



**T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**DERİN ÖĞRENME TABANLI NESNE TESPİT ALGORİTMALARI
İLE HİSSE SENEDİ AL-SAT KARAR DESTEK SİSTEMİNİN
MODELLENMESİ**

GÜNAY TEMÜR

**DOKTORA TEZİ
ELEKTRİK-ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI**

**DANIŞMAN
DOÇ. DR. SERDAR BİROĞUL**

DÜZCE, 2021

T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME TABANLI NESNE TESPİT ALGORİTMALARI
İLE HİSSE SENEDİ AL-SAT KARAR DESTEK SİSTEMİNİN
MODELLENMESİ

Günay TEMÜR tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Serdar BİROĞUL
Düzce Üniversitesi

Eş Danışman

Doç. Dr. Utku KÖSE
Süleyman Demirel Üniversitesi

Jüri Üyeleri

Doç. Dr. Serdar BİROĞUL
Düzce Üniversitesi

Prof. Dr. Uğur GÜVENÇ
Düzce Üniversitesi

Doç. Dr. Serhat DUMAN
Bandırma Onyedli Eylül Üniversitesi

Prof. Dr. Tuncay YİĞİT
Süleyman Demirel Üniversitesi

Dr. Öğr. Üyesi Özkan ŞAHİN
Düzce Üniversitesi

Tez Savunma Tarihi: 02/02/2021

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

02 Şubat 2021

Günay TEMÜR



TEŞEKKÜR

Bir lisansüstü eğitimde en önemli unsur anlaşabildiğiniz, kolay iletişim kurabildiğiniz ve yol gösterici danışman hocanızdır. Doktora eğitimim süresince sağladığı destek ve katkılarından dolayı danışman hocam Doç.Dr. Serdar BİROĞUL'a teşekkür ediyorum. Konu arayışı içerisinde önerimize destek veren, beni cesaretlendiren eş danışmanım Doç.Dr. Utku KÖSE hocama, yapıcı eleştirileri ve yönlendirmeleri ile çalışmalarımıza olumlu katkılar sağlayan tik üyesi hocalarım Prof.Dr. Uğur GÜVENÇ ve Doç.Dr. Serhat DUMAN hocama teşekkür ederim.

Her telefon konuşmamızda “senin işin vardır, fazla meşgul etmeyeyim ama bir şey soracağım” diyen Babama ve hala okumanız bitmedi diyen Anneme sonsuz teşekkür ederim. Her bensiz gidişinde “tezine çalışıyor gelemedi” söyleminde, eşimi ve beni anlayışla karşılayan eşimin ailesine de teşekkür ederim.

Benim en büyük şansım eşim Dr. Ayşe SOY TEMÜR. Kendisinin de bir akademisyen olması, tez yazmak nasıl bir süreç, makale yazmak ne demek bilmesi işimi bir hayli kolaylaştırmıştır. Özellikle bu dönemlerde çocuklarımızın yükünü biraz daha fazla üstlenmesinden dolayı kendisine borçluyum. Gösterdiği sabırdan ötürü eşime çok teşekkür ederim.

Ayrıca çalışmada kullanılan hisse senedi geçmiş fiyat verilerini tarafımıza ücretsiz sağlayan BIST'e teşekkür ederim. Bu çalışmada yer alan bilgiler bir bilimsel çalışma niteliğinde olup kesinlikle al-sat-tut tavsiyesi içermez.

Son olarak kızlarımız Eylül İkra ve Elif Ada. Çalışma koltuğumu lunaparka çevirip “bizi biraz döndürsene” demeleri bitmedi. Tez yazma dönemi zorlu bir dönem. Özellikle bu dönemde kendilerinin benimle birliktelik zamanından çalmamaya gayret gösterdim. Onlar için şimdilik bir oyuncak kadar anlamı olmasa da Tezimi iki küçük kızıma ithaf ediyorum.

02 Şubat 2021

Günay TEMÜR

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ.....	viii
ÇİZELGE LİSTESİ.....	xi
KISALTMALAR.....	xii
ÖZET	xiv
ABSTRACT	xv
EXTENDED ABSTRACT.....	xvi
1. GİRİŞ.....	1
1.1. PROBLEM VE MOTİVASYON	7
1.2. TEZ KATKILARI	8
1.3. TEZ YAPISI	11
2. FİNANSAL ZAMAN SERİSİ VERİLERİ ANALİZİ	12
2.1. TEMEL ANALİZ.....	13
2.2. TEKNİK ANALİZ.....	13
2.2.1. Dow Teorisi	14
2.2.2. Eğilim Kavramı	15
2.3. FİNANSAL ZAMAN SERİSİ VERİLERİNİN İSTATİKSEL / MATEMATİKSEL ANALİZİ.....	17
2.4. FİNANSAL ZAMAN SERİSİ VERİLERİNİN GRAFİKSEL ANALİZİ.....	17
2.4.1. Grafik Türleri.....	17
2.4.1.1. Çizgi Grafik.....	17
2.4.1.2. Çubuk Grafik.....	18
2.4.1.3. Mum Grafik.....	19
2.4.1.4. Nokta ve Şekil Grafik	20
2.4.1.5. Kagi Grafik	21
2.5. FİNANSAL PİYASALARDA AL-SAT KARAR ANALİZLERİ.....	22
2.5.1. İSTATİKSEL/MATEMATİKSEL KARARLAR.....	22
2.5.2. GRAFİKLER ÜZERİNDE GERÇEKLEŞTİRİLEN KARARLAR	23
2.5.2.1. Formasyonlar.....	23
2.5.2.2. Teknik Göstergeler (Indicators).....	36
2.5.2.3. Fibonacci Düzeltmeleri.....	41
2.5.2.4. Elliott Dalga Teorisi	41
2.5.2.5. Pivot Noktası.....	42
3. DERİN ÖĞRENME ALGORİTMALARI	44
3.1. DERİN ÖĞRENME.....	44
3.1.1. DERİN ÖĞRENME YAKLAŞIMLARI.....	44
3.2. KONVÜLSİYON SINIR AĞLARI (CNN).....	45
3.2.1. CNN'nin İşleyişi	46
3.2.1.1. Konvolüsyon Katmanı.....	47
3.2.1.2. Doğrultulmuş Doğrusal Birimler Katmanı (ReLU-Rectified linear units layer).....	48
3.2.1.3. Havuzlama Katmanı.....	49
3.2.1.4. Tam Bağlantılı Katman.....	49
3.2.1.5. Kayıp Katmanı	50

3.2.1.6. Alt-Örnekleme Katmanı	51
3.2.1.7. Sınıflandırma Katmanı.....	51
3.2.2. Popüler Ağ Mimarileri	52
3.2.2.1. LeNet (1998)	53
3.2.2.2. AlexNet (2012).....	53
3.2.2.3. ZFNet / Clarifai (2013).....	54
3.2.2.4. Network in Network	55
3.2.2.5. VGGNET (2014)	55
3.2.2.6. GoogLeNet (2014)	56
3.2.2.7. ResNet (2015).....	57
3.2.2.8. FractalNet (2016).....	57
3.2.2.9. Yoğun Bağlantılı Ağ (DenseNet - Densely Connected Network) (2017).....	58
3.2.3. Diğer DNN (Deep Neural Networks) Modelleri	59
3.2.4. CNN Uygulamaları.....	59
3.2.4.1. Grafik Sorunu Çözmek için CNN'ler	59
3.2.4.2. Görüntü İşleme ve Bilgisayarla Görme	60
3.2.4.3. Konuşma İşleme.....	61
3.2.4.4. Tıbbi Görüntüleme.....	61
3.3. TRANSFER ÖĞRENME	61
3.4. BİLGİSAYAR GÖRÜŞÜ	62
4. NESNE TANIMA ALGORİTMALARI VE UYGULAMALARI..	64
.....	64
4.1. NESNE TANIMA ALGORİTMALARI	64
4.1.1. Faster R-CNN	66
4.1.2. YOLO (You Only Look Once)	68
4.1.3. SSD (Single Shot Detector).....	68
4.1.4. Benzer Algoritmalar	69
4.2. VERİ ETİKETLEME	72
4.2.1. Sınırlayıcı Kutular	72
4.2.2. Çokgen Bölümleme (Segmentasyon)	73
4.2.3. Anlamsal Bölümleme	73
4.2.4. 3D Küpoidler	74
4.2.5. Anahtar Nokta ve Yer İşareti.....	74
4.2.6. Çizgiler ve Spline'lar.....	75
4.2.7. Görüntü Ek Açıklama Formatları.....	75
4.2.7.1. Pascal VOC.....	75
4.2.7.2. YOLO	76
4.3. ALGORİTMALARIN METRİK PERFORMANS ÖLÇÜTLERİ.....	77
4.4. ALGORİTMALARIN EĞİTİMİ İÇİN DONANIM SEÇİMİ	81
4.4.1. Merkezi İşlem Birimi (MIB)	81
4.4.2. Grafik İşlem Birimi (GIB).....	82
4.5. ALGORİTMALARIN EĞİTİMİNDE KULLANILAN KÜTÜPHANELER	83
4.5.1. CUDA	83
4.5.2. cuDNN	84
4.5.3. Tensorflow	84
4.5.4. OpenCV.....	85
4.5.5. NumPy	85
4.5.6. Matplotlib	86
5. NESNE TESPİT ALGORİTMALARI İLE HİSSE SENEDİ AL-	87
SAT KARAR DESTEK MODELİNİN OLUŞTURULMASI	87
5.1. ÖNERİLEN MODEL	87

5.2. VERİ HAZIRLAMA	89
5.2.1. Veri Seti Oluřturma	89
5.2.2. Veri Etiketleme.....	91
5.2.2.1. Etiketleme Ařamaları	97
5.2.3. Veri Gruplandırma	98
6. BULGULAR VE TARTIřMA	100
6.1. DONANIM ALT YAPISI VE YAZILIM KURULUMU	100
6.2. MODEL İÇİN ALGORİTMALARIN BELİRLENMESİ VE EĐİTİMİ.....	100
6.2.1. Eđitimlerin Dosya Yapılanması	101
6.2.2. Eđitimlerin Metrik Deđerlendirme Sonuřları	111
7. SONUÇLAR VE ÖNERİLER.....	131
7.1. GELECEK ÇALIřMALAR	133
8. KAYNAKLAR	135
9. EKLER	153
9.1. EK 1: <i>DONUSTUR.PY</i> KODLAMA DOSYASI.....	153
9.2. EK 2: <i>PAYLAS.PY</i> KODLAMA DOSYASI	155
9.3. EK 3: PERFORMANS VERİ SETİ TEST TABLOSU	156
ÖZGEÇMİř	161

ŞEKİL LİSTESİ

Sayfa No

Şekil 1.1. Önerilen modele ait akış diyagramı.....	8
Şekil 2.1. Finansal analiz yöntemleri.....	12
Şekil 2.2. Dow Teorisi fiyat gelişimi.....	15
Şekil 2.3. Farklı eğilim görünüşleri.....	16
Şekil 2.4. Bir hisse senedine ait çizgi grafik görünüm.....	18
Şekil 2.5. Tek çubuk gösterim.....	18
Şekil 2.6. Bir hisse senedine ait çubuk grafik görünüm.....	19
Şekil 2.7. Mum grafik.....	20
Şekil 2.8. Bir hisse senedine ait mum grafik görünüm.....	20
Şekil 2.9. Bir hisse senedine ait nokta ve şekil grafik görünüm.....	21
Şekil 2.10. Bir hisse senedine ait kagi grafik görünüm.....	22
Şekil 2.11. Omu-Baş-Omuz-Tepe formasyon görünüm.....	24
Şekil 2.12. Omuz-Baş-Omuz-Dip formasyon görünüm.....	25
Şekil 2.13. Fincan Kulp formasyon görünüm.....	25
Şekil 2.14. İkili Tepe formasyon görünüm.....	26
Şekil 2.15. İkili Dip formasyon görünüm.....	26
Şekil 2.16. Üçlü Tepe formasyon görünüm.....	27
Şekil 2.17. Üçlü Dip formasyon görünüm.....	27
Şekil 2.18. Genişeyen Tepe formasyon görünüm.....	28
Şekil 2.19. Düşen Takoz formasyon görünüm.....	29
Şekil 2.20. Yükselen Takoz formasyon görünüm.....	29
Şekil 2.21. Yükselen Üçgen formasyon görünüm.....	30
Şekil 2.22. Alçalan Üçgen formasyon görünüm.....	31
Şekil 2.23. Simetrik Üçgen formasyon görünüm.....	31
Şekil 2.24. Bayrak formasyon görünüm.....	32
Şekil 2.25. Flama formasyon görünüm.....	32
Şekil 2.26. Ada Dönüş formasyon görünüm.....	33
Şekil 2.27. Kopuş Boşluğu formasyon görünüm.....	33
Şekil 2.28. Standart Boşluk formasyon görünüm.....	34
Şekil 2.29. Bitkinlik Boşluğu formasyon görünüm.....	34
Şekil 2.30. Mum formasyon görünüşleri.....	35
Şekil 2.31. MACD indikatör görünüm.....	37
Şekil 2.32. Parabolik SAR indikatör görünüm.....	38
Şekil 2.33. RSI indikatör görünüm.....	38
Şekil 2.34. CCI indikatör görünüm.....	39
Şekil 2.35. SO indikatör görünüm.....	40
Şekil 2.36. Bollinger Bands indikatör görünüm.....	41
Şekil 2.37. Fibonacci düzeltme oranları.....	41
Şekil 2.38. Elliott dalgaları.....	42
Şekil 2.39. Pivot noktaları.....	43
Şekil 3.1. AI taksonomisi.....	45
Şekil 3.2. Konvolüsyon sinir ağının (CNN) genel mimarisi.....	47
Şekil 3.3. Doğrultulmuş doğrusal birimler katmanının nasıl çalıştığına örnek.....	49
Şekil 3.4. Maksimum havuzlama örneği.....	49
Şekil 3.5. Konvolüsyon ve havuzlama sonrası öznitelik haritaları.....	52
Şekil 3.6. LeNet mimarisi.....	53

