonu ve son işlem aşamasında gerçek sınırlayıcı kutuyu en baskın hale getirmek için DIoU-nms'den oluşur. YOLOv4 algoritması yapısı itibariyle optimizasyonu kolay olan bir nesne algılama ağıdır. Orijinal YOLOv4 makalesinde de gösterildiği üzere YOLOv4'ten daha iyi sonuçlar elde etmek için BoF ve BoS yöntemlerinden en uygun olan algoritmaların farklı kombinasyonları bir araya getirilerek oluşturulmaya çalışılmıştır. Yapılacak olan çalışmada daha hızlı ve iyi sonuçlar almak için orijinal YOLOv4 algoritmasının optimize edilmesi ile oluşturulan YOLOv4-CSP (Wang *et al.*, 2021) kullanıldı.

YOLOv4-CSP algoritması, YOLOv4 algoritması ve CSPNet'in birleştirilmesi ile oluşturulmuştur. Şekil 3.22'de mimarisi verilen YOLOv4-CSP, YOLOv4 algoritmasında bulunan PAN ve SPP modüllerine CSPNet eklenmesiyle oluşturulmuş bir boyun yapısına sahiptir. YOLOv4-CSP algoritmasının en önemli özelliği yapısında CSPNet mimarisinin kullanılmasıyla hem hesaplama maliyetinin düşmesi hem de tahmin doğruluğunun ve hızının artmasıdır (Pacal ve ark., 2022). Ayrıca, YOLOv4-CSP algoritması YOLOv4 algoritmasının yukarı ve aşağı ölçeklendirilmesine dayanmaktadır. YOLOv4-CSP algoritmasının modelin ağırlık ve derinliğine göre küçük (small), orta (medium) ve büyük (large) olarak ayrılabilir. Aslında bu, katman sayısında nesne tespit algoritmalarının ölçeklenebilir bir yapıya sahip olmaları hasebiyledir. Bu yüzden, yapılacak olan çalışmada daha hızlı ve iyi sonuçlar almak için YOLOv4-CSP nesne algılama ağı üzerine odaklanmıştır.



Şekil 3.22. YOLOv4-CSP (Scaled YOLOv4) (Wang *et al.*, 2021)

Sinir ağları yapısında, ağıın önemli bir paçası oluşturan, performansını ve sınıflandırma hassasiyetini önemli derecede etkileyen iki önemli faktör vardır. Bunlar aktivasyon ve kayıp fonksiyonlarıdır. Literatürde araştırmacılar tarafından birçok aktivasyon fonksiyonu ve kayıp fonksiyonu önerilmektedir. Bu fonksiyonlar, nesne algılama ağıının tasarlanma amacına göre ve fonksiyonun işlevine göre farklılık gösterebilir. YOLOv4-CSP’de optimum sonuçların elde edilmesini sağlamak için tamamlayıcı seçeneklerden ağı boyunca Mish aktivasyon fonksiyonu ve hızlı yakınsama hızına ve sınırlayıcı kutu regresyonunda daha iyi performansa sahip olmak için CIOU kayıp fonksiyonu benimsenmiştir. Bir sonraki bölümlerde bu yapılar detaylı olarak açıklanmaktadır.

3.5.1.1. Aktivasyon Fonksiyonu

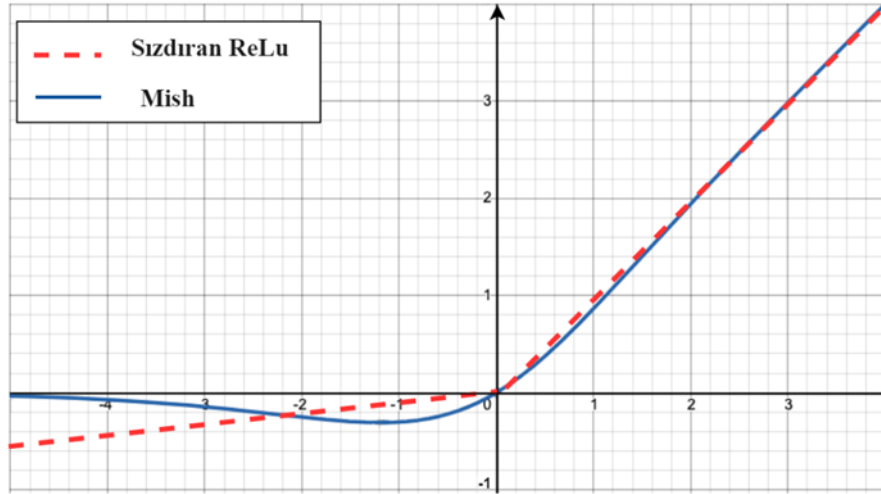
Aktivasyon fonksiyonu sinir ağlarında önemli bir role sahiptir. Sinir ağlarının çıktısını belirlemek için kullanılan aktivasyon fonksiyonları matematiksel bir dönüşüm uygulayarak bir sonraki katman için bir girdiye dönüştürür. Aktivasyon fonksiyonları, sinir ağlarının sağlam bir eğitim gerçekleştirmesi ve daha yüksek sınıflandırma hassasiyetine sahip olması açısından önemli bir parametredir. Aktivasyon fonksiyonlarının türevlenebilir olması geri yayılım (Back Propagation-BP) için önemli bir etkidir. Geri yayılım, sinir ağı eğitiminde minimum kayıp oranını veren ve ağıın genelleme yeteneğini arttıran parametrelere sahip bir algoritmadır. Ağıın iyi öğrenmesini sağladığı için önemli bir yere sahiptir.

YOLOv4 algoritması tasarlanırken omurga kısmında Mish aktivasyon fonksiyonu, boyun ve baş kısımlarında ise Sızdıran ReLu kullanılmaktadır. Sızdıran ReLu aktivasyon işlevi, derin öğrenmede popüler bir etkinleştirme işlevidir. Sızdıran ReLu, ReLu'da sıfır değeri olarak ölü gradyan problemine sebep olan negatif değerler için bir çözüm sunar. Negatif değerlere sahip girdileri çok küçük bir α katsayısı ile çarparak çok küçük bir negatif sayı döndürür ve sabit bir eğime izin verir. Böylece, negatif girdi için bile geri yayılımı mümkün kılar. Sızdıran ReLu aktivasyon fonksiyonunun matematiksel denklemi:

$$f(x, \alpha) = \begin{cases} \alpha * x, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases} \quad (3.3)$$

Mish aktivasyon fonksiyonu (Misra, 2019), kendi kendini düzenleyen, pürüzsüz, sürekli ve monoton olmayan bir aktivasyon işlevidir. Mish aktivasyon fonksiyonunun matematiksel ifadesi şu şekildedir:

$$f(x) = x * \tanh(\ln(1 + e^x)) \quad (3.4)$$



Şekil 3.23. Mish ve Sızdıran ReLu aktivasyon fonksiyon grafikleri

Şekil 3.23'te Sızdıran ReLu ve Mish aktivasyon fonksiyonlarının grafikleri verilmektedir. Şekilde görüldüğü gibi her iki aktivasyon fonksiyonu pozitif değerler için sonsuza doğru giderken negatif değerler için Mish aktivasyon fonksiyonu sıfıra daha çok yakınsamaktadır. Böylece, Mish aktivasyon fonksiyonu ağıın eğitim hızının iyi olmasını sağlayarak fazla uydurmaların önüne geçer ve ağıın genelleme kabiliyetini ve sınıflandırma doğruluğunu arttırmış olur. Ayrıca, Mish aktivasyon fonksiyonun düşük gradyan kaybına sahip olması ve eğitim sürecinde daha iyi performans gösterir. Bu da nesne algılama algoritmalarında onu daha popüler hale getirir (Gustineli, 2022).

3.5.1.2. Kayıp Fonksiyonu

Kayıp fonksiyonları, nesne algılamada hedef nesneyi algılama hassasiyetini etkileyen diğer bir önemli faktördür. YOLO algoritmasında birden fazla sınırlayıcı kutu tahmin edilir. Ancak, hedef nesneyi algılamaktan yalnızca bir sınırlayıcı kutu sorumludur. YOLO algoritmalarında kayıp fonksiyonları, nesne sınıflandırma kaybı, nesne yerleştirme kaybı ve güven kaybı değerlerinin toplam kare hatalarının toplamı ile bulunur (Redmon *et al.*, 2016).

- Sınıflandırma kaybı: tespit edilen her bir nesne, her sınıf için sınıf koşullu olasılıklarının kare hatası hesaplanır. Sınıf kaybı denklemi:

$$L_{\text{sınıf}} = \sum_0^{S^2} 1_i^{\text{obj}} \sum_{c \in \text{sınıflar}} (p_i(c) - \hat{p}_i(c))^2 \quad (3.5)$$

1_i^{obj} : Hücrede nesne varsa 1 yoksa 0 olmaktadır.

$\hat{p}_i(c)$: i hücresindeki c sınıfı için koşullu sınıf olasılığı.

- Yerleştirme kaybı: Tahmin edilen sınırlayıcı kutunun konumundaki ve boyutundaki hataları ölçer. Sadece nesneyi tespit eden sınırlayıcı kutu için hesaplanır. Yerleştirme kaybı denklemi:

$$L_{\text{yerel}} = \alpha_{\text{koord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]$$

$$+ \alpha_{koord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \quad (3.6)$$

1_{ij}^{obj} : i. hücredeki j. sınırlayıcı kutu nesne algılamaktan sorumlu ise 1 değilse 0

α_{koord} : Sınırlayıcı kutu koordinatlarındaki yitim değerinin ağırlığını artırır.

- Güven kaybı: Sınırlayıcı kutudaki nesne tespit edildiği zaman kutunun nesnellik ölçütünü verir. Güven kaybı denklemi:

$$L_{güven} = \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 \quad (3.7)$$

1_{ij}^{obj} : i. hücredeki j. sınırlayıcı kutu nesne algılamaktan sorumlu ise 1 değilse 0.