Şekil 3.7. AlexNet mimarisi.	54
Şekil 3.8. VGG ağ mimarisi.....	55
Şekil 3.9. Giriş katmanı.	56
Şekil 3.10. Boyut küçültmeli giriş katmanı.	56
Şekil 3.11. ResNet temel blok diyagramı.	57
Şekil 3.12. FractalNet modülü.	58
Şekil 3.13. DenseNet mimarisi.	59
Şekil 4.1. Nesne tespit algoritmalarının tarihsel gelişimi.	64
Şekil 4.2. Nesne tespiti performans gelişmeleri.	65
Şekil 4.3. Faster R-CNN mimari.	67
Şekil 4.4. YOLO mimari.....	68
Şekil 4.5. SSD mimari.	69
Şekil 4.6. Sınırlayıcı kutu örnek gösterim.	73
Şekil 4.7. Çokgen segmentasyon gösterim.	73
Şekil 4.8. Anlamsal bölümlenme örnek gösterim.	74
Şekil 4.9. 3D Küpoid örnek gösterim.	74
Şekil 4.10. Anahtar nokta ve yer işareti örnek gösterim.	75
Şekil 4.11. Çizgiler ve Spline'lar örnek gösterim.	75
Şekil 4.12. Pascal VOC ek açıklama dosya içeriği.	76
Şekil 4.13. YOLO ek açıklama dosya içeriği.	76
Şekil 4.14. Görüntü sınıflandırma ve yerelleştirme.	77
Şekil 4.15. IoU hesaplama örneği.	78
Şekil 4.16. Etiket ve tahmin ilişkisi (IoU).	78
Şekil 4.17. Nesne algılama örneği.	79
Şekil 4.18. Hassas geri çağırma eğrisi.	80
Şekil 4.19. Düzeltilmiş hassas geri çağırma eğrisi.	81
Şekil 4.20. CUDA işlem akış diyagramı.	84
Şekil 5.1. Çalışmaya ait akış diyagramı -1.	88
Şekil 5.2. Çalışmaya ait akış diyagramı-2.	89
Şekil 5.3. Bir hisse senedine ait bir yıllık 2D mum grafik.	90
Şekil 5.4. LabelIMG arayüz görünümü (a).	92
Şekil 5.5. LabelIMG arayüz görünümü (b).	92
Şekil 5.6. Etiket klasörü içeriği.	92
Şekil 5.7. Mum grafiği etiketleme stratejisi-1.	93
Şekil 5.8. Mum grafiği etiketleme stratejisi-2.	94
Şekil 5.9. Program üzerinde etiketleme gösterimi.	95
Şekil 5.10. Etiketleme işlemi örnek-1.	96
Şekil 5.11. Etiketleme işlemi örnek-2.	96
Şekil 5.12. Etiketleme işlemi örnek-3.	96
Şekil 5.13. Etiketleme öncesi yapay eğilim çalışması-1.	97
Şekil 5.14. Etiketleme öncesi yapay eğilim çalışması-2.	98
Şekil 5.15. Eğitim ve test veri seti gruplandırma.	99
Şekil 6.1. Eğitim ve test ek açıklamaları .xml to .csv komut istemi.	103
Şekil 6.2. Ek açıklama .csv dosya içeriği.	103
Şekil 6.3. Faster R-CNN ve SDD algoritmaları eğitim başlatma komut istemi.	103
Şekil 6.4. YOLO versiyonları eğitim başlatma komut istemi.	104
Şekil 6.5. Faster R-CNN ve SSD eğitim kayıp raporu.	104
Şekil 6.6. YOLO versiyonları eğitim kayıp raporu.	104
Şekil 6.7. Faster R-CNN ve SSD eğitim ağırlık değerleri dışa aktarım komut istemi.	105
Şekil 6.8. YOLO test işlemi komut istemi.	105

Şekil 6.9. Ekran görüntüsü alma kod bloğu.....	106
Şekil 6.10. YoloV3 VG-2 test çıktısı örnek görünüm-1.....	106
Şekil 6.11. Faster R-CNN VG-2 test çıktısı örnek görünüm-2.....	107
Şekil 6.12. YoloV3 VG-2 test çıktısı örnek görünüm-3.....	107
Şekil 6.13. Faster R-CNN VG-2 test çıktısı örnek görünüm-4.....	107
Şekil 6.14. YoloV3 VG-2 test çıktısı örnek görünüm-5.....	108
Şekil 6.15. Faster R-CNN test çıktısı örnek görünüm-6.....	108
Şekil 6.16. YoloV3 VG-2 test çıktısı örnek görünüm-7.....	108
Şekil 6.17. YOLO modelleri dosya yapılanması.....	109
Şekil 6.18. Faster R-CNN ve SSD modelleri dosya yapılanması.....	110
Şekil 6.19. Modellerin AP değerlendirmesi-1.....	111
Şekil 6.20. Modellerin AP değerlendirmesi-2.....	112
Şekil 6.21. Modellerin mAP değerlendirmesi.....	113
Şekil 6.22. Modellerin tespit sonuçları (FP-TP).....	114
Şekil 6.23. YoloV3 VG-2 tespit sonuçları değerlendirmesi.....	115
Şekil 6.24. YoloV3 VG-2 test çıktısı.....	118
Şekil 6.25. YoloV4 VG-2 test çıktısı.....	118
Şekil 6.26. Faster R-CNN VG-2 test çıktısı.....	118
Şekil 6.27. SSD VG-2 test çıktısı.....	119
Şekil 6.28. Algoritmaların test veri seti üzerindeki tahmin değerleri.....	120
Şekil 6.29. Faster R-CNN ve SSD algoritmalarının hatalı karar görüntüleri.....	120
Şekil 6.30. Yatırım için "Al" kararını belirleyen örnek grafik görüntüleri.....	121
Şekil 6.31. Yatırım için "Sat" kararını belirleyen örnek grafik görüntüleri.....	122
Şekil 6.32. Modellerin hisse bazlı kazanç dağılım grafiği.....	123
Şekil 6.33. Örnek Sepet-1 görünümü.....	125
Şekil 6.34. Al sinyali üreten senet görünüm örnek-1.....	126
Şekil 6.35. Al sinyali üreten senet görünüm örnek-2.....	126
Şekil 6.36. Al sinyali üreten senet görünüm örnek-3.....	126
Şekil 6.37. Sat sinyali üreten senet görünüm örnek-1.....	127
Şekil 6.38. Sat sinyali üreten senet görünüm örnek-2.....	127
Şekil 6.39. Sat sinyali üreten senet görünüm örnek-3.....	128
Şekil 6.40. Anapara kazanç grafiği.....	130

ÇİZELGE LİSTESİ

	<u>Sayfa No</u>
Çizelge 4.1. AP hesaplama tablosu.....	80
Çizelge 6.1. Tez uygulamasında kullanılan bilgisayar konfigürasyonu.	100
Çizelge 6.2. Tez uygulamasında kullanılan yazılım ve yazılım kütüphaneleri.	100
Çizelge 6.3. Etiket dosyaları içeriği.....	102
Çizelge 6.4. Konfigürasyon dosyaları içeriği.	102
Çizelge 6.5. Test veri seti model değerlendirmesi.....	116
Çizelge 6.6. Test veri seti tahmin değerleri.	117
Çizelge 6.7. Modellerin BIST-100 seçili hisse değerlendirmesi.	122
Çizelge 6.8. Sepet işlemleri özet tablosu.	124
Çizelge 6.9. Modellerin toplam sepet portföy değerlendirmesi.....	129



KISALTMALAR

AE	Otokodlayıcı
AI	Yapay Zekâ
ANN	Yapay Sinir Ağları
AÖM	Aşırı Öğrenme Makineleri
API	Uygulama Programlama Arayüzü
AR	Otoregresif
ARIMA	Otoregresif Bütünleşik Hareketli Ortalama
ARMA	Otoregresif Hareketli Ortalama
BB	Bollinger Bandı
BIST	Borsa İstanbul
BN	Parti Normalizasyonu
CCI	Emtia Kanal İndeksi
CGRN	Konvolüsyonel Geçitli Tekrarlayan Ağ
CNN	Evrişimli Sinir Ağları
CPU	Merkezi İşlem Birimi
CUDA	Hesaplama Birleşik Cihaz Mimarisi
cuDNN	Derin Sinir Ağı Kütüphanesi
DCNN	Derin Konvolüsyon Sinir Ağı
DL	Derin Öğrenme
DMLP	Derin Çok Katmanlı Algılama
EIC	Ekonomi Sanayi Şirketi
EMA	Üstel Hareketli Ortalama
ETF	Borsa Yatırım Fonları
FC	Tam Bağlı
FCN	Tam Konvolüsyon Ağı
FCNN	Tam Konvolüsyonel Sinir Ağları
FPS	Saniyedeki Kare Hızı
GA	Genetik Algoritma
GAP	Global Ortalama Havuzlamayı
GARCH	Genelleştirilmiş Otoregresif Koşullu Heteroskedastisite
GPGPU	Grafik İşleme Birimlerinde Genel Amaçlı Hesaplama
GPU	Grafik İşlem Birimi
GUI	Grafiksel Kullanıcı Arayüzü
IBB	Biyoteknoloji Endeksi
ILSVRC	ImageNet Büyük Ölçekli Görsel Tanıma Yarışması
IRCNN	Giriş Konvolüsyonlu Tekrarlayan Sinir Ağları
KOSPI	Kore Stok Fiyat Endeksi
LiDAR	Lazer Görüntüleme Tespiti ve Değişimi
LRN	Yerel Yanıt Normalleştirme
LSTM	Uzun Kısa Süreli Bellek
MA	Hareketli Ortalama
MACD	Hareketli Ortalama Yaklaşma/Uzaklaşma Göstergesi
mAP	Ortalama Ortalama Hassasiyet
MDNet	Çok Alanlı Evrişimli Sinir Ağı
MLP	Çok Katmanlı Algılama
MSE	Ortalama Karasel Hata
MS COCO	Microsoft Yaygın Nesnelere
MS-CNN	Çok Ölçekli CNN

NMS	Maksimal Olmayan Bastırma
NN	Sinir Ağları
OHLC	Açılış-Yüksek-Düşük-Kapanış-Grafiği
PASCAL VOC	PASCAL Görsel Nesne Sınıfları
PCA	Temel Bileşenler Analizi
PVANET	Hafif Ağırlıklı Derin Sinir Ağları
RBFNN	Radyal Temel Fonksiyon Sinir Ağı
RBM	Kısıtlı Boltzmann Makinesi
RCNN	Bölge Bazlı CNN
RCNN	Tekrarlayan Konvolüsyon Sinir Ağı
ReLU	Doğrultulmuş Doğrusal Birimler Katmanı
RNN	Tekrarlayan Sinir Ağları
RoI	İlgi Bölgesi
RoIPool	İlgi Bölgesi Havuzlama
RPN	Bölge Önerisi Ağı
RSI	Göreceli Güç Endeksi
S&P	Standard & Poor's Endeksi
SAE	Yığılmış Otomatik Kodlayıcılar
SAR	Bitiş ve Dönüş
SMA	Basit Hareketli Ortalamadır
SO	Stokastik Osilatör
SSD	Tek Seferde Tespit
SSE	Enerji Şirketi
STL	Standart Üçgen Dili
TAR	Eşiksel Otoregresif
VG	Veri Grubu
VGG	Görsel Geometri Grubu
VIP CNN	Görsel Faz Gülümlü CNN
WT	Dalgacık Dönüşümleri
YOLO	Sadece Bir Kez Bak
YSA	Yapay Sinir Ağları

ÖZET

DERİN ÖĞRENME TABANLI NESNE TESPİT ALGORİTMALARI İLE HİSSE SENEDİ AL-SAT KARAR DESTEK SİSTEMİNİN MODELLENMESİ

Günay TEMÜR

Düzce Üniversitesi

Fen Bilimleri Enstitüsü, Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

Danışman: Doç. Dr. Serdar BİROĞUL

Şubat 2021, 155 sayfa

Finansal alanda yatırım gerçekleştiren tüm yatırımcıların ana hedefi, bir yatırım enstrümanını düşük fiyattan alıp yüksek fiyattan satış yaparak kar elde etmektir. Bu bakış açısı çerçevesinde çalışma kapsamında, hisse senedi piyasası, parite analizi, endeks analizi ve diğer borsaların hisse analizinde, kısacası mum grafiği oluşturulan tüm yatırım enstrümanları için kullanılabilir etkili bir “Al-Sat” karar destek modeli tasarlanmıştır. Nesne tespit tabanlı çalışan bu model tasarımı hem finansal hem de bilimsel olan iki yönlü farklı bakış açısıyla tasarlanan yenilikçi bir model yaklaşımıdır. Çalışmada BIST hisse senetlerine ait 2D mum grafik verileri kullanılmıştır. Grafikler iki ayrı veri seti şeklinde etiketlenmişlerdir. İlk olarak 550 adet 2D mum grafik üzerinde 10000 adet veri etiketleme gerçekleştirilmiş ve YoloV3 Veri Grubu-1 (VG-1) ile eğitilmiştir. Ardından veri seti 20000 adete çıkarılmıştır. Hazırlanan bu 20000 adet etiketli veri seti üzerinden rastgele 10000 etiketli veri seçilmiştir. Yeni oluşturulmuş 10000 adet etiketli veri seti VG-2 olarak isimlendirilmiş ve YoloV3, YoloV4, Faster R-CNN, SDD algoritmaları ile eğitilmişlerdir. Seçilen bu dört algoritma için gerçekleştirilen eğitimler sonucu elde edilen başarımlar değerlendirilmiştir. Bu değerlendirme için üç farklı senaryo düzenlenmiş olup, tüm bu senaryolardan iyileştirilmiş veri seti ile eğitilmiş YoloV3 VG-2 algoritması en başarılı sonuçlara sahiptir. Gerçekleştirilen karşılaştırmalı test senaryoları sonucunda YoloV3 VG-2 modeli,%98 nesne tespiti başarısı sağlamıştır. Bununla birlikte tespit edilen nesne sınıfı için %100 doğru tahmin başarısı ve altı aylık periyotta %89,84 kazanç getirisi sağladığı gözlemlenmiştir. Gözlemlenen bu kazanç başarısında herhangi bir ek parametre kullanılmamıştır. Sonuç olarak final model olarak belirlenen YoloV3 VG-2 mum grafik görünümü oluşturulabilen tüm yatırım enstrümanları için bir karar destek modeli olarak uygulanabilir.