$\hat{C}_i$: i hücrelerinde j. kutunun güven değeri .

Eğer sınırlayıcı kutuda nesne tespit edilmezse:

$$\alpha_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \quad (3.8)$$

1_{ij}^{noobj} : 1_{ij}^{obj} değerinin tümleyeni

α_{noobj} : arka planı tespit ederken kaybı azaltır

- Toplam kayıp:

$$L = L_{sınıf} + L_{yerel} + L_{güven} \quad (3.9)$$

Birçok nesne algılama algoritmalarında sınırlayıcı kutu tahmini için IoU işlevi (Yu *et al.*, 2016) kullanılır. IoU işlevi, tahmin edilen sınırlayıcı kutu ve gerçek sınırlayıcı kutu arasındaki örtüşme derecesini tanımlamak için kullanılan bir metriktir.

Gerçek pozitiflik (GP) durumunda olan kaybı hesaplamak için oluşan sınırlayıcı kutular içerisinde güven puanı en yüksek olan IoU değeri alınır. Belirlenen bir eşik IoU değerinden küçük olan değerler ihmal edilir. IoU, aşağıdaki gibi formüle edilir:

$$IoU = \left| \frac{B \cap B^{gt}}{B \cup B^{gt}} \right| \quad (3.10)$$

- IoU işlev kaybı:

$$L_{IoU} = 1 - IoU \quad (3.11)$$

IoU işlev kaybı sınırlayıcı kutular çakıştığı zaman çalışır aksi durumlarda herhangi bir öğrenme sağlamamaktadır. YOLOv4'te IoU işlevi yerine bir CIoU (Zheng *et al.*, 2020) işlevi de önerilmektedir. CIoU işlev kaybı, IoU işlev kaybına kıyasla sınırlayıcı kutu regresyonu için gerçek sınırlayıcı kutunun merkez noktası ile tahmin edilen sınırlayıcı kutunun merkez noktası arasındaki mesafeyi ve sınırlayıcı kutuların en ve boy oranlarını dikkate alan önemli bir geometrik faktör sunarak sınırlayıcı kutuların en ve boy oranının tutarlılığını korur. CIoU kaybı şu şekilde formüle edilir:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \quad (3.12)$$

$$v = \frac{4}{\pi^2} \quad (3.13)$$

$$\alpha = \frac{v}{(1 - IoU) + v} \quad (3.14)$$

Burada,

b, b^{gt} : B ve B^{gt} 'nin merkez noktaları

$\rho(\cdot)$: Öklid mesafesidir

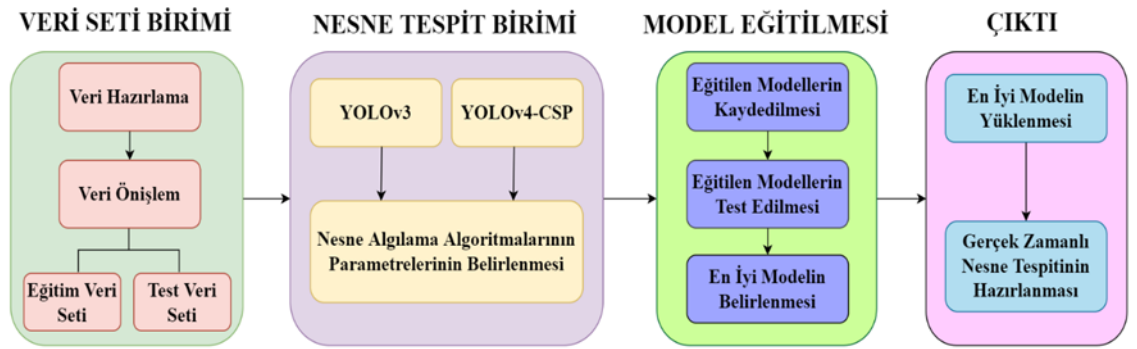
c: iki kutuyu kaplayan en küçük kapalı kutunun köşegen uzunluğudur.

α : pozitif bir deęiřtirme parametresi

v : en boy oranının tutarlılıęını ölçer

3.6. Türk İřaret Dilinin Makine Öğrenmesi ile Tespiti

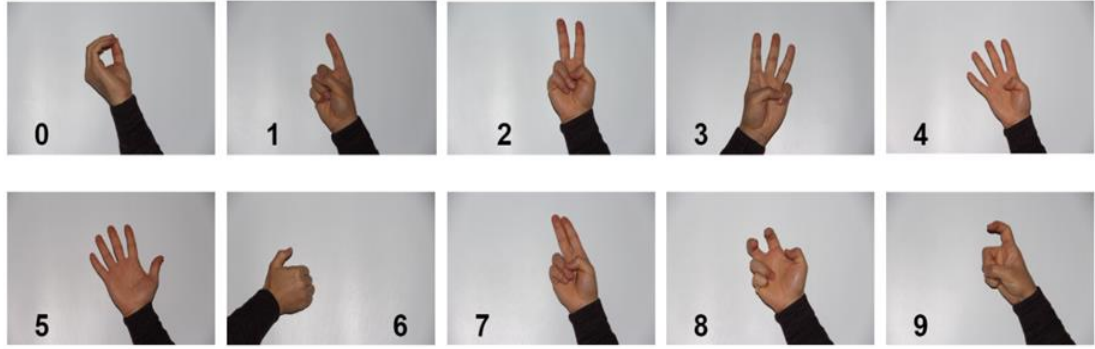
Türk İřaret Dili tanıma ile ilgili yaptığımız tez çalışmamızda takip ettiğimiz yol Şekil 3.24'te ayrıntılı olarak gösterilmiştir. İlk olarak Türk İřaret Diline ait 10 tane rakamı içeren görüntüler hazırlandı ve bir takım görüntü ön işlem çalışmaları yapıldıktan sonra model eğitiminde kullanılacak veri setleri oluşturuldu. Daha sonra, çıkan sonuçları karşılařtırmak için belirlediğimiz YOLOv3 ve YOLOv4-CPS nesne tespit algoritmaları oluşturulup aę parametreleri belirlendi. Belirlenen iki modelde eğitim ve test veri ile eğitilerek en iyi model tespit edildi. Son aşama olan çıktı kısmında ise bir kamera yardımıyla gerçek zamanlı nesne tespiti yapıldı.



Şekil 3.24. Sistem akış diyagramı

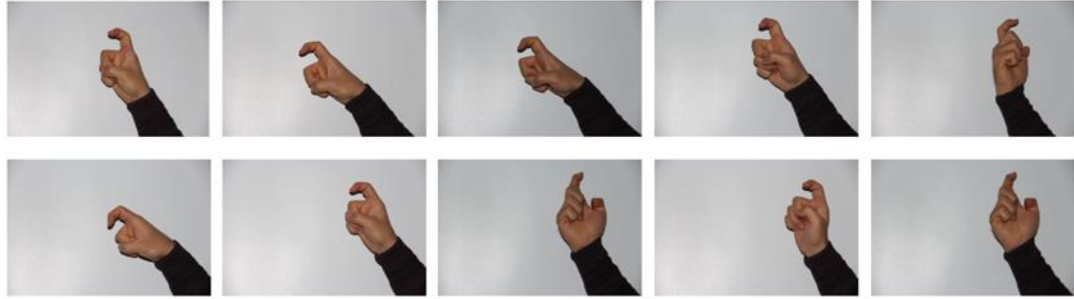
3.6.1. Veri seti

Veri seti, Türk iřaret diline özgü rakamları temsil eden 0'dan 9'a kadar olan rakamların el hareketlerini içeren görüntülerinden oluşmaktadır. Tüm görüntüler, 18 megapiksel fotoğraf çekebilen CANON 700d fotoğraf makinesi ile gönüllü olarak poz veren 6 farklı kiřiden çekilmiştir. Veri setinde ki her resim orijinali olan 5184x3456 boyutları kullanılmıştır. Şekil 3.25'te veri setinin bir parçası olan Türk iřaret dilindeki rakam hareketlerinin RGB görüntüleri gösterilmektedir.



Şekil 3.25. Türk işaret dili rakamları

Her bir kişiden, deneysel veri setinin daha çeşitli ve temsili olması için el pozisyonunda ve açısında sürekli bir değişiklik yapılarak her bir rakam için 25 farklı görüntü sağlanmıştır. Aynı rakama ait farklı el görüntüsü örnekleri Şekil 3.26’da verilmiştir. Bu yolla 6 kişiden toplam 1500 görüntü elde edilerek veri seti oluşturulmuştur.