Anahtar sözcükler: Derin öğrenme, Nesne tanıma, Örüntü tespit, Finans, Eğilim tespit.

ABSTRACT

MODELING OF TRADING DECISION SUPPORT SYSTEM WITH DEEP LEARNING BASED OBJECT RECOGNITION ALGORITHMS

Günay TEMÜR

Duzce University

Graduate School of Natural and Applied Sciences, Department of Electrical-Electronic
and Computer Engineering

Doctoral Thesis

Supervisor: Assoc. Prof. Dr. Serdar BİROĞUL

February 2021, 155 pages

The fundamental purpose of every investor making investments in financial fields, is to make profit by buying an investment instrument at a low price and selling the same at a higher price. In this study, within the framework of the aforementioned standpoint, an effective “Trading” decision support model was designed, which can be used for stock market analyses, parity analyses, index analyses, and for the stock analyses of other stock exchanges, briefly for all investment instruments for which candlestick charts are created. This object detection-based model design is an innovative model approach designed with a two-way different perspective, both financial and scientific. The study incorporated the use of 2D candlestick charts of the BIST stocks. The charts were labelled in two separate data sets. Initially, 10,000 pieces of data were labelled on 550 2D candlestick charts, which were trained with YoloV3 Data Group-1 (DG-1). Subsequently, the data set was increased to 20,000 pieces. Out of this set of 20,000 labelled data prepared, 10,000 labelled data were picked randomly. The newly-created set of 10,000 labelled data was named VG-2, which was trained with the YoloV3, YoloV4, Faster R-CNN, SDD algorithms. An assessment was made regarding the performance results obtained following the trainings implemented for these four chosen algorithms. For the aforementioned assessment, three different scenarios were created, and out of all these scenarios, the YoloV3 DG-2 algorithm, which was trained with an improved data set, was observed to be most successful one. As a result of the comparative test scenarios, the YoloV3 DG-2 model achieved a pattern recognition success of 98%. On the other hand, it was also observed to have achieved a prediction success of 100%, while bringing in a return by 89.94%, regarding the object class detected. In addition, no additional parameters were used in this observed gain success. Consequently, the YoloV3 DG-2, determined as the final model, could be implemented as a decision support model for all investment instruments for which a candlestick chart can be created.

Keywords: Deep learning, Object recognition, Pattern detection, Finance, Trend decision.

EXTENDED ABSTRACT

MODELING OF TRADING DECISION SUPPORT SYSTEM WITH DEEP LEARNING BASED OBJECT RECOGNITION ALGORITHMS

Günay TEMÜR

Duzce University

Graduate School of Natural and Applied Sciences, Department of Electrical-Electronic
and Computer Engineering

Doctoral Thesis

Supervisor: Assoc. Prof. Dr. Serdar BİROĞUL

February 2021, 155 pages

1. INTRODUCTION

The model approach recommended in this study, is a decision support model that is generally based on an object recognition and classification algorithm. As it is beyond numerical processes, it is quite different than the deep learning-based prediction models being exemplified, as is seen in the literature review. The objective of the model is to ensure that the points at which trading transactions would be made for investment instruments through 2D candlestick charts, are determined as objects. Labels were created in a systematic manner for the model created based on the object recognition and classification algorithms, and a series of trainings was implemented with those labels. Consequently, it was ensured that the model recognized the points at which trading transactions would be made, and that it could make a decision by learning this transaction only through the images for once, like an autonomous vehicle could recognize road lines, traffic signs, other different vehicles, and pedestrians.

2. MATERIAL AND METHODS

A data set is needed for the training and test processes of the recommended models. The 2D candlestick images of the price information of the current stocks listed on BIST were used for this data set that was needed. Following the addition of 709 pieces of 2D candlestick chart images to the 550 pieces of 2D candlestick chart images [1], which were created for the first study to have been conducted, the data set reached a total of 1,259 pieces of chart images.

Labeling process is the initial step of computer vision studies. This study employed the use of the “LabelIMG” image labeling tool that provided the opportunity to function in

the bounding boxes feature, through which the processes were conducted on 2D images. Separate remark files were created for each image while executing the labeling process. The labeling processes were entirely carried out in a manual manner, and within the scope of the “Dow Theory”.

Obtained from 550 pieces of images that were initially created, 10,000 pieces of labeled data were grouped as 20% test data, and 80% training data. Subsequently, this data was trained, the necessary tests were made and the results were recorded (YoloV3 DG-1) [1]. Then, following the addition of 709 pieces of data sets within the scope of this study, a total of 1,259 data sets were created, on which 20,000 pieces of data were labeled. Out of this set of 20,000 pieces of labeled data that was created, 10,000 labeled data were picked randomly. In a similar manner to the previous groups, the newly-created set of 10,000 pieces of labeled data were grouped as 20% test data, and 80% training data.

The newly-created set of 10,000 labeled data was named DG-2, which was trained with the YoloV3, YoloV4, Faster R-CNN, SDD algorithms. Following the training of the algorithms, new weight values were created for the objects, and the models were separately tested using these weight values.

3. RESULTS AND DISCUSSIONS

Following the training phase of the models, a series of tests was implemented in order to assess the object recognition performances of the weight values obtained. Initially, training outputs were analyzed in terms of Precision and Recall values. This data set assessment process is a comparison process that is conducted regarding the prediction accuracy by testing the test data set with the models that have new weight values. As per the detection results, the YoloV3 DG-2 object detection algorithm made the highest number of and most successful detections. At another phase of assessment of the study, 5 different stocks listed on BIST-100 were randomly selected. The chart images of the candlestick image of the aforementioned stocks over the course of the past six months, were recorded. At this phase, the test processes were implemented for the stock chart images selected between the months of January-July, using four different models (YoloV3 DG-2, YoloV4 DG-2, SSD DG-2, Faster R-CNN DG-2), as well as YoloV3 DG-1. Ultimately, the model testing processes were carried out using the basket model, which is known in the literature as one of the most lucrative methods of the investment

system. Consequently, the YoloV3 VG-2 model structured within the scope of this study, was determined as the final model, as it yielded successful results in all tests.

4. CONCLUSION AND OUTLOOK

Having been designed as an innovative financial decision support model within the scope of this study, this model was trained with four different deep learning-based object detection algorithms. The models created were tested using three different scenario methods. At the first testing phase, the models were subjected to a test set performance assessment within the data set created. This assessment process was carried out in the form of comparison of the number of decision detections produced by the models using the number of labels, as well as the determination of the accuracy of the decision detections. The YoloV3 DG-2 and YoloV4 DG-2 algorithms achieved a rate of success of 100% in these test assessments. Of the stocks listed on BIST-100, five different stocks were selected randomly for the second assessment scenario. Tests were carried out regarding the “Trading”-oriented object detection of the models on the candlestick charts, which were selected within a period of six months. Similarly, the YoloV3 DG-2 model, which was created within the scope of the thesis, yielded the highest amount of profit, achieving the most successful result. Finally, a comparison was made between YoloV3 DG-2, which was the most successful model, and YoloV3 DG-1. This final comparison process can actually be expressed as the assessment of the results of the data improvement process. Considering the test results obtained for the six-month period, it is possible to say that YoloV3 DG-2 certainly yielded more profitable results, compared to YoloV3 DG-1, which means that the data improvement affected the profits.

1. GİRİŞ

Günümüzde teknolojik alanlarda gerçekleşen ileri düzey gelişmeler ile finansal alanda özellikle borsa ve piyasa işlemlerinde yapay zekâ uygulamalarının kullanımı oldukça fazla yer tutmaya başlamıştır. Bu gelişmeler, hem finansal enstrümanların (hisse senedi, ETF (Exchange Traded Funds), döviz, altın, petrol vb.) sayısının artmasına hem de yatırımcı sayısının artmasına sebep olmuştur. Özellikle artan finansal enstrümanlar ile birlikte ana hedefi kazanç elde etmek olan yatırımcılar, çeşitli yatırım olanaklarına sahip olmuşlardır. Tasarrufların değerlendirilmesi aşamasında yatırımcılar, çeşitli finansal enstrümanlardan hangisi ya da hangilerine yatırım yapacaklarının kararını birtakım analiz yöntemlerine göre belirlemektedirler. Bu sebeple bir finansal enstrümanı düşük fiyattan satın alıp, yüksek fiyattan satarak kazanç elde etme mantığı ile hareket eden yatırımcılar açısından finansal piyasalarda fiyat değişimlerinin analiz edilmesi önemli bir araştırma konusudur [2].

Günümüzde finansal piyasa yatırımcılarının alım-satım kararlarını etkileyen zaman serisi verilerinin analizinde finansal analiz göstergeleri olarak bilinen başlıca iki analiz yöntemi söz konusudur. Bunlar temel analiz ve teknik analizdir.

Temel analiz, halka açıklanan bilgilerden yararlanılarak şirketlerin gerçek değerlerinin belirlenmesi çalışmasıdır. Teknik analiz ise, piyasalarda büyük önem taşıyan fiyat değişimlerinin yanı sıra herhangi bir finansal varlığın portföye ne zaman dâhil edilip ne zaman portföyden çıkarılacağına kararının verilmesinde kullanılmaktadır. Finansal verilerin analizinde, yine temel ve teknik analiz yaklaşımlarının temel aldığı yapay sinir ağları, destek vektör makineleri, makine öğrenme algoritmaları ve hibrit çözümler ile oluşturulan modeller de kullanılmaktadır.

Bu çözümlerin yanı sıra son zamanlarda derin öğrenme algoritmaları, finansal verilerin analizinde kullanılmaya başlanmış ve oldukça başarılı sonuçlar elde edilmiştir. Çok katmanlı derin yapay sinir ağları [3], [4], tekrarlamalı yapay sinir ağları, uzun kısa vadeli hafıza ağları [5], [6], sınırlı boltzmann makinesi [7], derin düşünce ağları başlıca kullanılan derin öğrenme algoritmalarıdır. Derin öğrenme (DL - Deep Learning), özellikle görüntü, metin ve ses verileri ile ilgili sınıflandırma çözümleri üreten yapay

zekâ algoritmalarıdır. Son yıllarda, özellikle derin öğrenme algoritmalarına dayanan tahmin ve sınıflandırma modellerinin görüntü, video ve ses işleme alanlarında çok iyi performans verdiği gibi finansal alanda da kullanımlarının arttığı görülmektedir. Bu sebeple hedefi kazanmak olan yatırımcılar için, piyasalarda karar verebilen, otonom ve akıllı uzman sistemlerin geliştirilmesi ile ilgili çalışmalar oldukça önemli hale gelmiştir.

Bu tez çalışmasında, önerilen model yaklaşımı genel olarak bir nesne tanıma ve sınıflandırma algoritması tabanlıdır. Bu model, sayısal işlemlerden uzak olması sebebi ile literatür taramasında da örnekleri verilen derin öğrenme metotlu tahmin modellerinden farklı olarak, daha önceden çalışılmamış bir modeldir. Modelin amacı, 2D mum grafikleri üzerinden yatırım enstrümanı için alınacak ve satılacak noktaların birer nesne olarak tespit edilmesini sağlamaktır. Nesne tespit ve sınıflandırma algoritmalarından yola çıkarak oluşturulan model için sistematik bir şekilde etiketler oluşturulmuş ve bu etiketler ile bir dizi eğitimler gerçekleştirilmiştir. Sonuçta, tıpkı bir otonom aracın yol çizgilerini, trafik işaretlerini, diğer farklı araçları ve yayaları tanıması gibi grafik üzerinde alınacak ve satılacak noktaların model tarafından tanınması ve bu işlemi sadece görseller üzerinden öğrenip görseller üzerinden karar vermesi sağlanmıştır. Oluşturulan bu model, diğer tahmin yöntemlerinden farklı olarak bir zaman serisinin mum grafikleri görünümü üzerinde performanslı bir şekilde çalışmaktadır. Ayrıca yine etiketleme işlemlerinin de görseller üzerinden gerçekleştirilmesi sebebi ile çalışma tamamen nesnel bir çalışmadır. Çalışmanın matematiksel yöntemlerden, sayılardan uzak olmasındaki amaç ise, yapay zekâ sisteminin temelinde yatan bakış açısıdır. Finans sektörü her ne kadar sıralı sayılar kümesi olan zaman serilerinden oluşsa da bu serileri temsil eden grafiklerle anlamlandırılmaktadır. Bu grafiklere bakan insan gözü sayılardan elde edemeyeceği analizleri bir bakışta beyinde canlandırabilmektedir. İnsan zekâsı, sayılardan ziyade gördüğünde yorumlayabildiği grafiksel gösterimler sonucu daha iyi kararlar verebilmektedir.

Diğer bir husus olan grafikler içerisinde mum grafikler üzerinde çalışılmasının sebebi ise, bu grafik türünün diğer grafik türlerine göre bir nesne anlamına daha fazla sahip olmasından kaynaklanmaktadır. Günümüz finansal piyasalarda zaman serilerini temsil edebilecek çok fazla grafik mevcut olmakla birlikte en çok kullanılanları mum, çubuk, çizgi grafikler olarak ifade edilebilir. Grafikler ile analiz yöntemleri bölümünde bu konuya geniş yer verilmiştir. Bu bağlamda mum grafik türünün diğer grafik türlerinden

farklı olarak bir gövde bölgesinin olması iniş ve çıkış hareketlerini farklı renkler ile temsil edebilmesi yönünden bilgisayar görüşünde daha verimli sonuçlar elde edileceği düşünülmüştür.

Bu yaklaşımdan yola çıkılarak sayısal ifadelerden tamamen uzak durup, sadece finansal zaman serilerinin oluşturduğu mum grafikler üzerinden karar veren derin öğrenme tabanlı nesne tespit modeli tasarlanmıştır.

DL teknikleri, bilgisayarla görme [8], [9], otomatik ses tanıma [10], [11], doğal dil işleme [12], [13], [14], el yazısı tanıma [11], ve biyobilişim [15] gibi birçok alanda uygulanmış ve başarılı sonuçlar elde edilmiştir. Ayrıca, DL'nin finansal zaman serisi verilerinde kullanımı son yıllarda daha popüler hale gelmiştir. Bu bağlamda derin öğrenme modeline odaklanan bir dizi makale incelenmiştir.

[16]'da finansal riskin gazete haberlerine dayalı tahmini için bir DL yaklaşımı önerilmiştir. Bankacılık ile ilgili raporların seviyesini ölçmek için kamuya açık metinsel veriler kullanılmış ve bir sınıflandırıcı verilen bir cümleyi sıkıntı veya huzur olarak sınıflandıracak şekilde eğitilmiştir.

[17]'de haber başlıklarına dayanarak Alman hisse senedi getirilerini tahmin etmek için metin tabanlı bir sınıflandırıcı eğitmiş ve bir işlem stratejisi geliştirmeksizin bir sonraki işlem gününün üç sınıflı tahmininde %56 oranında isabetlilik bildirilmiştir.

[18]'de günlük S&P 500 hareketlerini tahmin etmek için manşetlerden çıkarılan yapılandırılmış bilgileri kullanarak benzer bir konu incelenmiştir.

[19]' da portföy optimizasyonu, yatırım aktiflerinin optimal dinamik seçimidir ve Heaton tarafından biyoteknoloji endeksi IBB'den daha iyi performans gösterecek bir portföy öngörmeye çalışmak suretiyle araştırılmıştır.

[20]'de Google hisse senedi yurt içi eğilimlerini piyasa dalgalanmasının bir göstergesi olarak kullanılmış ve S&P500 dalgalanmasını modellemek için Uzun Kısa Süreli Bellek (LSTM - Long-Short-Term Memory) sinir ağı uygulanmıştır. Böylece, çok parametrelili makro ekonomik durumun yanı sıra kamuoyunda hâkim olan havayı ve lineer Ridge/Lasso ve GARCH gibi, ortalama mutlak hata yüzdesine göre daha iyi performans gösteren ölçütleri yansıtılmıştır.

[21]'de finansal piyasa eğilimleri tahmini için, çoklu finansal araçları kullanarak (43 emtia ve foreks aktifini içeren) bir tahmin modeli eğiten Dixon tarafından, gelecekteki

eğilimi pozitif, düz veya negatif olarak sınıflandırmayı amaçlayan derin bir sinir ağı önerilmiştir.

Yayın istatistiklerine göre, finansal zaman serileri tahmini için çoğu araştırmacının tercihi LSTM olmuştur. LSTM ve varyasyonları, zamanla geciken verileri geri bildirim gömülü temsiller ile birlikte kullanmış, bunun sonucunda zaman serileri tahmin uygulamalarında daha yüksek performans elde etmişlerdir. Finansal verilerin çoğu, şu ya da bu şekilde zamana bağlı bileşenler içerdiğinden, LSTM finansal zaman serileri tahmin sorunları için doğal seçim olmuştur. Bu arada, zaman serisi tahmin sorunları için başka modeller de kullanılmıştır. Bunlar arasında DMLP (Derin Çok Katmanlı Algılama - Deep Multi Layer Perception), yüzeysel kuzeni MLP (Çok Katmanlı Algılama - Multi Layer Perception)'nin piyasadaki egemenliği, makine öğrenmesi camiası içerisindeki yaygın kabulü ve köklü tarihi sayesinde en fazla ilgi çeken model olmuştur. Bununla birlikte, DMLP ve RNN (Recurrent Neural Network – Tekrarlayan Sinir Ağları) tabanlı modellerin finansal zaman serisi tahmin problemlerinde kullanım biçimleri arasında temel bir fark vardır. DMLP hem regresyon hem de sınıflandırma problemleri için uygundur. DMLP çalışmalarının çoğunda, ilgili sorun özellikle eğilim tahmin modellerinde sınıflandırma olarak ele alınırken, RNN tabanlı modeller zaman serisinin bir sonraki değerini doğrudan tahmin etmiştir. Her iki yaklaşım da temel ölçütleri karşılamakta başarılı olmuştur; bu nedenle bir modelin diğerinden üstün olduğunu iddia etmek mümkün değildir. Bununla birlikte, genel temel kural gereği, araştırmacılar zaman serisi regresyonu için RNN tabanlı modelleri, eğilim sınıflandırması (ya da alım satım noktası belirlemek) için ise DMLP'yi tercih etmektedirler [22].

Diğer taraftan zaman serisi analizlerinden geçmişteki verileri kullanarak gelecekteki değerler hakkında tahminler yapmak ve politikalar oluşturmak adına makine öğrenmesi ve derin öğrenme yöntemleri ile gerçekleştirilen birçok yöntem literatürde bulmak mümkündür. Gerçekleştirilen her yöntemin kendine göre çeşitli avantaj ve dezavantajları mevcuttur. Bu bağlamda;

[23]'te kural tabanlı sinir ağlarını kullanarak kısa vadeli ve uzun vadeli eğilimlerin tahminine göre alım ve satım sinyallerini tahmin edebilen akıllı bir hisse senedi alım satım karar sistemi önerilmiştir.

[24] ve [25]'te aşırı öğrenme makineleri (AÖM) ve yapay sinir ağları (YSA) yöntemleri

ile Goodyear, Amazon.com, Wal-Mart isimli şirketler ve S&P500 endeksine ilişkin geçmiş fiyat ve hacim bilgilerine ait 12 teknik gösterge girdi değişkeni olarak dikkate alınmıştır. Çıktı değişkeni olarak ise, bir sonraki güne ait kapanış fiyatları kullanılmıştır. Elde edilen bulgulara göre üç adet hisse senedinde AÖM yöntemi YSA yönteminden daha yüksek isabet oranları sergilerken, Endekste ise YSA modelinin daha iyi isabet oranı performansı sergilediği belirlenmiştir. Ayrıca AÖM ile eğitilen modellere göre hareket eden yatırımcının pozitif getiri sağlayabileceği sonucuna da ulaşılmıştır.

[26]'da hisse senedi tahmini için derin öğrenme yöntemleri kullanılmış ve yöntemin performansı NASDAQ'tan alınan Google hisse senedi fiyatı multimedya verisinde (grafik) değerlendirilmiştir. Principal Component Analysis (2D) PCA + Deep Neural Network (DNN) methodu ile en gelişmiş yöntem 2-Directional 2-Dimensional (2D) 2 PCA + Radial Basis Function Neural Network (RBFNN) yöntemi karşılaştırılmış, önerilen yöntemin mevcut bir yöntem olan RBFNN'den daha iyi performans gösterdiği sonucuna ulaşılmıştır.

[27]'de hisse senedi fiyat tahmini için dalgacık dönüşümleri (wavelet transforms-WT), yığılmış otomatik kodlayıcılar (stacked autoencoders-SAE) ve LSTM'in birleştirildiği yeni bir derin öğrenme çerçevesi sunulmuştur. Önerilen modelin hem tahmin doğruluğu hem de karlılık performansı açısından benzer modellerden daha iyi performans gösterdiği sonucuna ulaşılmıştır.

[28]'de hisse senedi hareketlerini önceden tahmin edebilmek için hem çevrimiçi finans haberlerine hem de geçmiş hisse senedi fiyatları verilerine dayalı yeni bir tahmin modeli geliştirilmiştir. Elde edilen deneysel sonuçlara göre modelin, S&P 500 endeks tahmininde yaklaşık %60, bireysel hisse senedi fiyat tahmininde ise %65'in üzerinde bir tahmin doğruluğuna ulaştığı belirtilmiştir.

[29]'da mevcut finansal verileri kullanarak LSTM ve genetik algoritma (GA) içeren karma bir yaklaşım ile yeni bir borsa tahmin modeli geliştirilmesi amaçlanmıştır. Çalışmada Kore Stok Fiyatı Endeksi (KOSPI) verisi kullanılarak hem önerilen modeller tek başına hem de iki modelin birleşiminden oluşturulan hibrit yaklaşım ile tahmin yapılmıştır. Elde edilen sonuçlara göre hibrit modelin diğer modellerden daha iyi performans gösterdiği ifade edilmiştir.

[30]'da hisse senedi fiyat tahmini için Attention-Based Multi-Input LSTM yöntemi kullanılmıştır. Çin borsasından seçilen hisse senetlerinin açılış fiyatları ile 3 farklı faktör

üzerinde denemelerin yapıldığı çalışmada; önerilen yaklaşımın LSTM'in farklı türevleri ile karşılaştırılması sonucunda en düşük ortalama karesel hata (mean squared error-MSE) değerine sahip olduğu ve daha iyi performans gösterdiği sonucuna ulaşılmıştır.

[31]'de "Deep Active Learning for Object Detection" başlıklı çalışmada eğitim görüntülerinin sadece bir kısmını kullanarak nesne tespitinde son teknoloji sonuçlar üreten aktif öğrenme yaklaşımlarını önerilmiştir. Elde edilen sonuçta ise, önerilen yöntemin klasik belirsizlik temelli aktif öğrenme algoritmalarından daha iyi bir performans sergilediği ifade edilmiştir.

[32]'de temel hisse senedine ilişkin herhangi bir ek zaman serisi sunmaksızın doğrudan 2-D hisse senedi çubuk grafik görüntüleri üzerinden "Al", "Sat" ve "Tut" senaryolarını belirlemek için evrişimli sinir ağını kullanan, stok tahmini için geleneksel olmayan bir yaklaşım önerilmiştir. Elde edilen sonuçlara göre, önerilen modelin genel olarak tutarlı sonuçlar üretebildiği ve piyasa koşullarına bağlı olarak başarıyı sağladığı ifade edilmiştir.

[33]'te Arima, LSTM ve Hybrid modeller ile ayrı ayrı gerçekleştirilen tahminleme çalışmalarında geleceğe yönelik olarak Türkiye'deki konutların satış miktarları tahmin edilmiştir. Farklı bir zaman serisi kullanarak derin öğrenme yaklaşımı ile gerçekleştirilen çalışmada başarılı sonuçlar elde edilerek literatüre katkı sağlanmıştır.

[34]'te BIST'te listelenen hisselerin günlük ve saatlik hareket yönleri yapay öğrenme ve derin öğrenme yöntemleriyle tahmin edilmiştir. Günlük yön tahmininde yapay öğrenme modelleri kullanılırken, saatlik yön tahmininde yapay öğrenme modellerinin yanında derin öğrenme modellerinden de yararlanılmıştır.

[35]'te farklı eşik değerleri kullanılarak gün içi dönemlerde BIST 100 endeksinin yönünü öngörmeye Derin Sinir Ağları, Destek Vektör Makinesi, Rastgele Orman ve Lojistik Regresyon ile karşılaştırılmıştır. Elde edilen sonuçlara göre modelin karışıklık matrisine göre hassasiyeti yüzde 60'a yakın bir değerde olup modeli kullanmanın sonucunun karlı olacağı belirtilmiştir.

Finansal tahminlerin doğrusal olmayan yani matematiksel olarak ifade edilemeyen bir sistem olmasından dolayı, fiyat tahminlerinde bulunmak ve sonucunda kazanç elde etmek hem finansçılar için hem de yapay zekâ alanında çalışan akademisyenler için her zaman ilgi çekici bir araştırma konusu olmuştur.

Benzer çalışmalarda da görüleceği üzere bugüne kadar özellikle finansal alanda zaman

serisi tahminlemeleri üzerine sayısız çalışmayı literatürde bulmak mümkündür. Ancak bireysel yatırımcılar açısından bakıldığında, özellikle geliştirilen matematik tabanlı formüsel yöntemlerin kullanıldığı yöntemlerdir. Genellikle de çalışma yapıları geçmişteki birçok verinin incelenmesi sonucu kısıtlı geleceğe dair tahminlerde bulunulmasıdır. Bu çalışmada kapsamında ise yenilikçi bir finansal karar destek modeli tasarlanmıştır. Tasarlanan model bir nesne tespit modelidir ve bir kez eğitilmek suretiyle gün sonu kapanışlı 2D mum grafik görüntüsüne sahip tüm yatırım enstrümanları için ilerleyen tüm zamanlarda kullanılabilir.