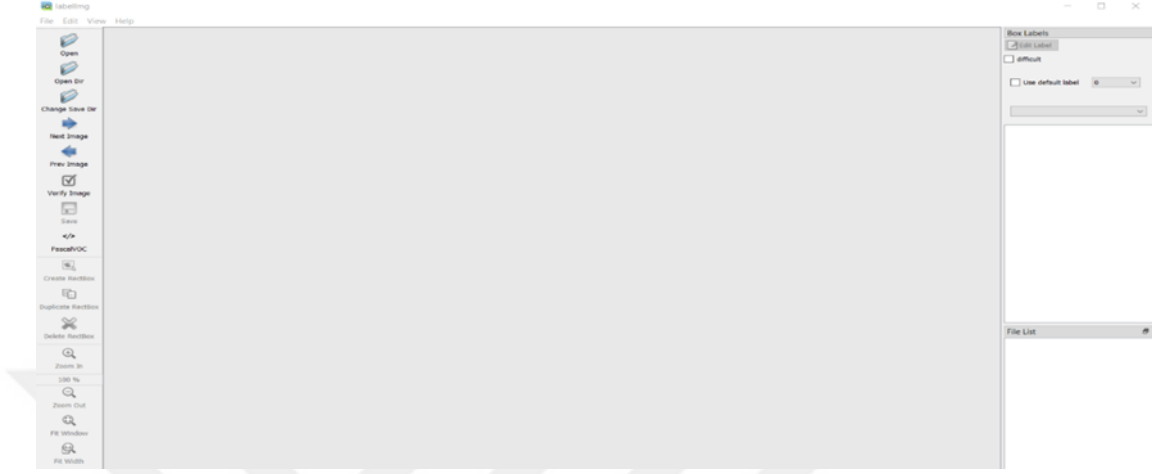


Şekil 3.26. Türk işaret dilinde 9 rakamına ait farklı el işaretleri

3.6.2. Veri etiketleme

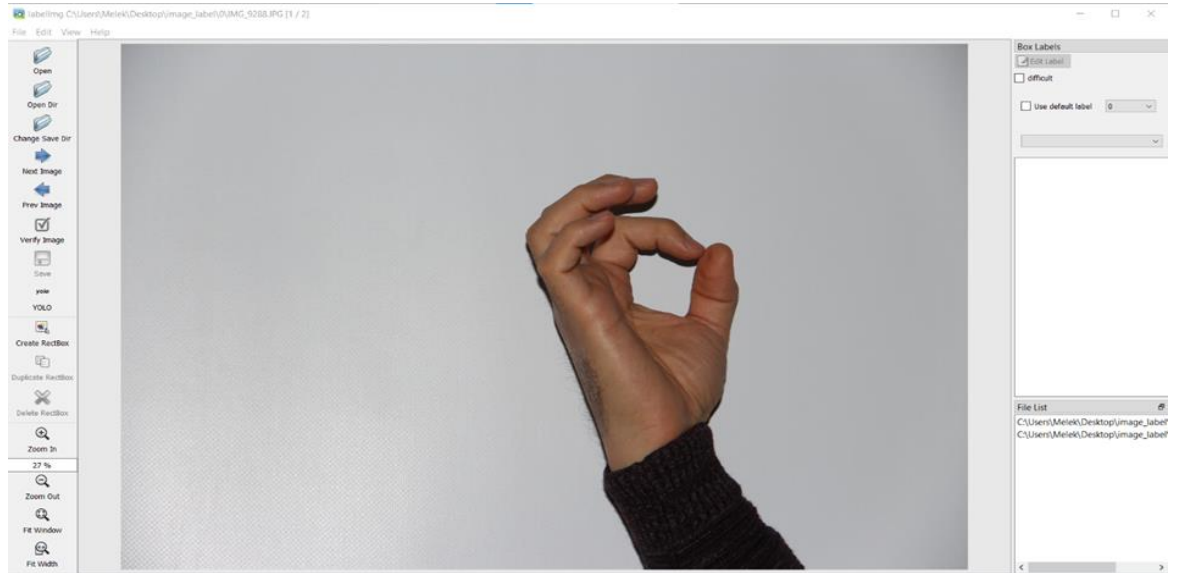
Sınıflandırma işlemi manuel olarak gerçekleştirilen bir işlemdir ve söz konusu nesneler tek tek görüntüde bulunur ve etiketlenir. Bu çalışmada, Türk İşaret dili veri setinde etiketleme işlemi için bir grafik görüntü açıklama aracı olan LabelImg yazılımını (Windows sürümü) (<https://github.com/tzutalin/labelImg>) kullanılmıştır. LabelImg, görüntülerdeki etiketli nesnelerin kutularının sınırlandırılmasına izin verir ve her görüntü için etiket verilerini içeren bir .txt dosyası kaydeder. Ayrıca, kullanılan yazılım, görüntüdeki bir veya birden çok etiketleme işlemine izin verecek özelliktedir.

Veri etiketleme için kullandığımız LabelImg etiketleme programının ara yüzü şekil 3.27’deki gibidir.



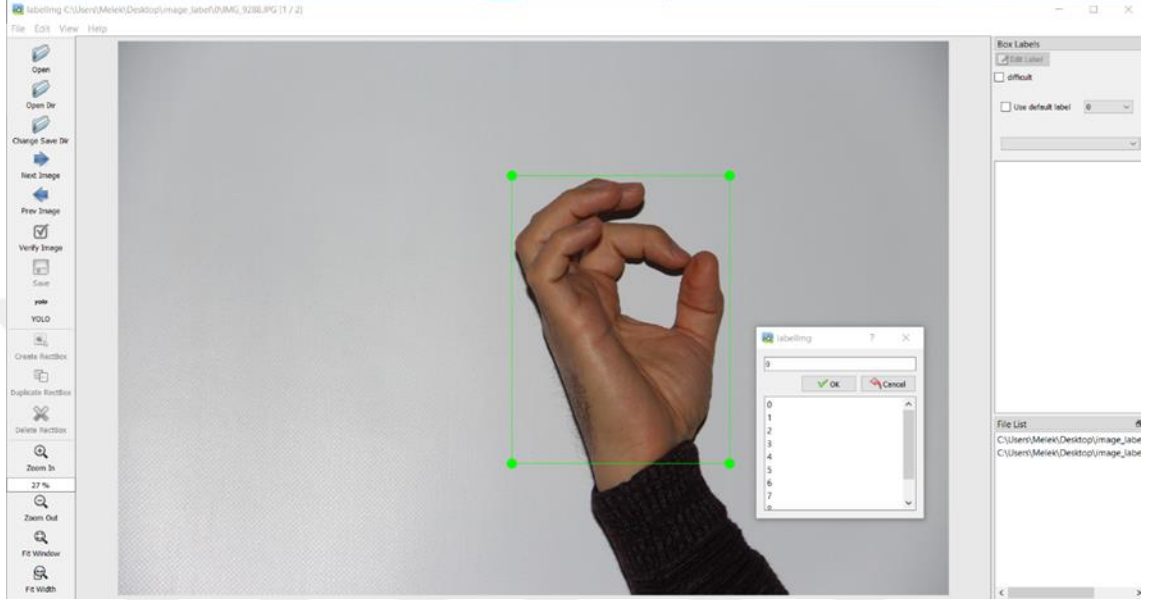
Şekil 3.27. LabelImg ara yüzü

Programın sol üst köşesinde bulunan ‘Open Dir’ seçilerek etiketleme yapacak görüntü dosyası seçilir. Görüntü dosyası seçtikten sonra Şekil 3.28’de gösterildiği gibi etiketleme yapacak görüntüler ekrana gelir.



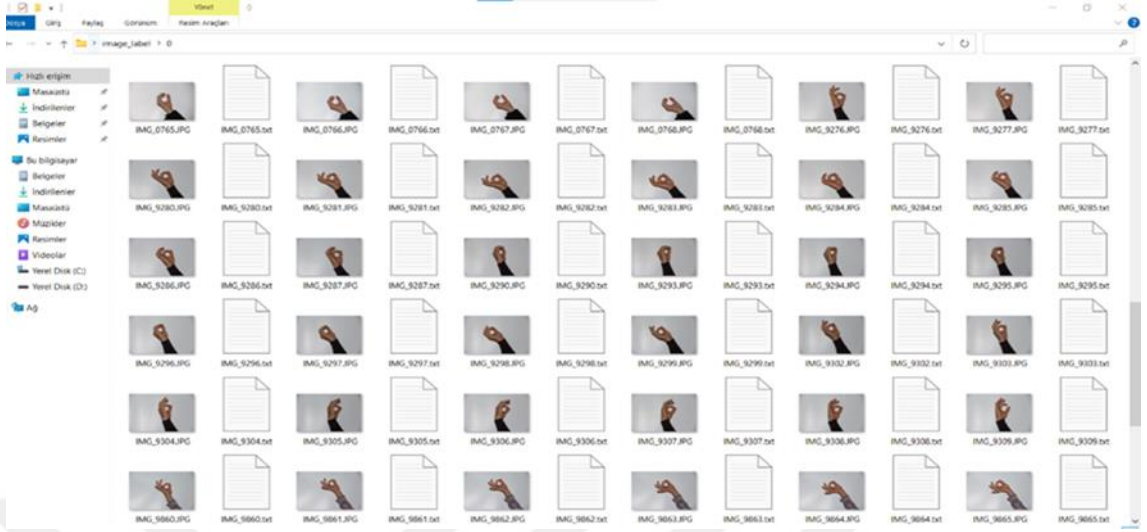
Şekil 3.28. LabelImg ara yüzündeki görüntü

Görüntülerdeki ilgili el işareti Şekil 3.29'da gösterildiği gibi kutu içine alınarak uygun etiket verilir. Veri setindeki tüm görüntüler için etiketleme işlemi aynı şekilde yapılır.



Şekil 3.29. LabelImg'de verinin etiketlenmesi

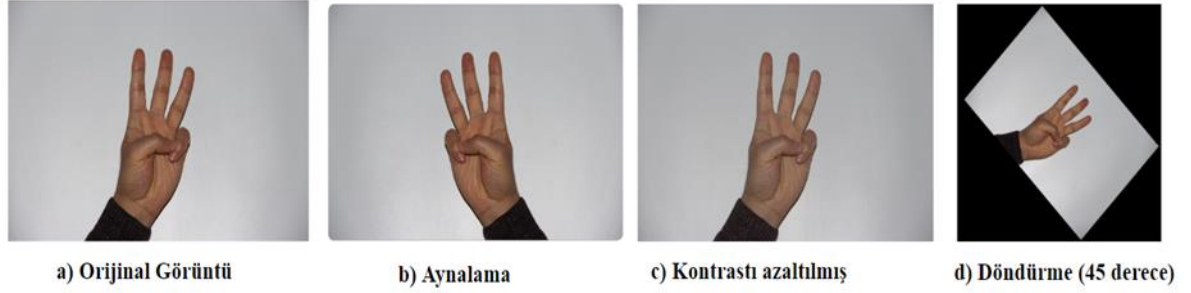
Etiketlemesi yapılan her bir görüntü Şekil 3.30'daki gibi birer .txt dosyası oluşturularak kaydedilir. Görüntüler ve onlara ait oluşturulan her bir .txt dosyası YOLO algoritmasının veri işleme birimine gönderilir. Böylece YOLO'da eğitimin ilk aşaması başlatılmış olur.