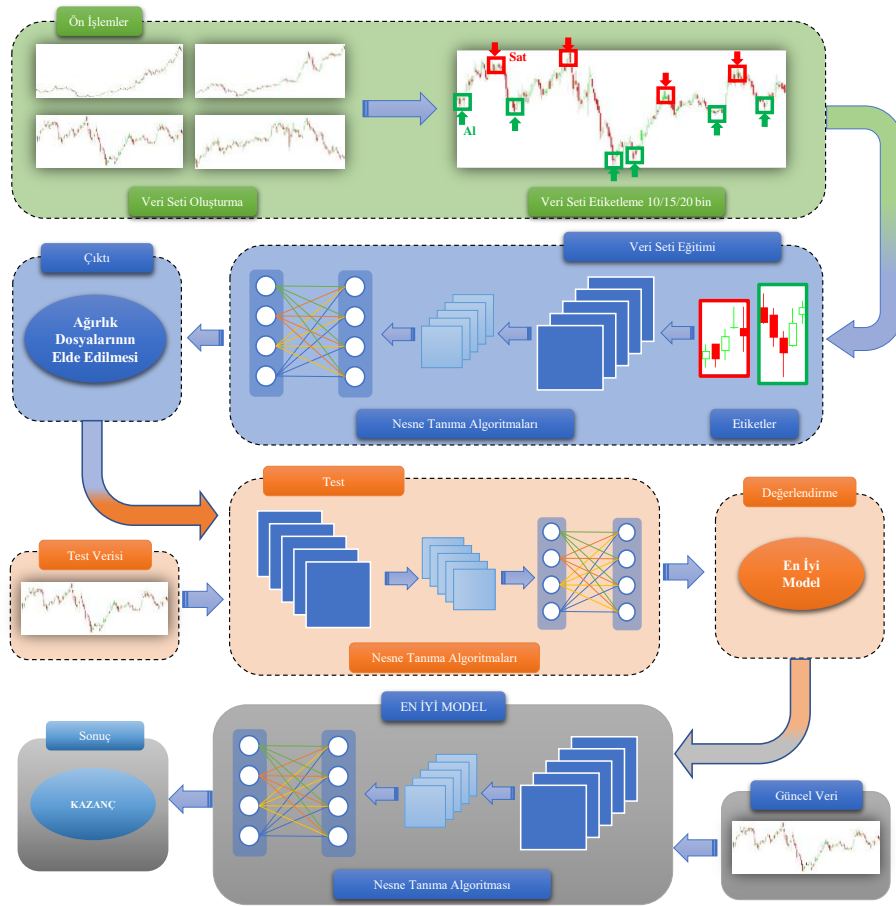
1.1. PROBLEM VE MOTİVASYON

Bugüne kadar yukarıda da bahsettiğimiz özellikle finansal alanda zaman serisi tahminlemeleri üzerine yapılmış çok sayıda çalışmayı literatürde bulmak mümkündür ve bu konuda oldukça başarılı sonuçlar elde edilmiştir. Ancak bireysel yatırımcılar açısından bakıldığında, özellikle geliştirilen matematik tabanlı formüsel yöntemlerin çoğunun, basit olarak aktif piyasalarda kullanılması pek mümkün değildir. Ayrıca bu çalışmaların genel yaklaşımları, sayısal zaman serisi verilerinin ele alınarak bir regresyon problemi gibi değerlendirilmesi sonucu tahminlerde bulunmak veya yatırım aracına ait belirli bir geçmiş davranışına göre gelecek durum analizinin yapıldığı yapay sinir ağları çalışmalarıdır. Genel olarak bir sonraki değeri tahmin etmek adına belirli bir geçmişini inceleme yöntemleridir. Yani sistemin sürekli olarak yeniden eğitilmesi şeklinde söylem oldukça doğru olacaktır. Bu yöntemlerin daha çok yatırım şirketlerinin arka planlarında yatırımcılara destek olarak çalıştırıldığı ve sonuçlarını sadece belirli uzmanların anladığı şekilde söylem daha doğru olacaktır. Diğer taraftan, finansal piyasalar için üretilmiş ve grafikler üzerinde gerçekleştirilen pek çok teknik analiz yöntemi, formasyon, indikatör sistemleri mevcuttur. Bu sistemler genellikle istatistiksel/matematiksel hesaplamalar ve algoritmalar ile çalışmakta olup, sonuç değerlendirmelerini ve kararlarını sayısal veriler üzerinden gerçekleştirmektedirler. Sonrasında herkesin daha kolay anlayabileceği bir dil olan grafiksel görünümlere aktarılmaktadır. Aslında buradaki en önemli husus; tüm yatırım analiz araçlarının avantajlarından faydalanarak yatırımcının en karlı yatırımı gerçekleştirmesini sağlamaktır. Çünkü her bir aracın diğerine göre farklı avantajları mevcuttur. Bu noktada yatırım araçları birbirleri ile bir yarış içerisinde olmayıp, birlikte kullanımları sayesinde en iyi getiriye yön verecek destek modelleri olarak düşünülmelidir. Bu analiz araçları,

çalışmada da bahsedildiği üzere dünyaca kabul edilmiş yöntemlerdir.

1.2. TEZ KATKILARI

Genel akış diyagramı Şekil 1.1’de verilen çalışmada; bir yatırım elemanın gelecek eğilim (trend) yönüne karar verecek “Al-Sat” karar destek sistemi modellenmiştir. Bu karar işlemlerinin, 2D görsel mum grafik dışında ek herhangi bir zaman serisine gerek kalmaksızın gerçekleştirilmesi hedeflenmiştir. Tasarlanan model, kendi başına başarılı tahminler gerçekleştirebilmekte olup modelin kullanımının asıl amacı; temel ve teknik analiz sonrasındaki karar aşamasında verilecek “Al-Sat” kararlarını güçlendirici bir öngörü sağlamaktır. Temel ve teknik analizi yapılmış bir senet için nihai karar adına sistem sonucu talep edilebileceği gibi, sisteme talebinin önceden yapılması ve ardından al veren senetler için temel ve teknik analiz gerçekleştirilerek bir portföy oluşturulabilmektedir.



Şekil 1.1. Önerilen modele ait akış diyagramı.

Finansal alanda yatırım gerçekleştiren tüm yatırımcıların ana hedefi düşük fiyattan alıp

yüksek fiyattan satış yaparak kar elde etmektir. Bu hedefle bu tez çalışmasında oluşturulan model, hisse senedi piyasası, parite analizi, endeks analizi ve diğer borsaların hisse analizinde, kısacası mum grafiği oluşturulan tüm yatırım enstrümanları için kullanılabilir etkili bir “*Al-Sat*” karar destek modelidir. Bu tez kapsamında hem finansal hem de bilimsel olan iki yönlü farklı bakış açısıyla yenilikçi bir model tasarımı gerçekleştirilmiştir. Bu model tasarım derin öğrenme tabanlı nesne tanıma algoritmaları üzerinedir. Bir nesne tanıma algoritması belirli nesnelerin yerleştirilmesi ve sınıflandırılması üzerine yapılandırılmış sistemler olarak bilinir. Bu sistemler özellikle günümüz popüler teknolojisi olan otonom sistemler için kullanılmaktadır. Bir nesne tanıma algoritmasının sorunsuz çalışmasının en temel ve önemli aşaması veri setinin oluşturulmasıdır. Öncelikle makine öğrenmeleri açısından düşünüldüğünde nesnelere ait olabildiğince fazla veriye ihtiyaç duyulmaktadır. Diğer önemli nokta, elde edilen veriler üzerinde anlamlı etiketleme işlemlerinin gerçekleştirilmesidir. Bu işlem aslında öğrenmiş insan beyninin nesnelere tanımlaması ile makinenin öğrenebileceği dile dönüştürme işlemidir. Geleneksel modeller bakımından düşünüldüğünde, bir insan yüzünü veya bedenini, bir araç sınıfını veya bir hayvan türünü herhangi bir resim üzerinden seçmek her ne kadar profesyonellik gerektirmese de on binlerce nesneyi resim üzerinde etiketleme işlemi oldukça uzun zaman alacaktır. Bu sebeple nesne tanıma algoritmalarını modellemek noktasındaki en meşakkatli olan kısmı veri seçme ve etiketleme işlemleridir.

Bu tez kapsamında geliştirilen model, yatırım araçlarının 2D mum grafikleri üzerinden eğilim yönlerini tespit ederek yatırımcıların kazanç elde etmelerini sağlayacağı yenilikçi bir bakış açısıyla gerçekleştirilmiş finansal nesne tanıma modelidir. Bu model tasarımı neredeyse tüm analizlerin temelini oluşturan sayısal verilerden tamamen uzak durup, görseller üzerine tasarlanan bir modeldir. Tamamen tez kapsamında oluşturulan veri seti üzerinde etiketlemeler ile dört farklı model eğitilerek gerçekleştirilmiştir. Etiketleme işlemleri geleneksel nesne etiketlemelerinden oldukça farklı bir çalışmadır. Bu bağlamda mutlak bir finansal bakış açısına ihtiyaç duyulmaktadır. Tüm etiketler “***Dow Kuramı***” kapsamında etiketlenmiştir ve çalışmanın en uzun süreç alan adımını oluşturmaktadır. Bu süreç sonunda büyük bir veri seti etiketli bir şekilde oluşturulmuştur.

Etiketleme işlemlerinin tamamı iki ayrı yapıda gerçekleştirilmiştir. Öncelikle çalışmaya ait düşüncenin çalışabilirliğinin ispatı adına 10000 adet etiket oluşturulmuştur. Çalışma

kapsamında oluşturulan etiketli veri seti Veri Grubu-1 (VG-1) olarak isimlendirilmiştir. Bu etiketli veri seti modellenen YoloV3 VG-1 ile eğitilmiş ve gerekli testler gerçekleştirilmiştir. Ardından çalışmanın genişletilmesi adına etiketli veri seti sayısı 20000'e çıkartılmış ve 20000 etiketli veri seti içerisinde 10000 etiket rastgele seçilmiştir. Bu aşamada oluşturulan veri seti de VG-2 olarak isimlendirilmiştir.

Bir sonraki adımda VG-2 veri seti, dört farklı nesne tanıma algoritması ile eğitilmiştir. Bu algoritmalar, karşılaştırma oluşturabilmek adına yine YoloV3 ve farklı algoritma performanslarının değerlendirilmesi adına YoloV4, Faster R-CNN ve SSD algoritmalarıdır. Bu algoritmaların her biri son veri seti olan VG-2 ile eğitilmişlerdir. Her bir algoritma, tasarlanan model için eğitim çıktısı olarak yeni ağırlık değerleri oluşturmuşlardır. Elde edilen ağırlık değerleri, algoritmalar için yeni ağırlık değerleri olarak belirlenmiş ve modeller ayrı ayrı testlere sokulmuşlardır.

Birinci test aşamasında modeller, oluşturulan veri seti içerisindeki test seti başarımlarını değerlendirmesinden geçirilmişlerdir. Bu değerlendirme işlemi, etiketleme sayısı ile modellerin ürettiği karar tespit sayılarının karşılaştırılması üzerine gerçekleştirilmiştir. Bu test değerlendirmesinde YOLO algoritmaları %100 başarı sağlamışlardır.

Diğer bir değerlendirme yöntemi olarak BIST-100'e dahil senetler içerisinde beş farklı senet seçilmiştir. Seçilen senetlerin altı aylık zaman sürecindeki günlük mum grafikleri üzerinde modellerin karar tespiti yapmaları sağlanmıştır. Yine bu aşamadaki değerlendirme sonucunda tez kapsamında oluşturulan YoloV3 VG-2 modeli en başarılı sonuca ulaşarak en çok getiri sağlanmıştır.

Son aşamada önceki değerlendirme aşamalarında en başarılı sonuç alınan YoloV3 VG-2 modeli ile bu çalışmaların fikri olarak görülen YoloV3 VG-1 arasında bir karşılaştırma işlemi gerçekleştirilmiştir. Bu karşılaştırma işlemi BIST-TUM hisse senetleri üzerinde gerçekleştirilmiştir. Sepet yöntemi kullanılarak gerçekleştirilen bu test aşamasında, herhangi bir sayı belirlenmeyip algoritmaların "AI" verdiği tüm senetler test tablosuna eklenmiştir. YoloV3 VG-1 ile testleri gerçekleştirilmiş ve tablo haline getirilen 73 adet hisse senedine karşılık tez çalışmasındaki YoloV3 VG-2 algoritması benzer ve farklı senetlerden oluşan 78 adet hisse senedi için AI sinyali üretmiştir. Yine "AI" sinyali veren senetler test tablosuna eklenmiştir. Altı aylık dönem için elde edilen test sonuçlarına bakıldığında YoloV3 VG-2 modeli ile YoloV3 VG-1'den kesinlikle daha kazançlı bir sonuca ulaşılmıştır.

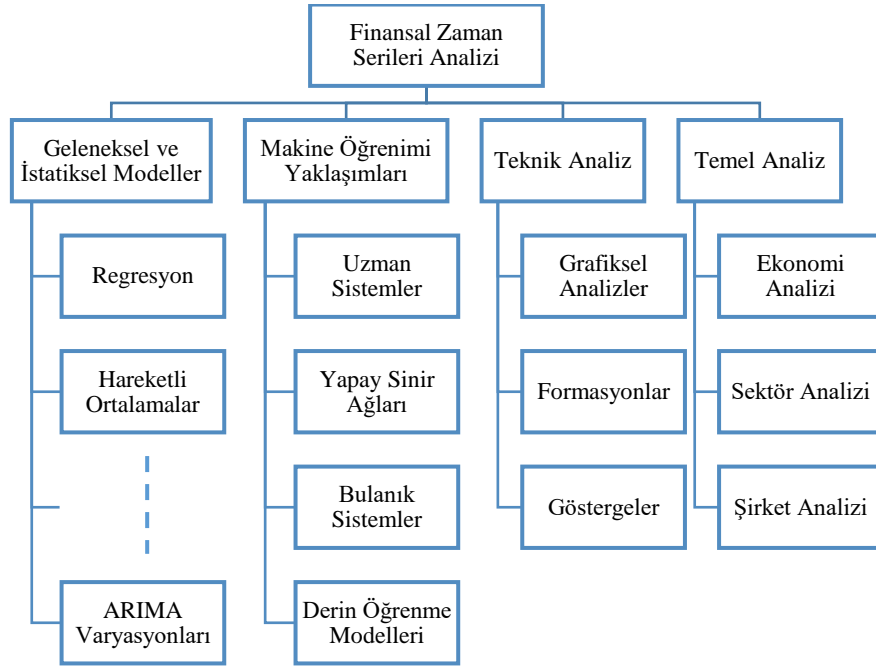
Sonuç olarak, tez kapsamında geliştirilen bu modelleme, genel literatüre bakıldığında farklı bir bakış açısına sahip olup, gün sonu kapanış verilerine sahip bir yatırım aracının 2D mum grafikleri üzerinde “*Al-Sat*” karar tespiti için oldukça yenilikçi bir yaklaşımdır. Ayrıca daha önce çalışılmamış bir farklı yaklaşımı temsil etmesi dolayısıyla literatüre öncülük edeceği düşünülmektedir.