Şekil 3.30. Görüntülerin .txt dosyaları

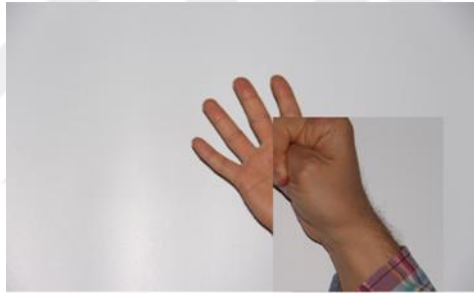
3.6.3. Veri arttırma

Derin sinir ağlarında, aşırı öğrenmenin önüne geçerek ağın iyi bir performans göstermesi için veri setinin büyüklüğü önemli bir etkidir. Bu yüzden, nesne algılama algoritmalarında veri büyütme teknikleri çok önemlidir. Veri seti hazırlarken Türk işaret diline ait her rakamdan 25 farklı görüntü kullanıldığı daha önce belirtilmişti. Bu veri seti tek başına yetersiz olduğu için modelin test verileri üzerinde genelleme kabiliyetini arttırmak için veri büyütme teknikleri kullanıldı. Türk işaret dili ile yaptığımız çalışmada veri çeşitliliğini arttırmak amacı ile; kontrast ayarı, histogram eşitleme, döndürme, aynalama, parlaklık gibi veri büyütme teknikleri de kullanılmıştır. Şekil 3.31’de veri seti yapılan bazı veri arttırma teknikleri verilmiştir. Ağın öğrenmesini daha iyi hale getirmek için bu tarz teknikler ile elde edilen yeni görüntüler de eğitim sırasında kullanılmaktadır.



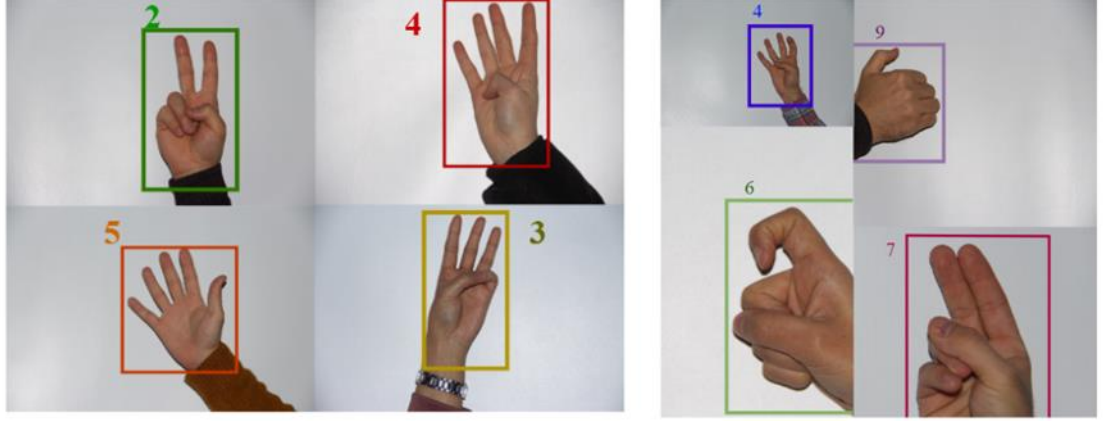
Şekil 3.31. Veri arttırma teknikleri

Veri artırma tekniklerine ek olarak YOLOv4 algoritmasında yer alan mozaik ve cutmix veri büyütme teknikleri de eğitim sırasında kullanıldı. Bu teknikler Şekil 3.32 ve Şekil 3.33'te gösterilmektedir. Şekil 3.33'te gösterilen Cutmix, 2 görüntüden belli kısımlar alır ve bunları birbirine yama yaparak bir görüntü elde eder.



Şekil 3.32. Cutmix veri artırma tekniği

Mozaik tekniği ise Şekil 3.33'te gösterildiği gibi 4 orijinal giriş görüntüsünü belirli oranlarda ya da alarak tek bir görüntüde birleştirir. Bu teknikte görüntünün dört köşesi ayrı ayrı bulur ve ardından veri arttırma işlemi bu köşelere uygulanır. Oluşan görüntüde aranan el işareti tam olarak bulunmayabilir. Bu, oluşan çerçevenin farklı bölümlerinde farklı görüntüleri lokalize ederek aranan el işaretini normal bağlamlarının dışında algılanmasını sağlar. Tüm bu veri arttırma teknikleri, konuşma esnasında oluşabilecek her türlü elin ve parmakların açısı değişikliği, konumu gibi faktörler düşünülerek seçilmiştir. Ayrıca, bu işlemler sırasında kendini tekrar eden veya piksellerde bozulmalar tespit edilen görüntüler de veri setinden temizlenmiştir.



Şekil 3.33. Mozaik Veri Artırma Tekniği

3.6.4. Değerlendirme metrikleri ve uygulama detayları

3.6.4.1. Deneysel kurulum

Bu çalışmadaki tüm deneyler, Ubuntu 20.04 ile bir Linux işletim sistemine sahip olan yüksek güçte bir bilgisayar ile yapılmıştır. Derin öğrenme modellerini eğitmek ve test etmek için kullanılan bu bilgisayarın özellikleri şu şekildedir: Intel Core i9 9900X (10 çekirdekli 3.50 GHz, 19.25 MB Intel® Smart Cache) işlemci, 32 GB DDR4 RAM ve tekli RTX 2080TI (11 GB GDDR6) 4352 cuda çekirdekli grafik kartı. NVIDIA CUDA Toolkit 11.1 ve NVIDIA GPU-Hızlandırılmış Kitaplık (cuDNN) 8.1 ile PyTorch ve DarkNet çerçevelerinin en son kararlı sürümü kullanılarak deneyler yapılmıştır.

Model tasarlanırken eğitim sürecinde iterasyon parametre güncellemeleri için Adam (Kingma and Ba, 2015) optimize edici kullanıldı. Adam optimizasyonu, derin öğrenme modellerini eğitmek için değişken gradyan inişleri için kullanılan bir algoritmadır. Adam, daha hızlı hesaplama süresi, uygun optimasyon için daha az parametre ihtiyacına sahip olması ve az bellek gereksinimine ihtiyaç duymasından dolayı tercih edildi. Yapılan çalışmada, momentum 0.937, ağırlık azalması 0.0005 ve diğer parametreler ise orijinal YOLOv4 referans makalesinde belirtilen değerler ile aynı şekilde kullanılmıştır.

3.6.4.2. El işareti algılama ve metrikleri değerlendirme

YOLO algoritmalarında nesne tespiti, giriş görüntüsü SxS ızgaralara bölünerek üretilen ızgaralar içinde hedef nesnenin bulunma olasılığını ifade eden güven puanı üretilerek yapıldığını daha önceki bölümlerde belirtilmişti. İşaret dili algılamalarında

kullanıcılar tarafından gösterilen el işaretleri hedef nesne konumundadır. Güven puanı, sınırlayıcı kutu içerisinde yer alan bu el işaretlerinin ne kadar doğru algılandığını göstermektedir. Gerçek zamanlı Türk işaret dili algılama sisteminde, yapılan el işaretinin başarılı bir şekilde tespit edilmediğinin belirlenmesine, daha önce 3.4.3.2 bölümünde değinilen IoU değeri ile karar verilir. IoU işlevinin değeri, tahmin edilen ve temel gerçek sınırlayıcı kutu arasındaki örtüşme ilişkisine bakılarak belirlenen bir eşik değerine göre belirlenir. Bu çalışma için, IoU eşik değeri 0,5 olarak ayarlandı. Tahmin edilen kutusu ve temel gerçek kutu arasındaki örtüşme 0,5'ten büyükse, eğitilen modelin tahmin ettiği el işaretinin doğru olduğu kabul edilir.

Derin öğrenme algoritmalarının performansını ölçmek için birçok metrik kullanılabilir. IoU işlevine ek olarak, bu çalışmada, gerçek pozitif (True Positive - TP), yanlış pozitif (False Positive - FP), yanlış Negatif (False Negative - FN) ve gerçek negatif (True Negative - TN) metrik değerleri de kullanıldı. Gerçek pozitif (TP), tahmin edilen sınırlayıcı kutunun el işaretinin temel gerçeğe düştüğünü göstererek modelin, el işaretini doğru tespit ettiği anlaşılır. Yanlış pozitif (FP), tahmin edilen sınırlayıcı kutunun el işaretinin temel gerçeğinin dışında kaldığını gösterir. Yanlış negatif (FN), tahmin edilen bir sınırlayıcı kutu olmadığını ancak çerçevenin bir el işareti içerdiğini ve dolayısıyla kaçırıldığını gösterir. Gerçek negatif (TN), el işareti içermeyen görüntülerde yapılan işaretin tespit edilmediğini göstermektedir. Veri setinde el işareti içermeyen görüntüler olmadığı için bu çalışmada bu metrik dikkate alınmamıştır. Bütün bu metrik değerler ile eğitilen modelin performansını değerlendirmek için sistemin kesinlik (precision), geri çağırma (recall), F1 skor değeri ve mAP değerlerine bakılması gerekir.

- Kesinlik (Precision): Doğru tahmin edilen pozitif sonuçların, TP değerlerinin, tahmin edilen tüm pozitif sonuçlara (TP+FP) oranı olarak hesaplanır. Kesinlik değeri formülü:

$$\text{Kesinlik} = \frac{TP}{TP + FP} \quad (3.15)$$

- Geri çağırma (Recall): Doğru tahmin edilen pozitif sonuçların oranını ölçmek için kullanılır. Geri çağırma aynı zamanda modelin hassasiyetini ölçer. Geri çağırma formülü:

$$\text{Geri çağırma} = \frac{TP}{TP + FN}$$

- F1 skor değeri: Kesinlik ve geri çağırma değerlerinin harmonik ağırlığını hesaplar. F1 skor formülü:

$$F1 = 2 \times \frac{\text{Kesinlik} \times \text{Geri çağırma}}{\text{Kesinlik} + \text{Geri çağırma}} \quad (3.17)$$

- mAP değeri: Ortalama kesinlik(Average precision- AP) değeri nesne algılama detektörleri için popüler bir metriktir. Her bir sınıf için AP değeri, 0 ile 1 arasındaki geri çağırma değeri için ortalama bir değer hesaplar. mAP değeri ise tüm AP değerlerinin ortalamasını tek bir sayı olarak veren bir değerdir. mAP değeri formülü:

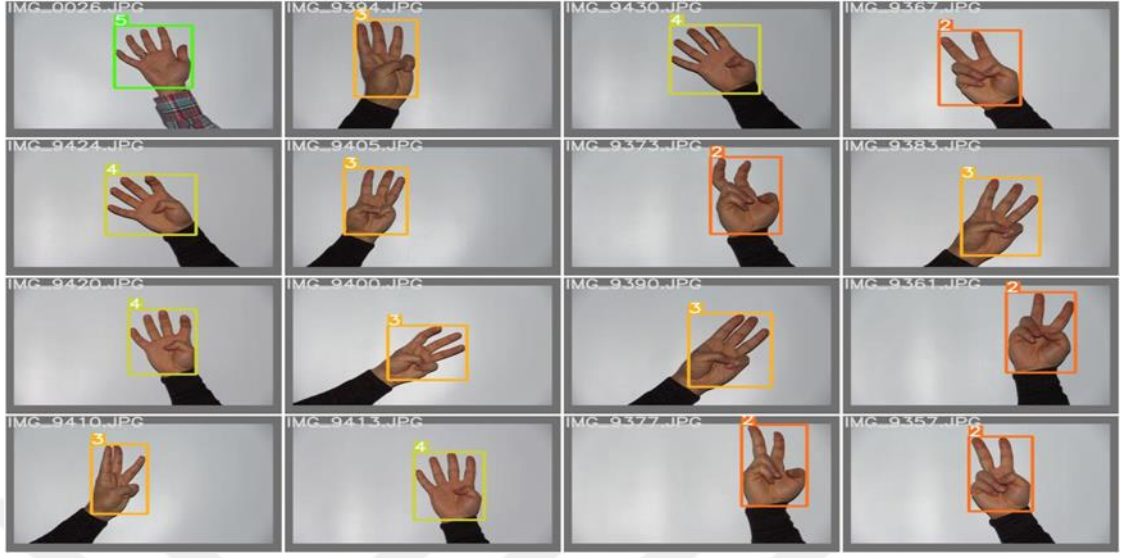
$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3.18)$$

4. BULGULAR ve TARTIŞMA

Geleneksel yöntemler kullanılarak yapılan işaret dili çeviri çalışmaları uzun süredir araştırmacılar tarafından üzerinde çalışılan bir konudur. Bu yöntemler, pratik kullanıma yönelik olmayıp sınırlı çalışma alanlarında verimlilik ve doğrulukları kanıtlanmış araştırma amaçlı çalışmalardır. Ancak, ESA'nın teknoloji hayatına girmesiyle mevcut işaret dilinin çalışma etkinliği de artmıştır. Her ne kadar ESA teknolojisinin birçok farklı işaret dilinde yaygın kullanımı görülmeye başlansa da literatürde Türk işaret dili tanıma sistemleri ile ilgili çok fazla sayıda çalışma bulunmamaktadır. Bu sebeple, bu tezde, Türk işaret dilinin makine öğrenmesi yardımı ile tespiti adına bir çalışma yapılmıştır. Bu tez çalışması, Türk işaret dili çevirisi için derin öğrenme yöntemlerini ve bu yöntemler içinde de özellikle YOLOv4'ü kullanmayı vurgulamaktadır. Bu çalışmada, çeviri süreci bir nesne algılama ve sınıflandırma görevi olarak görülmekte olup Türk işaret dili ile ilgili yapılmış olan mevcut literatür çalışmalarından farklı bir yaklaşım sunmaktadır.

4.1. Deneysel Sonuçlar

Bu çalışmada, gerçek zamanlı Türk işaret dili tespit sisteminde model eğitimi için Yolov3 ve Yolov4-CSP algoritmaları kullanılmıştır. Çalışmada kullanılan veri setinin %70'i eğitim, %10'u test ve %20'si ise doğrulama için kullanılmıştır. Model eğitime geçmeden önce nesne algılama ağının parametreleri belirlenir. Bu işlem yapıldıktan sonra model eğitime geçilir. Model eğitimi yapılırken her bir görüntü ve bunlara ait etiketler modele yüklenir. Eğitim boyunca model belirli bir iterasyon sonunda model hakkında kaç görüntü üzerinde çalıştığı, ne kadar sürede çalıştığı ve buna ait bir kayıp fonksiyon değeri verir. Model eğitim yaparken aynı zamanda test ve doğrulama veri setleri üzerinde nesne tanıma işlemi yaparak en iyi sonuçları bulmaya çalışır. Test ve doğrulamadaki görüntüler eğitimde kullanılan görüntülerden farklıdır. Bu farklılık sayesinde modelin doğruluğu test edilmiş olur.



Şekil 4.1. Örnek etiketli veri seti

Şekil 4.1 te eğitim sırasında modelin doğruluğunu bulmak için test ve doğrulama görüntülerinden oluşan Türk işaret diline ait olan bazı rakamlar içeren veri seti örneği gösterilmektedir. Model belirli bir iterasyon sayısı sonunda en az kayıp fonksiyon değerine ve en yüksek doğruluğa (ağırlığa) sahip değeri verir. Eğitim sonunda verilen örnek veri seti üzerinde Şekil 4.1.’deki gibi sonuçlara ulaşır. Bu sonuçlar ışığında, el işaretlerinde meydana gelen bilinçli bozulmalar neticesinde tahmin değerlerinde değişimler yaşansa da sayı tahminleri daima doğru olduğu gözlemlenmektedir.



Şekil 4.2. Tahmin değerleri verilen örnek veri seti

Türk işaret dilindeki rakamlarından oluşan veri seti üzerinde orijinal YOLOv3, YOLOv3-SPP ve YOLOv4-CSP modelleri ile eğitilerek en uygun nesne algılama model tespiti için performans değerlendirmesi yapılmıştır. Çalışma boyunca özellikle YOLOv4-CSP mimarisine odaklanılarak gerçek zamanlı bir tespit için hem doğruluk hem de hız açısından en uygun modelin haline getirilmeye çalışılmıştır. Sonuçları karşılaştırmalı değerlendirmek için modellere kesinlik (precision), geri çağırma (recall), F1-skor ve mAP değerleri Çizelge 4.1' de verilmiştir. Tüm modeller için IOU için 0,5'lik eşik değeri, güven eşik değeri 0,25 olarak alınarak değerlendirme yapılmıştır.

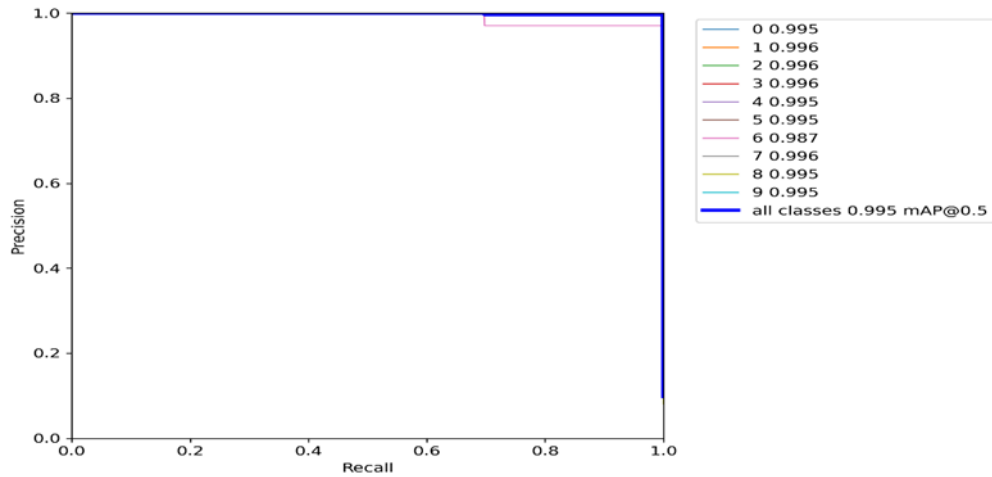
Çizelge 4.1.Türk işaret dili veri seti için sıfırdan eğitilmiş modeller ve sonuçları

Model	Precision %	Recall %	F1-score %	mAP (mAP@.5) %	Hız (milisaniye)
YOLOv3	96,56	94,45	95,49	97,13	10,3
YOLOv3-SPP	97,23	95,36	96,28	98,27	10,4
YOLOv4-CSP small	96,71	94,70	95,69	97,55	8,5
YOLOv4-CSP medium	97,53	95,76	96,63	98,67	8,6
YOLOv4-CSP large	98,84	97,27	98,04	99,32	9,8

YOLOv4-CSP modelinin Türk işaret dilindeki etkinliğini araştırma adına rakamlardan oluşan bir veri seti kullanılmıştır. Ayrıca YOLOv4-CSP modelinin diğer mevcut Evrişimli Sinir Ağları yöntemlerinden performans ve doğruluk bakımından farklılığının gösterilmesi amacı ile aynı veri seti YOLOv3 ve YOLOv3-SPP modelleri ile karşılaştırılmıştır. Çizelge 3.3'te elde edilen sonuçlarda görüldüğü gibi en düşük verimin YOLOv3 modelinden elde edildiği çok açıktır. Öte yandan, sonuçlardan yola çıkarak, YOLOv3 ile YOLOv4-CSP small ve YOLOv3-SPP ile YOLOv4-CSP medium yaklaşık olarak aynı doğruluk, geri çağırma, F1-skor değeri ve mAP değeri verdiği görülmektedir. YOLOv4-CSP modelinin ölçeklendirilmiş yapıları kendi aralarında kıyaslandığı zaman milisaniyelik fark olsa da YOLOv4-CSP small en hızlı model ve YOLOv4-CSP large ise yüksek doğruluğa sahip model olduğu anlaşılmaktadır. Fakat diğer modeller ile kıyaslandığında YOLOv4-CSP daha hızlı ve doğru sonuçlar verdiği gözlemlenmektedir. YOLOv4-CSP modelleri kendi aralarında değerlendirildiği zaman, ağır katman sayısı arttıkça milisaniye de olsa işlem hızında azalma olduğu fakat tahmin değeri olarak hala en yüksek değerde olduğu gözlemlenmektedir. YOLOv4-CSP,

orijinal YOLOv4'ten farklı olarak boyun kısmında da CSPNet kullanıldığına daha önce değinmiştik. CSPNet ağı, ağın algılama alanını arttırarak bu sayede tahmin değerlerinde daha yüksek değerler elde edilmesi sağlanmıştır. Bu çalışmalar neticesinde, kesinlik %98,84, geri çağırma 97,27, F1-skor değeri 98.04 ne mAP değeri ile 99,32 olarak elde edilmiştir.