1.3. TEZ YAPISI

Bu tez çalışması altı ana başlık altında toplanmıştır. Bu başlıklardan Bölüm1’de *giriş* olarak çalışmanın temel yapısı hakkında bilgi verilmiş olup, Bölüm 2’de *finansal zaman serisi verileri analizi* başlığı altında teknik analiz, temel analiz ve tez çalışmasının temelini oluşturan finansal mum çubuğu grafikleri ve benzer grafiksel çalışmalara değinilmiştir. Bölüm 3’te son dönemlerin popüler konusunu oluşturan *derin öğrenme tabanlı nesne tanıma algoritmaları* incelenmiştir. Özellikle otonom sistemlerinde ve görüntü ile alakalı tüm araştırmalarda kullanılan derin öğrenme tabanlı algoritmaların çalışmaları ve çeşitleri incelenmiştir. Bölüm 4 başlığı altında olan *önerilen model ve deneysel çalışmalar* da ise çalışmanın modellenmesi, buna bağlı veri seti oluşturulması ve eğitim aşamalarından bahsedilmiştir. Eğitim için gerekli donanım ve yazılım alt yapısı da bu bölümün önemli detayları arasındadır. Ayrıca tüm işlemlerin yürütülmesi adına dosya yapılanması da bu bölümde anlatılmıştır. Bölüm 5’teki *bulgular ve tartışma* bölümünde ise, yapılan çalışmaların deneysel sonuçlarına yer verilmiş ve kayıp kazançlar tablolar halinde sunulmuştur. Son olarak bölüm 6’daki *sonuç ve öneriler* başlığı altında konuyla ilgili çalışmalar ve önerilen yöntemler özetlenmekte ve gelecekteki hedeflenen çalışmalar hakkında fikirlere yer verilmiştir.

2. FİNANSAL ZAMAN SERİSİ VERİLERİ ANALİZİ

Finansal zaman serileri, finansal enstrümanların (hisse senedi, ETF (Exchange Traded Funds), döviz (USD/TR, USD/EUR, EUR/TR), altın, petrol, tahvil senedi, fon, hazine bonusu vb.) belirli zaman periyotları içerisinde oluşturduğu sayısal değerleri ifade etmektedir. Literatürde birçok alanda zaman serisi verilerinin çalışıldığı görülmektedir. Özellikle finans [36], [37], [38], ekonometri [38], işletme [39], [38] alanlarında finansal zaman serisi verilerinin analizi yapılmakta ve gelecekteki değerleri tahmin edilmektedir. Finansal zaman serisi verilerinin analizinde Şekil 2.1’de de görüleceği üzere birçok analiz yöntemleri kullanılmaktadır. Bu analizlerin amacı fiyatların değişim yönünü, zamanını ve gelecekteki ulaşabilecekleri hedefleri tahmin edebilmektir [40].



Şekil 2.1. Finansal analiz yöntemleri.

Finansal verilerin analizinde temel ve teknik analiz olarak bilinen en basit yaklaşımların yanı sıra gelişen teknoloji ile birlikte kullanılan yöntemler de farklılık göstermiştir. Bu yöntemler arasında yapay sinir ağları, destek vektör makineleri, makine öğrenme algoritmaları ve hibrit çözümler ile oluşturulan modeller geniş yer tutmaktadır. Özellikle son zamanlarda derin öğrenme algoritmalarının gelişmesiyle finansal verilerin analizinde derin öğrenme yöntemleri de kullanılmaya başlanmış ve oldukça başarılı

sonular elde edilmiřtir. Genel olarak alım-satım kararları burada anlatılan analiz yöntemlerine göre gerekleřtirilebilir. Bu bölümde analiz yöntemlerine kısaca değinilecektir.

2.1. TEMEL ANALİZ

Temel analiz, bir menkul değerin onunla iliřkili ekonomik, finansal ve diđer kalitatif (nicel verilerin analizi) ve kantitatif (sayısal verilerin sayısal yöntemler ile analizi) faktörleri incelemek suretiyle gerek değeri ölçmeye alıřarak değeri değerlendirme yöntemidir. Temel analiz, makro-ekonomik faktörler (genel ekonomik ve endüstriyel kořullar gibi) ve belli bařlı bireysel faktörler (řirketlerin finansal durumu ve yönetimi gibi) dâhil olmak üzere, menkul kıymetin değeri etkileyebilecek her řeyin arařtırılmasıdır. Bu bağlamda temel analizin bazı hedefleri mevcuttur. Bunlar;

- Ekonomik faaliyet kurumsal karı, yatırımcı tutum ve beklentilerini ve en nihayetinde menkul fiyatlarını etkilediđi için ülke ekonomisinin gidiřatını öngörmek,
- Ekonominin geneli üzerinde etkili olan güçlerin yanı sıra sektörler ve řirketlere özgü etkileri inceleyerek stok fiyat deđiřikliklerini tahmin etmek,
- Yatırım için dođru zamanı ve dođru menkul kıymetleri seçmek.

Diđer taraftan bu hedeflere ulařması için takip edilmesi gereken üç ařama mevcuttur. Bu ařamalar;

- Makroekonomik çevrenin ve geliřmelerin anlařılması (Ekonomik Analiz).
- řirketin ait olduđu sektörün beklentilerinin analiz edilmesi (Sektör Analizi).
- řirketin öngörülen performansının deđerlendirilmesi (řirket Analizi).

řekline ifade edilirler. Ü ařamalı temel analizin incelenmesi aynı zamanda EIC (Economy-Industry-Company/ Ekonomi-Sektör-řirket analizi) çerçevesi veya tepeden ařađı yaklaşım olarak da adlandırılır [41].

2.2. TEKNİK ANALİZ

Finansta teknik analiz, fiyat ve hacim bařta olmak üzere gemiř piyasa verilerini inceleyerek fiyatların gidiřatını tahmin etmeyi amaçlayan menkul kıymet analizi

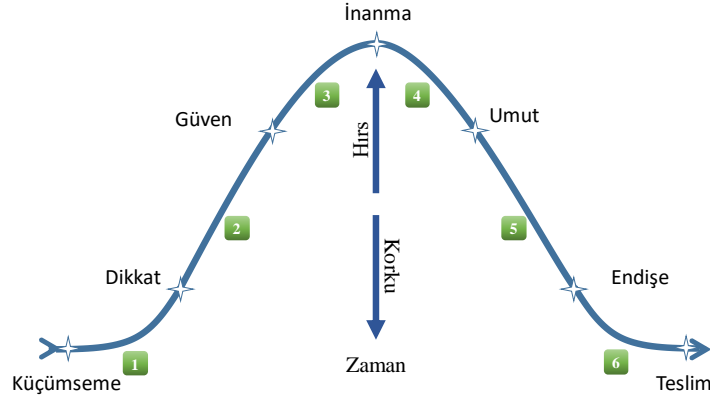
disiplinidir. Temel analiz kazançları, kar paylarını, yeni ürünleri, araştırmaları ve benzerlerini inceler. Teknik analiz ise, yatırımcıların nelerden korktuklarını, bu gelişmeler ile ilgili ne düşündüklerini ve fikirlerini destekleyecek gerekli kaynaklara sahip olup olmadıklarını inceler. Bu iki kavrama içe bakış (psikoloji) ve arz/talep denilmektedir. Ayrıca teknik analiz, grafikleri kullanarak finansal piyasalardaki fiyat desenlerini ve piyasa eğilimlerini belirlemeye ve bu desenlerden faydalanmaya çalışmaktır [42]. Analiz uzmanları çeşitli yöntemler ve araçlardan özellikle fiyat grafiklerini incelerler. Fiyat grafikleri, genel itibarı ile yatırımcıların zihinlerini yansıtmaktadır. Bu nedenle grafikler üzerinden yatırımcıların zihinlerini okumaya çalışarak onların ne düşündüklerini saptamak oldukça önemlidir.

Grafikleri kullanan uzmanlar, grafikler üzerinde iyi bilinen fiyat grafiği desenlerini ararlar. Teknik göstergeleri, hareketli ortalamaları incelerler ve destek, direnç çizgileri, kanallar ile bazı iyi bilindik desenleri bulmaya çalışırlar. Teknik analiz uzmanları, sıklıkla yukarı ve aşağı yönlü hacim, yükseliş/düşüş verileri ve diğer girdileri de kapsayan fiyatların matematiksel dönüşümleri gibi piyasa göstergelerini de yaygın olarak kullanırlar. Bu göstergeler, bir aktifin eğilim olup olmadığını ve eğer eğilim ise yönünü ve devamlılık olasılığını anlamaya yardımcı olurlar. Uzmanlar ayrıca fiyat/hacim endeksleri ile piyasa göstergeleri arasındaki ilişkileri de araştırırlar. Teknik analizde birçok yöntem vardır. Farklı yöntemleri destekleyenler (örneğin şamdan grafiği, Dow Teorisi ve Elliott dalga teorisi) diğer yaklaşımları görmezden gelebilir, ancak pek çok yatırımcı birden fazla yöntemden alınan unsurları birleştirmektedir. Bu konuda “Modern Teknik Analizin” atası olarak bilinen Charles Dow tarafından geliştirilen “*Dow Kuramı*“ sadece teknik analiz ve fiyat hareketlerini değil aynı zamanda pazar felsefesini de ifade etmektedir [43].

2.2.1. Dow Teorisi

Dow Jones, New York Menkul Kıymetler Borsası’nda işlem gören dünyanın en büyük 30 şirketinin hisse senetlerinden oluşan bir endekstir. Dünyanın en büyük endeksi olmasının yanı sıra en popüler endekslerin de başında yer almaktadır.

Dow kuramına göre Şekil 2.2’de görülen bir yatırım elemanının fiyatlarının 1 numaralı bölgede olması fiyatların küçümsenmesi anlamına gelmektedir. Geleceği göremeyen yatırımcılar hissenin bir yatay seyirde olduğunu düşünmektedirler. Alım için temel veya teknik analiz cinsinden bir karar oluşmasını beklemektedirler .



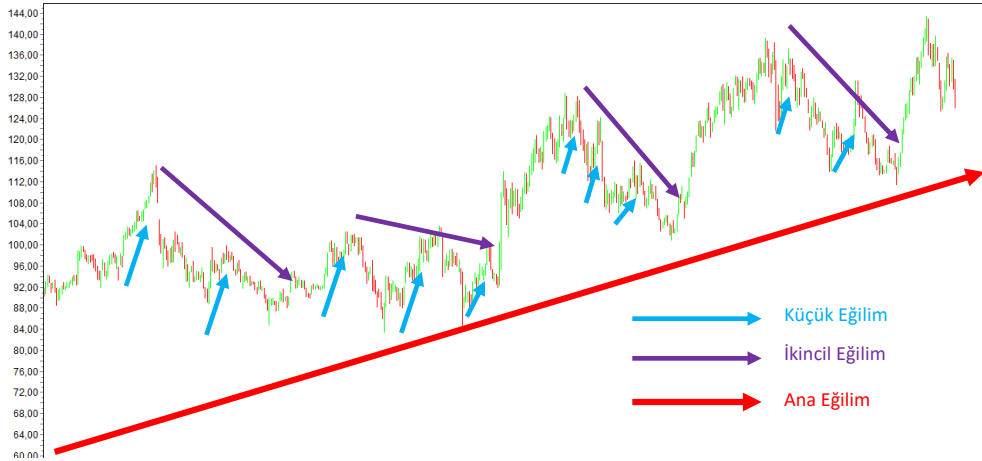
Şekil 2.2. Dow Teorisi fiyat gelişimi [44].

Zaman içerisinde yavaş yavaş yukarı doğru yönelen fiyatlar dikkat bölgesine ulaşmış ve ardından 2 numaralı bölgeye geçiş gerçekleştirmişlerdir. 1 numaralı bölgeden alım yapabilen yatırımcılar bu bölgede kar durumundadırlar ve yatırım elemanına güven duymaktadırlar. Çünkü fiyatlar iyice yükselmiş hatta zirveye ulaşmıştır. Fiyatın daha da yükselebileceğine inanan yatırımcılar için buldukları 3 ve 4. bölgeler aslında riskli bölgeler içerisinde. 3 ve 4. Bölgede satışını gerçekleştirebilen yatırımcılar için karlı bir yatırım gerçekleşmiştir. Fakat bu noktada kendilerine bir “Sat” sinyali gelmeyen veya yeni alım gerçekleştiren yatırımcılar için tehlike başlamış ve 4. bölgenin bitişinde yatırımcılar fiyatların tekrar yukarı dönüş yapacağına dair bir umut içerisine girerek daha da risk içerisine girmişlerdir. Artık 5. bölgeye ulaşan fiyatlar endişe verici duruma geçmiş 1 ve 2 numaralı bölgeden alıp satışını gerçekleştirmeyen yatırımcılar için durum kötü olmasa da güzel bir fırsatı kaçırmışlardır. Diğer taraftan 3 ve 4. Bölgeden alım yapan yatırımcılar ise zarara teslim olmuşlardır. Tüm bu durumlar göz önüne alındığında bir yatırımcı için 1 ve 2 numaralı bölgede “Al” tavsiyesi ve 3 veya 4 numaralı bölgede ise “Sat” tavsiyesi ile en iyi kazancın sağlanması şeklindeki kararlar oldukça başarılı karar anlamına gelmektedir. Genel olarak günümüzde kullanılan teknik analiz yöntemlerindeki indikatör ve formasyon gibi uygulama çalışmalarının ve finans piyasalarının tamamının amacı bu doğrultudadır. Ayrıca bu amaç için finans şirketleri büyük yatırımlar yapmakta, yatırımcılar da oldukça büyük kaynaklar harcamaktadırlar.

2.2.2. Eğilim Kavramı

Şekil 2.3’teki bir yatırım aracının zaman serisine ait küçük eğilim, ikincil eğilim ve ana eğilimleri görülmektedir. Bu eğilimler Dow kuramının doğruluğunu ispat ediyor şeklinde yorumlanabilir. Fakat buradaki asıl amaç, geçmişte görünen bu eğilim

yönlerinin tıpkı tarih yorumcuları gibi tespit edilmesi değil, yorumlardan elde edilen tecrübe ile gerçek zamanda tahminler gerçekleştirerek kazançlar elde edebilecek bir strateji geliştirmektir.



Şekil 2.3. Farklı eğilim görünümleri.

Bir yatırım aracının grafik üzerindeki ilerleyişinde onu duraksatan, durduran, çoğu kez de geri döndüren önemli fiyat seviyeleri bulunur. Fiyatların hareketini sağlayan insan doğası, grafik üzerinde belli seviyelerde yatırım kararlarını sert şekilde değiştiriyorsa fiyat seviyeleri bunu mutlaka dikkate alacaktır. Tarihsel zaman dilimi içerisindeki ilerleyişi esnasında önemli ölçüde engelle karşılaşan ve belli bir fiyat seviyesini aşamayan hareket bu noktayı tarihsel gelişim olarak hafızasına alır ve bir daha aynı fiyat seviyesiyle karşılaştığı esnada benzer hareketlerde bulunma eğilimi içerisine girer. Yatırım aracı fiyatlarının yükselişi esnasında takılacağı önemli durak noktalarının yatay düzlemdeki fiyat konumuna direnç, alçalışı esnasında karşılaşılabilecek engelle ise destek adı verilmektedir [45].

Destek ve direnç seviyelerinin yaklaşan fiyatla ilgili karakteristik eğilimleri yatırımcı açısından ilginç pozisyon olanakları doğurur. Zira bu seviyelerde fiyat hareketinin genel tecrübe edinilmiş davranış mekaniği basit pozisyonlarla para kazanma imkânı doğurabilir [46].

Sonuç olarak bir finansal zaman serisi veri grafiği mutlaka eğilimlere sahiptir. Bu durum yatırım aracının manipüle edilmediği sürece belirli noktalarda çıkış göstereceği ve belirli noktalarda ise iniş göstereceği anlamına gelmektedir. Her şey olup bittikten sonra yatırım aracının davranışları hakkında bilgilerin toplanarak bu bilgilerin gelecek için yorumlanmasının bu çalışmada gerçekleştirilen model yaklaşımına katkı

sağlayacağı düşünülmektedir.

2.3. FİNANSAL ZAMAN SERİSİ VERİLERİNİN İSTATİKSEL / MATEMATİKSEL ANALİZİ

Zaman serileri tahmini, aynı değişkenin geçmiş değerlerinin temeldeki bir ilişkiyi açıklamak için toplandığı ve analiz edildiği önemli bir tahminleme alanıdır [47]. Finansal zaman serisi verilerinin istatistiksel / matematiksel analizi verilere ait en küçük, en büyük, ortalama, hareketli ortalama, varyans, kovaryans, standart sapma, otokorelasyon gibi değerlerin hesaplanması şeklinde ifade edilir. Ayrıca matematiksel analiz olarak gelen zaman serisi verileri üzerinde regresyon analizi, fonksiyon tahmini gibi uygulamalar gerçekleştirilebilir. Matematiksel fonksiyonlar yardımı ve verilerin geçmiş değerleri üzerindeki analizleri ile gelecekte olabilecek değerleri tahmin edilir. Regresyon analizi ile de gelen verilerin birbirleriyle olan ilişkisi ölçülür, var olan ilişki matematiksel bir fonksiyonla ifade edilir [40].

En önemli ve yaygın olarak kullanılan zaman serisi modellerinden biri, Box-Jenkins (ARIMA) modelidir. Box-Jenkins yöntemi, tek değişkenli bir model olup kısa dönem tahmininde oldukça başarılı sonuçlar vermektedir. Bu yöntem, AR(p) (otoregresif), MA(q) (hareketli ortalama), ARMA(p,q) (otoregresif hareketli ortalama) ve ARIMA (p,d,q) (bütünleşik otoregresif hareketli ortalama) modeller arasından en uygun modeli seçerek tahmin yapmak amacıyla kullanılır [48], [49], [50].

2.4. FİNANSAL ZAMAN SERİSİ VERİLERİNİN GRAFİKSEL ANALİZİ

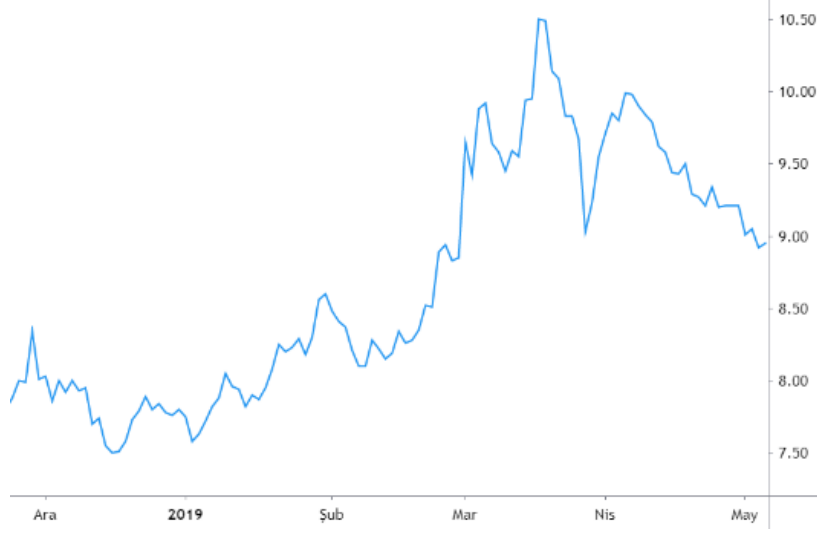
2.4.1. Grafik Türleri

2.4.1.1. Çizgi Grafik

Çizgi grafik, noktalar halindeki bir dizi bilgiyi düz doğru dilimleriyle birbirine bağlayarak görüntüleyen bir grafik türüdür. Birçok alanda da yaygın olarak kullanılmaktadır. Dağılım grafiğinin genişletilmiş halidir ve doğru dilimlerinin tek tek ölçümlerini temsil eden bir dizi noktanın birleştirilmesi ile oluşturulur. Bir çizgi grafik genellikle verilerde belli zaman aralıklarında gerçekleşen eğilimleri görselleştirmek için kullanılır, dolayısıyla doğru genellikle kronolojik olarak çizilir [51].

Çizgi grafik, bir dizi veri noktasının sürekli bir çizgiyle birleştirildiği, bir varlığın

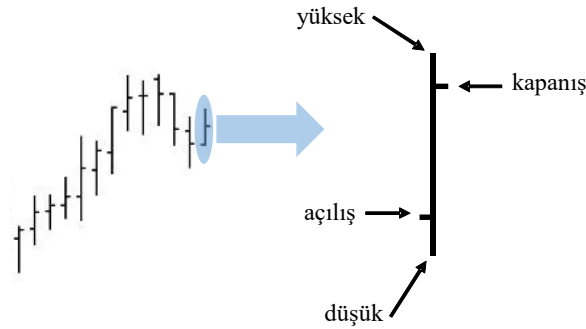
geçmiş fiyat işlemlerinin Şekil 2.4'te görüldüğü gibi grafiksel bir temsildir. Bu grafik türü, finasta kullanılan en temel grafik türü olup genellikle bir menkul kıymetin zaman içinde kapanış fiyatlarını gösterir. Çizgi grafikler herhangi bir zaman diliminde kullanılabilirken, çoğunlukla günlük fiyat değişikliklerinde kullanılırlar. Bir çizgi grafiğinin anlaşılması kolaydır ve form olarak basittir.



Şekil 2.4. Bir hisse senedine ait çizgi grafik görünüm [52].

2.4.1.2. Çubuk Grafik

Açılış-yüksek-düşük-kapanış [OHLC (open-high-low-close chart)] grafiği, genel olarak bir finansal aracın fiyatında zaman içinde gerçekleşen hareketleri göstermek için kullanılan bir grafik türüdür. Şekil 2.5'teki dikey çizgi, bir birim zaman içindeki fiyat aralığını (en yüksek ve en düşük fiyatları), örneğin bir gün veya bir saati gösterir. Doğrunun her iki tarafındaki işaret çentiklerinden soldaki açılış fiyatını (örneğin, günlük çubuk grafikte bu, o günün açılış fiyatıdır), sağdaki ise o zaman aralığındaki kapanış fiyatını gösterir.



Şekil 2.5. Tek çubuk gösterim.

Şekil 2.6'daki gibi bir hisse senedine ait çubuklar, fiyatların o zaman aralığında yükselmesine veya düşmesine bağlı olarak farklı renk tonlarında gösterilebilir.

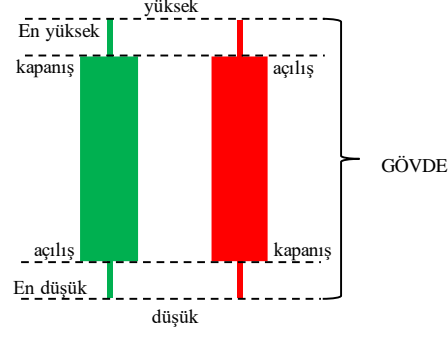


Şekil 2.6. Bir hisse senedine ait çubuk grafik görünüm [52].

2.4.1.3. Mum Grafik

Sınıf olarak teknik analiz yöntemi içerisine dahil olacak bir diğer farklı tahmin yöntemi mum formasyonlarıdır. Teknik analizin bu ilk türü, modern teknik analiz teorilerinin öncüsü olan Charles Dow'un 1900'lü yıllar civarında ortaya attığı teknik analiz yaklaşımından farklı olmakla beraber, benzer ilkeler üzerine kurulmuştur [44]. Steve Nison'a [53] göre ilk defa 1850'li yıllarda ortaya çıkmıştır. Mum çubuklarının gelişimi ve önemli bir teknik analiz haline gelmesine neden olan kişi Japonya'nın Sakata kendinde tanınmış ünlü pirinç tüccarı "Homma" olmuştur. Japon tüccarların pirinç alım satım sözleşmelerindeki fiyatlara bakarak ilerideki fiyatları tahmin etmeye yönelik kullandıkları bir yöntemdir. Yıllar içinde daha fazla kişinin kullanması üzerine yapılan çalışmalar, grafik teknolojilerinin ilerlemesi ve bugünkü haline gelmesini sağlamıştır.

Mum grafiğinin oluşturulabilmesi için bir açılış değerinin mevcut olması ön koşuldur. Açılış fiyatının yanında kapanış fiyatı, gün içi en yüksek ve gün içi en düşük değerler kullanılarak mum grafik oluşturulur. Mum grafiklerin çubuk grafiklerden en belirgin farkı, bir gövdeye sahip olmalarıdır. Şekil 2.7'de görüldüğü üzere mumun kalın kısmı seansın açılış ile kapanış değerleri arasındaki mesafeyi göstermektedir. Bu mesafenin BIST gibi piyasalarda mutlaka maksimum ve minimum sınırları bulunmaktadır.



Şekil 2.7. Mum grafik [54].

Yeşil gövdeli mum, kapanış fiyatının açılış fiyatından daha yukarıda olduğunu yani talebin fazla olduğunu göstermekte; kırmızı gövdeli mum ise, arzın fazla olduğunu yani fiyatların yukarıda açılıp aşağıda kapandığı bir seansın varlığını göstermektedir [54]. Gövdelerin ucunda bulunan çizgi gösterimleri gölgeyi temsil etmektedir. Bu bağlamda oluşmuş bir hisse senedine ait günlük mum grafik örneği Şekil 2.8’de verilmiştir.



Şekil 2.8. Bir hisse senedine ait mum grafik görünüm [52].

2.4.1.4. Nokta ve Şekil Grafik

Bu grafik türü, diğer tüm yöntemlerin aksine, fiyatı zamana karşı göstermemesi bakımından benzersizdir. Bunun yerine, fiyatı yön değişikliklerine göre gösterir; fiyat yükseldikçe *Xs* sütunu, düştükçe *Os* sütunu çizilir. Bu yöntem 100 yıldan fazla süredir kullanılmaktadır.

Çok sayıda kaynağın ve örneğin alıntılı olduğu “Nokta ve Şekil Grafiği için Kılavuz”

(The Definitive Guide to Point and Figure) adlı eserinde Du Plessis, nokta ve şekil grafiğinin fiyat kayıt sisteminden bir grafik yöntemine dönüşümünü açıklamıştır. Yatırımcıların fiyatları sütunlara yazarak takip etmesinden esinlenilerek fiyat kayıtlarındaki desenleri önce “dalgalanma grafikleri” sonra da “şekil grafikleri” olarak isimlendirilmişlerdir. Fiyat sütunlarından grafiğe aktarırken sayılar yerine *X*’ler ve *O*’lar kullanılmış ve bu grafiklere *nokta ve şekil grafikler* denilmiştir. Örnek olarak *Xs* ve *Os* ile çizilmiş bir hisse senedine ait modern nokta ve şekil grafiği Şekil 2.9’da verilmiştir. *Xs* sütunları yükselen fiyatları, *Os* sütunları ise düşen fiyatları göstermektedir [42].



Şekil 2.9. Bir hisse senedine ait nokta ve şekil grafik görünüm [52].

2.4.1.5. Kagi Grafik

Kagi grafik, fiyat hareketlerini izlemek ve hisse senedi satın alma kararı vermek için kullanılan bir grafik türüdür. Zamandan çoğunlukla bağımsız olması sebebiyle, şamdan grafik gibi geleneksel hisse senedi grafiklerinden farklıdır. Bu özellik, rastgele gürültüyü azaltan bir grafik ortaya çıkartılmasına yardımcı olmuştur. Fiyat hareketlerini net olarak göstermedeki etkililiği sebebiyle, Kagi grafik yatırımcıların hisse senetleri ile ilgili daha sağlıklı kararlar almak için kullandıkları çeşitli grafiklerden biridir. Bu grafiğin en önemli faydası, zamandan bağımsız olması ve yön değişikliğinin sadece belirli bir miktara ulaşıldığında gerçekleşmesidir. Kagi grafik ilk olarak Japonya’da, Japon borsasının işlemlere başladığı 1870’lerde geliştirilmiştir [55].