Öte yandan CSPNet, hesaplama karmaşıklığının azaltılması için kullanılan bir tekniktir ve son derece verimli ve basit bir yapıya sahiptir. CSPNet, geniş bir alıcı alan sunduğu için ağın daha fazla anlamsal bilgi üretmesini sağlar. Bu yüzden CSPNet, YOLOv4 modeline eklenerek ağın daha iyi öğrenmesine katkı sağlanması amaçlanmıştır. Sonuçlar neticesinde, CSPNet'in hem omurga hem de boyun üzerinde kullanılmasının YOLOv3'e göre daha başarılı olduğu gösterilmiştir.



Şekil 4.3. YOLOv4-CSP kesinlik-geri çağırma grafiği

İlgili modelin Şekil 4.3'de kesinlik-geri çağırma grafiği incelediğinde, tüm rakamlar için grafik eğrileri aynı yerde kesişmektedir. Bu grafikten, YOLOv4-CSP modelinin mükemmel hassasiyet ve tüm eşik değerlerinde hatırlamayı sağlayan sonuçlar verdiğini ve modelin ideal bir sınıflandırıcı olduğu anlaşılmaktadır. YOLOv4 ve CSPNet algoritmalarının birlikte en iyi uyumu sağlayarak hem doğruluk hem de doğruluk-hız oranı açısından en iyi sonucu verdiğini gösteriyor.

Yapılan çalışmanın etkinliğini tartışmak için işaret dili tanıma sistemleri hakkında son yıllarda yapılmış çalışmalar ile bu çalışmada üzerinde çalışılan Türk işaret

dili tanıma modeli karşılaştırılmıştır. Buradaki amaç, YOLOv4-CSP'nin işaret dili üzerindeki etkisini incelemek ve başarısını literatürdeki diğer sistemler ile karşılaştırarak bir değerlendirme yapmaktır. Bu amaçla, son yıllarda farklı işaret dillerine ait yapılmış birkaç çalışma ve TİD'e yapılan bazı çalışmalar Çizelge 4.2'de verilmiştir.

Çizelge 4.2. İşaret dili ile ilgili yapılmış güncel çalışmalar

Yazar	İşaret Dili	Doğruluk (%)	Method	Özellik	Açıklama
Khari <i>et al.</i> (2019)	Amerikan İşaret Dili	94,8	İyi ayarlanmış VGG-19,	RGB ve RGB-D (RGB-Deep image) görüntülere VGG-19(v1-v2) uygulanarak bir özellik birleştirme katmanı oluşturuldu	Model eğitimi iki farklı aşamada gerçekleştirilmiş.
Ganggrade and Bharti (2020)	Hintçe İşaret dili	99,3	CNN, Kinect sensör	Kinect sensor yardımıyla algılama ve segmentasyon ön işlemini yapıldı	Arka plandan bağımsız çalışması, algılama için sensor gibi bir harici referansa ihtiyaç duyulması
Sincan ve Keleş (2020)	Türkçe İşaret Dili	95,95	ESA, LSTM	Microsoft Kinect v2 görüntülerin derinlikleri elde edildi.	Büyük Ölçekli Çok Modlu Türk İşaret Dili veri seti ile arka plan zenginliği ve kullanıcıdan bağımsız değerlendirme için benchmark eğitimleri ve test setleri hazırlandı.

Çizelge 4.2. Devamı.

Rivera-Acosta <i>et al.</i> , (2021)	Amerika n İşaret dili	81.74 (harf) 98.07(keli me)	DNN	YOLO, LSTM	Görüntüler iki ana işlemden geçiriliyor. El şekli segmentasyonu ve imla düzeltme özelliği eklemek için çift yönlü uzun kısa süreli bellekli sinir ağı uyguladılar.
Özcan ve ark., (2021)	Türk İşaret dili	93,93	GoogLe Net tabanlı ESA	Grid rastgele genetik algoritma yöntemleri kullanılmıştır. En iyi sonuç grid arama tekniği ile elde edilmiştir.	Hiperparametre optimizasyonu destekli ESA ile Erciyes Üniversitesi İşaret Dili Tanıma (ERUSLR) veri seti kullanılarak hastanelerin acil servislerinde sıkça kullanılan kelimeler ile çalışıldı.
Khan <i>et al.</i> , (2022)	Malezya işaret dili	90	ESA Tabanlı Dikkat Modülü (CBAM- Resnet)	Bloklar İçinde ve Sınıflandırıcıdan Önce yöntemlerini uygulayan iki farklı deney yaptılar.	19 dinamik işaret için 2071 video da en verimli sonuca Sınıflandırıcıdan Önce yöntemi ile ulaşılar

Çizelge 4.2. Devamı.

Huang <i>et al.</i> , - (2022)		89,9	CBAM (Convol ution Block Attentio n Module)	Kinect sensor ile hareketlerin yakalanması, BN katmanı, CNN- Bi-LSTM	Dikkat mekanizması ile evrimsel ağla birleştirilmesiyle kendi kendini kodlayan bir zamanlama ağı oluşturuldu.
Tez Çalışması	Türk işaret dili	99,32	YOLOv 4-CSP	YOLOv4 CSPNet birleştirildi	ve 9,8 milisn'de tahmin yapılarak gerçek zamanlı ve tahmin doğruluğu yüksek sonuçlar elde edildi.

Çizelge 4.2’de verilen çalışmalardan, Khari *et al.*, (2019) ve Rivera-Acosta *et al.*, (2021) işaret dili tanıma sistemi için yaptıkları çalışmada giriş görüntülerini bir önışlem aşamasından geçirmişlerdir. Sunduğumuz model çalışmasında giriş görüntüleri üzerinde herhangi bir ön işlem çalışması yapılmadan RGB görüntüler üzerinde çalışıldı. Bu da ilgili işaretin tanınması için işlem karmaşasının önüne geçerek sistemi basit hale getirmektedir. Ganggrade *et al.*, (2020) ve Sincan ve ark., (2020) önerdikleri yöntemlerde kullanıcıdan ilgili el işaretini yakalamak için kinect sensor kullanmışlardır. Bu modeller her ne kadar iyi sonuçlar verse de kullanımları esnasında harici bir referansa ihtiyaç duymaları kullanıcıyı kısıtlar. Türk işaret dilinde Özcan ve ark., (2021) ve Malezya işaret dilinde Khan *et al.*, (2022) ESA tabanlı yöntemler önermişlerdir. Ayrıca, Huang *et al.*, (2022) yaptıkları çalışmada sensor kullanarak ESA tabanlı bir tanıma sistemi sundular. Çizelgede verilen çalışmaları dikkatli incelediğimizde çoğunlukla görüntüleri algılamak ve işlemek için sensor ya da önışlem aşaması ve sınıflandırma olmak üzere iki aşamada gerçekleştirilmiştir. Türk işaret dilinde ilgili el işaretini algılanması için önerdiğimiz YOLOv4-CSP modeli tek aşamada gerçekleştirilerek diğer modellerden daha iyi bir performans göstermiştir.

Gerçek zamanlı Türk işaret dili tanıma sistemi için önerdiğimiz YOLOv4-CSP modeli, statik el işaretini bir nesne olarak algılayıp ilgili bölgeyi belirleme ve sınıflandırma işlemini eş zamanlı olarak yapmaktadır. Bu yüzden, YOLOv4-CSP sadece ilgili el işaretini algıladığı için arka plan karmaşıklığı sorunu yaşamamaktadır. Bu durum, algıda seçiciliğin arka plana bağımlılığını ortadan kaldırarak öğrenme sürecinde ön işlem ihtiyacını da ortadan kaldırmaktadır. Buna ek olarak, önerilen model video algılaması yaparak arka plana bağlı kalmadan el şeklini hızlı ve doğru bir şekilde tahmin edebilme olanağını sağlamaktadır. Bu da gerçek zamanlı uygulamalarda kullanmak için uygun olduğunu gösterir.



5. SONUÇ ve ÖNERİLER

Bu tez çalışması, YOLOv4-CSP nesne algılama algoritması kullanılarak Türk İşaret Dilindeki rakamların doğru ve kesin bir şekilde gerçek zamanlı olarak tespitini gerçekleştiren bir model oluşturmaya ve önermeye odaklanmıştır. Literatürde YOLOv4 öğrenme yöntemi ile Türk işaret dilini çalışan ilk tez olma özelliğini taşıyan bu çalışmada kullanılan eğitim veri seti özel olarak oluşturulmuştur. Veri setinde rakam işaretleri yapılırken önerilen modelin seçiciliğini de test etmek için el ve parmak konumlarının tüm olasılıkları düşünülerek oluşturulmuştur.

Bu çalışmada, gerçek zamanlı performans gösteren son teknoloji nesne algılama algoritması olan YOLOv4 mimarisi optimize edilerek kullanılmıştır. Algoritmanın performansının artırılması için CSPNet ağı, YOLOv4 mimarisinin boyun kısmına uygulanmıştır. Yapılan bu uygulama ile ağın öğrenme alanı artırılmış ve böylece modelin öğrenme performansı daha da iyileştirilmiştir. Önerilen modelin ne kadar hızlı ve doğru tahmin ettiğini tespit etmek için var olan veri seti YOLOv3 ve YOLOv3-SPP modellerine de uygulanarak sonuçlar karşılaştırılmıştır. Bu karşılaştırma sonucunda, YOLOv4-CSP (Scaled-YOLOv4) algoritmasının performansının diğer algoritmaların performansından çok daha iyi olduğu gösterilmiştir.

Sonuç olarak, giderek artan dünya nüfusu ile birlikte sağır ve dilsiz bireylerin sayısı da artmaktadır. İşaret dilinin yaygın bir şekilde bilinmemesi ve bu bireylerin bir tercümana bağlı kalmadan günümüz dünyasına ayak uydurmaları açısından işaret dili çeviri sistemleri çok önemli bir yere sahiptir. İşaret dilinin gerçek zamanlı tespiti için bu çalışmada önerilen yöntem, harici bir donanıma ihtiyaç duyulmadan akıllı telefon, tablet gibi teknolojik aletlerde kullanılmasına imkân sağlayarak engelli bireylere günlük yaşamda yardımcı olabilmeyi sunmaktadır. Türk işaret dili için literatüre ilk defa kazandırılan bu modelin, ağın öğrenme kapasitesi de göz önüne alınarak daha büyük veri seti ile eğitilebilme potansiyelinden dolayı gelecek çalışmalar için önemli bir referans olacağı açıktır.

KAYNAKÇA

- Agarap, A. F. (2018). *Deep Learning using Rectified Linear Units (ReLU)*. arXiv:1803.08375. adresinden alındı
- Aksoy, B., Salman, O. K., & Ekrem, Ö. (2021). *Detection of Turkish Sign Language Using Deep Learning and Image Processing Methods*. Applied Artificial Intelligence, 35(12), 952-981.
- Al-Jarrah, O., & Halawani, A. (2001). *Recognition of gestures in Arabic sign language using neuro-fuzzy systems*. Artificial Intelligence, 133(1-2), 117-138.
- Almeida, S. G., Guimarães, F. G., & Ramirez, J. A. (2014). *Feature extraction in Brazilian Sign Language Recognition based on phonological structure and using RGB-D sensors*. Expert Systems with Applications: An International Journal, 14(6), 7259–7271.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Ayad Al-Dujaili, Y. D., Al-Shamma, O., Santamaría, J., . . . Farhan, L. (2021). *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*. Journal of big Data, 8(1), 1-74.
- Arslan, Y. (2021). *Geçmişten Günümüze İşitme Engellilerin Eğitimi ve Türk İşaret Dilinin Rolü*. Akademik Tarih ve Düşünce Dergisi, 8(1), 163-178.
- Baheti, P. (2022, 5 22). *A Comprehensive Guide to Convolutional Neural Networks*. <https://www.v7labs.com/blog/convolutional-neural-networks-guide> adresinden alındı
- Bauer, B., & Hienz, H. (2000). *Relevant features for video-based continuous sign language recognition*. Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580) (s. 440-445). IEEE.
- Bochkovski, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. arXiv:2004.10934. adresinden alındı

- Bordes, A., Glorot, X., Weston, J., & Bengio, Y. (2012). ***Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing***. Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (s. 127-135). PMLR.
- Brashear, H., Henderson, V., Park, K.-H., Hamilton, H., Lee, S., & Starner, T. (2006). ***American sign language recognition in game development for deaf children***. Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility , (s. 79-86).
- Chaudhuri, S., Dayal, U., & Narasayya, V. (2011). ***An overview of business intelligence technology***. Communications of the ACM, 54(8), 88-98.
- Chen, F.-S., Fu, C.-M., & Huang, C.-L. (2003). ***Hand gesture recognition using a real-time tracking method and hidden Markov models***. Image and vision computing, 21(8), 745-758.
- Convolutional Neural Network***. (tarih yok). 4 5, 2022 tarihinde Mathwork: <https://ch.mathworks.com/discovery/convolutional-neural-network-matlab.html> adresinden alındı
- Dalal, N., & Triggs, B. (2005). ***Histograms of oriented gradients for human detection***. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 1, s. 886-893.
- Deafness and hearing loss***. (2021, Nisan 1). Word Health Organization(WHO): <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss> adresinden alındı
- Demirel, F. (2013). ***Osmanlı'da sağır-dilsiz ve âmâların eğitimi: dilsiz ve âmâ mektebi***. İstanbul: İdeal Kültür & Yayıncılık.
- Deng, J., Xuan, X., Wang, W., Li, Z., Yao, H., & Wang, Z. (2020). ***A review of research on object detection based on deep learning***. Journal of Physics: Conference Series , 1684(1), 012028.

- Deng, L., & Yu, D. (2014). ***Deep Learning: Methods and Applications***. Foundations and Trends in Signal Processing, 7(3-4), 197–387.
- Elakkiya, R. (2020). ***Machine learning based sign language recognition: a review and its research***. JOURNAL OF AMBIENT INTELLIGENCE AND HUMANIZED COMPUTING, 12, s. 7205–7224.
- Erten, H., & Arıcı, N. (2022). ***İşaret Dilinin Tarihi Serüveni ve Türk İşaret Dili***. Afyon Kocatepe Üniversitesi Sosyal Bilimler Dergisi, 24(1), 1-14.
- Fan, J., Ma, C., & Zhong, Y. (2019). ***A selective overview of deep learning***. arXiv:1904.05526 . adresinden alındı
- Fang, G., Gao, W., & Zhao, D. (2007). ***Large-Vocabulary Continuous Sign Language Recognition Based on Transition-Movement Models***. IEEE transactions on systems, man, and cybernetics-part a: systems and humans, 37(1), 1-9.
- Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). ***Object Detection with Discriminatively Trained Part-Based Models***. IEEE Transactions on Pattern Analysis and Machine Intelligence , 32(9), 1627 - 1645.
- Fu, L., Duan, J., Zou, X., Lin, J., Zhao, L., Li, J., & Yang, Z. (2020). ***Fast and Accurate Detection of Banana Fruits in Complex Background Orchards***. IEEE Access , 8, 196835 - 196846.
- Gangrade, J., & Bharti, J. (2020). ***Vision-based Hand Gesture Recognition for Indian Sign Language Using Convolution Neural Network***. IETE Journal of Research, 1-10.
- Girshick, R. (2015). ***Fast R-CNN***. Proceedings of the IEEE international conference on computer vision , (s. 1440-1448).
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). ***Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation***. Proceedings of the IEEE conference on computer vision and pattern recognition , (s. 580-587).
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). ***Deep Learning (Adaptive Computation and Machine Learning Series)***. The MIT Press.

- GÖK, S. (1958). *Dünyada ve Türkiyede sağır, dilsiz okulları tarihçesi ve eğitim sistemi*. Hüsnütabiat Matbaası.
- Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & S.Lew, M. (2016). *Deep learning for visual understanding: A review*. Neurocomputing, 187, 27-48.
- Gustineli, M. (2022). *A survey on recently proposed activation functions for Deep Learning*. arXiv:2204.02921 . adresinden alındı
- Günay, R., & Görür, H. İ. (2013). *Osmanlı Devleti'nde sağır, dilsiz ve a'mâ mektebi*. Tarih Araştırmaları Dergisi, 32(53), 55-76.
- Güncel Türk İşaret Dili Sözlüğü*. (2021). <http://tidsozluk.net/> adresinden alındı
- Gündüz, C., & POLAT, H. (2021). *Turkish Sign Language Recognition Based on Multistream Data Fusion*. Turkish Journal of Electrical Engineering & Computer Sciences, 29(2).
- Han, J., Awad, G., & Sutherland, A. (2009). *Modelling and segmenting subunits for sign language recognition based on hand motion analysis*. Pattern Recognition Letters, 30(6), 623-633.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9), 1904 - 1916.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (s. 770-778).
- Huang, J., Zhou, W., Li, H., & Li, W. (2015). *Sign Language Recognition using 3D convolutional neural networks*. 2015 IEEE International Conference on Multimedia and Expo (ICME). IEEE.
- Huang, Y., Huang, J., Wu, X., & Jia, Y. (2022). *Dynamic Sign Language Recognition Based on CBAM with Autoencoder Time Series Neural Network*. Mobile Information Systems, 2022.

- Ioffe, S., & Szegedy, C. (2015). ***Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift***. Proceedings of the 32nd International Conference on Machine Learning. 37, s. 448-456. PMLR.
- Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). ***A Survey of Deep Learning-Based Object Detection***. IEEE Access, 7, 128837 - 128868.
- Jocher, G. (2021). ***YOLOV5-Master***. 6 18, 2022 tarihinde github: <https://github.com/ultralytics/yolov5> adresinden alındı
- Justesen, N., Bontrager, P., Togelius, J., & Risi, S. (2020). ***Deep Learning for Video Game Playing***. IEEE Transactions on Games, 12(1), 1 - 20.
- Ke, Q., Liu, J., Bennamoun, M., An, S., Sohel, F., & Boussaid, F. (2018). ***Computer Vision for Human-Machine Interaction***. Computer Vision for Assistive Healthcare (s. 127-145). Academic Press.
- Khan, R., Wong, W. S., Ullah, I., Algarni, F., Haq, M. I., Barawi, M. H., & Khan, M. A. (2022). ***Evaluating the Efficiency of CBAM-Resnet Using Malaysian Sign Language***. CMC-COMPUTERS MATERIALS & CONTINUA, 71(2), 2755-2772.
- Khari, M., Garg, A. K., Crespo, R. G., & Verdú, E. (2019). ***Gesture Recognition of RGB and RGB-D Static Images Using Convolutional Neural Networks***. INTERNATIONAL JOURNAL OF INTERACTIVE MULTIMEDIA AND ARTIFICIAL INTELLIGENCE, 5(7), 22-27.
- Khomami, S. A., & Shamekhi, S. (2021). ***Persian sign language recognition using IMU and surface EMG sensors***. Measurement, 168, 108471. doi:10.1016/j.measurement.2020.108471
- Kim, S., Ji, Y., & Lee, K.-B. (2018). ***An Effective Sign Language Learning with Object Detection Based ROI Segmentation***. 2018 Second IEEE International Conference on Robotic Computing (IRC) (s. 330-333). IEEE.
- Kingma, D. P., & Ba, J. (2015). ***Adam: A Method for Stochastic Optimization***. 3. International Conference on Learning Representations (ICLR), (s. 1-15).

- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). **ImageNet Classification with Deep Convolutional Neural Networks**. Advances in Neural Information Processing Systems 25.
- Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). **Feature Pyramid Networks for Object Detection**. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (s. 2117-2125).
- Mao, C., Huang, S., Li, X., & Zhongfu, Y. (2017). **Chinese Sign Language Recognition with Sequence to Sequence Learning**. CCF Chinese Conference on Computer Vision (s. 180–191). Springer.
- Memiş, A., & Albayrak, S. (2013). **Turkish Sign Language recognition using spatio-temporal features on Kinect RGB video sequences and depth maps**. 2013 21st Signal Processing and Communications Applications Conference (SIU) (s. 1-4). IEEE.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). **Distributed Representations of Words and Phrases and their Compositionality**. Advances in neural information processing systems, 26.
- Misra, D. (2019). **Mish: A Self Regularized Non-Monotonic Activation Function**. arXiv:1908.08681 . adresinden alındı
- Moghaddam, M., Nahvi, M., & Pourmomtaz, N. (2020). **Fast recognition and classification of static and dynamic signs for Persian sign language**. International Journal of Computer Applications in Technology, 63(3).
- Naglot, D., & Kulkarni, M. (2016). **Real time sign language recognition using the leap motion controller**. International conference on inventive computation technologies (ICICT). 3, s. 1-5. IEEE.
- Nakjai, P., Maneerat, P., & Katanyukul, T. (2019). **Thai finger spelling localization and classification under complex background using a YOLO-based deep learning**. Proceedings of the 11th International Conference on Computer Modeling and Simulation, (s. 230-233).

- Nimisha, K., & Jacob, A. (2020). **A Brief Review of the Recent Trends in Sign Language Recognition.** In 2020 International Conference on Communication and Signal Processing (ICCSP) (s. 186-190). IEEE.
- Ongsulee, P. (2017). **Artificial intelligence, machine learning and deep learning.** 2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE) (s. 1-6). IEEE.
- Özcan, T., & Baştürk, A. (2021). **ERUSLR: A new Turkish sign language dataset and its recognition using hyperparameter.** Journal of the Faculty of Engineering and Architecture of Gazi University, 36(1), 527-542.
- Pacal, I., & Karaboga, D. (2021). **A robust real-time deep learning based automatic polyp detection system.** Computers in Biology and Medicine, 134, 104519.
- Pacal, I., Karaboga, D., Basturk, A., Akay, B., & Nalbantoglu, U. (2020). **A comprehensive review of deep learning in colon cancer.** Computers in Biology and Medicine, 126.
- Pacal, I., Karaman, A., Karaboga, D., Akay, B., Basturk, A., Nalbantoglu, U., & Coskun, S. (2022). **An efficient real-time colonic polyp detection with YOLO algorithms trained by using negative samples and large datasets.** Computers in Biology and Medicine, 141, 105031.
- Rastgoo, R., Kiania, K., & Escalerab, S. (2021). **Sign Language Recognition: A Deep Survey.** Expert Systems with Applications, 164, 113794. doi:10.1016/j.eswa.2020.113794
- Redmon, J., & Farhadi, A. (2017). **YOLO9000: Better, Faster, Stronger.** Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (s. 7263-7271).
- Redmon, J., & Farhadi, A. (2018). **YOLOv3: An Incremental Improvement.** arXiv:1804.02767 . adresinden alındı
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). **You Only Look Once: Unified, Real-Time Object Detection.**

- Ren, J., & Wang, Y. (2022). **Overview of object detection algorithms using convolutional neural networks**. Journal of Computer and Communications, 10(1), 115-132.
- Ren, S., He, K., Girshick, R., & Sun, J. (2016). **Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks**. Advances in neural information processing systems 28.
- Ren, Z., Yuan, J., Meng, J., & Zhang, Z. (2013). **Robust Part-Based Hand Gesture Recognition Using Kinect**. IEEE Transactions on Multimedia, 15(5), 1110 – 1120.
- Rezende, T. M., Almeida, S. G., & Guimarães, F. G. (2021). **Development and validation of a Brazilian sign language database for human gesture recognition**. Neural Computing and Applications, 33(16), 10449-10467.
- Rivera-Acosta, M., Ruiz-Varela, J. M., Cisneros, S. O., & Dominguez, J. R. (2021). **Spelling Correction Real-Time American Sign Language Alphabet Translation System Based on YOLO Network and LSTM**. Electronics, 10(9), 1035.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., . . . Fei-Fei, L. (2015). **ImageNet Large Scale Visual Recognition Challenge**. International journal of computer vision, 115(3), 211-252.
- Shinde, P. P., & Shah, S. (2018). **A Review of Machine Learning and Deep Learning Applications**. 2018 Fourth international conference on computing communication control and automation (ICCUBEA) (s. 1-6). IEEE.
- Shukor, A. Z., Miskon, M. F., Jamaluddin, M. H., binAli@Ibrahim, F., FareedAsyraf, M., & binBahar, M. B. (2015). **A new data glove approach for Malaysian sign language detection**. Procedia Computer Science, 76, 60-67.
- Sincan, O. M., & Keles, H. Y. (2020). **AUTSL: A Large Scale Multi-Modal Turkish Sign Language Dataset and Baseline Methods**. IEEE ACCESS(8), 181340-181355.

- Starner, T. E., Weaver, J., & Pentland, A. (1998). *Real-time American sign language recognition using desk and wearable computer based video*. IEEE Transactions on pattern analysis and machine intelligence, 20(12), 1371-1375.
- T.C. Aile ve Sosyal Hizmetler Bakanı. (2021). [https://www.aile.gov.tr/haberler/aile-ve-sosyal-hizmetler-bakanliginca-turkiyenin-ilk-isitme-engelliler-iletisim-merkezi-acilacak/ adresinden alındı](https://www.aile.gov.tr/haberler/aile-ve-sosyal-hizmetler-bakanliginca-turkiyenin-ilk-isitme-engelliler-iletisim-merkezi-acilacak/adresinden-alindi)
- Tateno, S., Liu, H., & Ou, J. (2020). *Development of Sign Language Motion Recognition System for Hearing-Impaired People Using Electromyography Signal*. Sensors, 20(20), 5807.
- Türk İşaret Dili Sözlüğü*. (2015). 05 21, 2022 tarihinde T.C. MEB Özel Eğitim ve Rehberlik Hizmetleri Genel Müdürlüğü,: http://orgm.meb.gov.tr/meb_iys_dosyalar/2015_06/25021547_tidsozlukortaboyut_paral.pdf adresinden alındı
- Viola, P., & Jones, M. (2001). *Rapid object detection using a boosted cascade of simple features*. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. 1, s. 1. IEEE.
- Wang, C., Gao, W., & Shan, S. (2002). *An approach based on phonemes to large vocabulary Chinese sign language recognition*. Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition (s. 411-416). IEEE.
- Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2021). *Scaled-YOLOv4: Scaling Cross Stage Partial Network*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (s. 13029-13038).
- Wang, C.-Y., Liao, H.-Y. M., Yeh, I.-H., Wu, Y.-H., Chen, P.-Y., & Hsieh, J.-W. (2020). *CSPNet: A New Backbone that can Enhance Learning Capability of CNN*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (s. 390-391).
- Wu, Y., & Huang, T. S. (1999). *Vision-Based Gesture Recognition: A Review*. In International gesture workshop (s. 103-115). Berlin Heidelberg: Springer.

- Xu, B., Wang, N., Chen, T., & Li, M. (2015). ***Empirical Evaluation of Rectified Activations in Convolutional Network***. arXiv:1505.00853 . adresinden alındı
- Yakura, H., Shinozaki, S., Nishimura, R., & Oyama, Y. (2018). ***Malware Analysis of Imaged Binary Samples by Convolutional Neural Network with Attention Mechanism***. The 8th ACM Conference on Data and Application Security and Privacy, (s. 127-134).
- Ye, L., Lan, S., Zhang, K., & Zhang, G. (2020). ***EM-Sign: A Non-Contact Recognition Method Based on 24 GHz Doppler Radar for Continuous Signs and Dialogues***. Electronics , 9(10), 1577.
- Yu, J., Jiang, Y., Wang, Z., Cao, Z., & Huang, T. (2016). ***UnitBox: An Advanced Object Detection Network***. MM '16: Proceedings of the 24th ACM international conference on Multimedia, (s. 516–520).
- Yu, S., Jia, S., & Xu, C. (2017). ***Convolutional neural networks for hyperspectral image classification***. Neurocomputing, 219, 88-98.
- Yun, S., Han, D., Chun, S., Oh, S. J., Yoo, Y., & Choe, J. (2019). ***CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features***. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE.
- Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., & Presti, P. (2011). ***American sign language recognition with the kinect***. Proceedings of the 13th international conference on multimodal interfaces , (s. 279-286).
- Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). ***mixup: Beyond empirical risk minimization***. International Conference on Learning Representations,.
- Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). ***Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression***. Proceedings of the AAAI Conference on Artificial Intelligence , 34, s. 12993-13000.

Zhiqiang, W., & Jun, L. (2017). *A review of object detection based on convolutional neural network*. 2017 36th Chinese Control Conference (CCC) (s. 11104-11109). IEEE.

Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). *Object Detection in 20 Years: A Survey*. arXiv:1905.05055. adresinden alındı