Kagi grafiklerinin zaman eksenini yoktur [56]. Bir kagi grafiği, kısa yatay çizgilerle birbirine bağlanmış bir dizi dikey çizgi ile oluşturulur. Çizgilerin kalınlığı ve yönü hissenin veya aktifin fiyatına bağlıdır. Fiyat bir önceki dikey çizginin en üstüne veya en

altına ulaştığında çizginin kalınlığı değişir. Fiyat, önceden ayarlanmış geri dönüş miktarına ulaştığında ki bu miktar genellikle %4'tür, çizginin yönü değişir. Bir yön değişikliği meydana geldiğinde, zıt yön çizgileri arasında kısa bir yatay çizgi çizilir. Alternatif olarak ince ve kalın çizgiler yerine, Şekil 2.10'daki yeşil/kırmızı örneğinde olduğu gibi, farklı renkteki çizgiler kullanılabilir [42].



Şekil 2.10. Bir hisse senedine ait kagi grafik görünüm [57].

Çizgi kalınlığındaki/renklerdeki değişiklikler, işlem sinyalleri oluşturmak için kullanılır. Alış sinyalleri Kagi çizgisi inceden kalına/kırmızıdan yeşile doğru gittiğinde, satış sinyalleri ise çizgi kalından inceye/yeşilden kırmızıya döndüğünde ortaya çıkar.

2.5. FİNANSAL PİYASALARDA AL-SAT KARAR ANALİZLERİ

2.5.1. İSTATİKSEL/MATEMATİKSEL KARARLAR

Yıllar boyunca çeşitli zaman serileri tahmin modelleri geliştirilmiştir. Rastgele yürüyüş, otoregresif (AR), hareketli ortalama (MA), otoregresif hareketli ortalama (ARMA) ve otoregresif bütünleşik hareketli ortalama (ARIMA) doğrusal bir zaman serisinin geçmiş gözlem değerleriyle gelecekteki değerlerini öngörmek için yaygın olarak kullanılan istatistiksel tahmin modellerindendir [47]. Bu modellerin kısıtı, zaman serisinin doğrusal olması şartıdır. Bu sebepten dolayı, bu tür modellerin kısıtlarının üstesinden gelmek amacıyla literatürde çeşitli doğrusal olmayan modeller geliştirilmiştir. Bunlar çift doğrusal zaman serisi modeli, eşiksel otoregresif (TAR) modeli, otoregresif koşullu değişen varyans modeli gibi modellerdir. Doğrusal olmayan bu modellerde bir miktar iyileşme görülmesine rağmen, bu modellerin genel tahmin problemlerinde kullanma

kazanımı sınırlıdır [58]. Literatürde ARIMA/ARMA modellerinden elde edilen zaman serisi verilerinin, k-means ve k-medoid algoritmaları ile kümeleme çalışmaları mevcuttur [59]. Bu modeller, spesifik doğrusal olmayan modeller için geliştirildiğinden, zaman serilerindeki diğer doğrusal olmayan türleri modelleme yeteneğine sahip değildirler. Son zamanlarda zaman serileri tahminlerinde sıklıkla kullanılan diğer bir yöntem ise, yapay sinir ağlarıdır. Yapay sinir ağları, birçok göze çarpan özelliği ile zaman serilerinde hem doğrusal hem de doğrusal olmayan verilerin tahmini için popüler bir yöntemdir.

2.5.2. GRAFİKLER ÜZERİNDE GERÇEKLEŞTİRİLEN KARARLAR

2.5.2.1. Formasyonlar

Grafik Desenleri:

Grafik deseni, fiyatların grafiğe dönüştürülmesi ile görsel olarak oluşturulan bir desendir. Hisse senedi ve emtia piyasaları ticaretinde, grafik deseni çalışmaları teknik analiz sırasında önemli rol oynamaktadır. Veriler yerleştirilirken, genellikle kendiliğinden ortaya çıkan ve belli bir süre boyunca tekrar eden bir desen vardır. Grafik desenleri, geri dönüş veya devam sinyalleri olarak kullanılırlar. Bazı insanlar grafik desenlerini tanıyarak gelecekteki hisse senedi fiyatlarını ve böylelikle elde edecekleri karı tahmin edebildiklerini iddia ederken, bazıları buna “geçmiş performans gelecekteki sonuçların garantisi değildir” sözünü alıntıyla olarak yanıt verirler ve grafik desenlerinin insanların bilinçaltında oluşan yanılsamalardan başka bir şey olmadıklarını iddia ederler. Bazı iktisat teorilerine göre, gelecekteki hisse senedi fiyatlarını ve bundan elde edilecek karı tahmin etmenin bir yolu olsaydı, bu teknikler yeterince insan kullandığında etkisizleşirlerdi ve faydalı olamazlardı. Öte yandan, diğer insanların piyasanın hareketlerine dair tahminlerini tahmin edebiliyorsanız, bu önemli ve değerli bir bilgidir [42].

Desenler:

Edwards ve Magee tarafından anlaşılır biçime getirilen ve tacirler ile yatırımcılar tarafından yaygın olarak kullanılan “klasik” grafik desenlerine örnek olarak aşağıdakiler gösterilebilir:

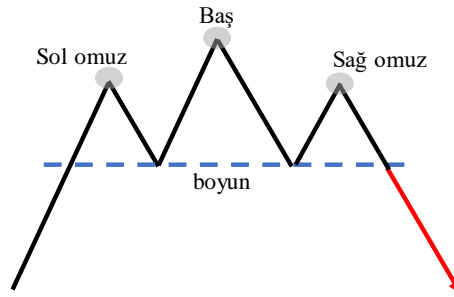
- Omuz-baş-omuz
- Fincan ve kulp
- Üçlü tepe ve üç dip
- Eğilim çizgileri
- İkili tepe ve ikili dip
- Genişleyen tepe

- Fiyat kanalları
- Üçgen (teknik analiz)
- Elit desenler
- Takoz deseni
- Bayrak ve flama desenleri

Omuz-Baş-Omuz (Grafik Deseni)

Bu desen en iyi bilinen dönüş desenlerinden biridir. Teknik analiz tablosunda, bir fiyat eğilimi boğa veya ayı eğiliminden dönüş sürecindeyse, karakteristik bir desen şekillenir ve dönüş formasyonu olarak kabul edilir.

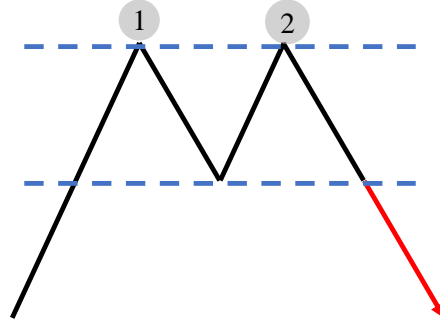
Omuz-Baş-Omuz-Tepe formasyonu bir sol omuz, bir baş ve bir sağ omuz ile boyun çizgisi olarak çizilmiş bir doğrudan oluşur. Şekil 2.11’de verilen örnek bir grafik üzerinde de görüleceği gibi sol omuz, hacmin belirgin ölçüde yüksek olduğu geniş bir hareketin sonunda oluşur. Sol omuzun zirvesi oluştuktan sonra takip eden bir reaksiyon gerçekleşir ve fiyatlar genellikle düşük hacimde görülen belli ölçüde aşağıya kayar. Fiyatlar, baştan normal veya yüksek hacimle yükselişe geçer ve takip eden aşağı doğru reaksiyon daha düşük hacimlidir. Sağ omuz, fiyatlar tekrar yükseldiğinde, ancak Baş denilen merkezi tepe noktasının altında kaldığında ve neredeyse sol omuz ile baş arasındaki ilk çukura eşit hizaya veya en azından sol omuzun tepe noktasının altına düştüğünde oluşur. Sağ omuz oluşumunda hacim, sol omuz ve baş oluşumuna kıyasla daha düşüktür. Sol omuz, baş ve sağ omuzun dip noktaları boyunca bir boyun çizgisi çizilir. Fiyatlar bu boyun çizgisini aştığında ve sağ omuz oluştuktan sonra düşmeye devam ettiğinde, baş-omuz-baş tepe formasyonu kesin şekilde oluşmuş demektir. Fiyatların düşüş eğilimine devam etmeden önce yaka çizgisine değmek için geri çekilmesi oldukça muhtemeldir [60], [61].



Şekil 2.11. Omuz-Baş-Omuz-Tepe formasyon görünüm.

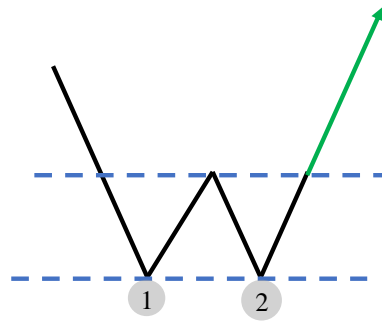
Omuz-Baş-Omuz-Dip formasyonu basitçe omuz-baş-omuz-tepe formasyonunun tersidir ve genellikle aynı eğilimde ve duyarlılıkta bir değişime işaret eder. Formasyon, Şekil 2.12’de görüldüğü üzere omuz-baş-omuz-tepe formasyonunun tersi olarak baş aşağıdır. Desenin boyun çizgisi bir destek seviyesini temsil eder ve bu seviye

de görüldüğü üzere fiyat boyun çizgisinin altına düştüğünde tamamlanır ve doğrulanır, bu da daha ileri seviyede fiyat düşüşünün an meselesi veya çok muhtemel olduğuna işaret eder. İki tepe arasındaki süre, aynı zamanda ikili tepe deseninin varlığı için belirleyici bir unsurdur. Tepeler aynı seviyede ancak zamanda birbirine çok yakınsa, yüksek ihtimalle konsolidasyonun parçasıdır ve eğilim devam edecektir. Hacim, bu formasyonu yorumlamaya yarayan bir diğer göstergedir [63][42].



Şekil 2.14. İkili Tepe formasyon görünüm.

İkili dip, gerileyen bir pazardaki son formasyondur. Şekil 2.15'te görüldüğü üzere fiyatlardaki ters ilişki hariç, ikili tepe ile aynıdır. Desen, boyun çizgisini belirleyen yerel tepe noktası ile ayrılmış iki minimum fiyat ile oluşur. İkili tepe formasyonu ile ilişkilendirilen kuralların çoğu, ikili dip deseni için de geçerlidir. Hacim, fiyatlar ikinci dipte yatay seyrederken, yeniden yükselme sırasında formasyonun oluşumu için belirgin bir artış göstermelidir [63].

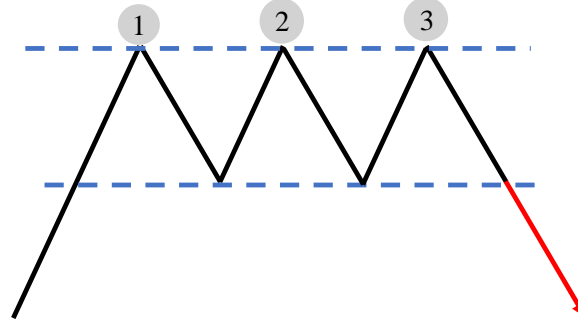


Şekil 2.15. İkili Dip formasyon görünüm.

Üçlü tepe ve üçlü dip, hisse senetlerinin, emtiaların, para birimlerinin ve diğer aktiflerin teknik analizinde kullanılan birbirine zıt grafik desenleridir.

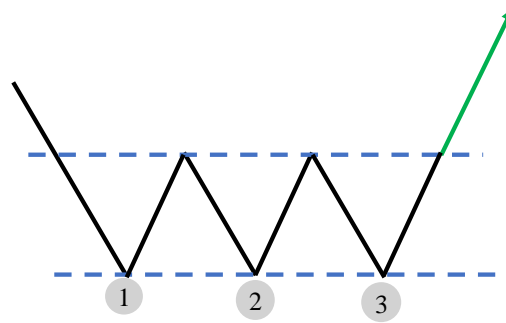
Üçlü tepe formasyonunda tepelerin oluşumu, yükselen piyasa hareketlerindeki ikili tepelerin oluşumundan daha nadirdir. Hacim, ikinci yeniden yükseliş sırasında

genellikle düşük, *üçüncü tepe* oluşumu sırasında ise daha da düşüktür. Şekil 2.16'daki gibi tepe noktaları ile de ikili tepe oluşturanlar gibi eşit aralıklarla hizalanmayabilir. Araya giren çukurlar tam olarak aynı seviyede dibe vurmayabilir, örneğin ya birinci ya da ikinci daha alçak olabilir. *Üçlü tepe*, *üçüncü tepedeki* fiyat gerilemesi üç tepe noktası arasındaki en alçak çukurun *dibinin* altına düştüğünde doğrulanmış olur [64].



Şekil 2.16. Üçlü Tepe formasyon görünüm.

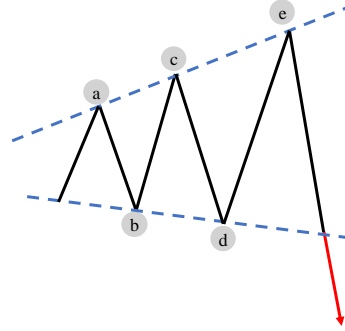
Üçlü dip. Üçlü tepenin oluşumunda geçerli olan kuralların çoğu, *üçlü dipte tersine çevrilebilir*. Hacim söz konusu olduğunda, üçüncü alt dip düşük hacimde olmalı ve bu dipten gelen yeniden yükseliş, aktivitede belirgin bir artış göstermelidir. Şekil 2.17'deki gibi üçlü dip oluşumları birikim döneminde ortaya çıkar [42].



Şekil 2.17. Üçlü Dip formasyon görünüm.

Genişleyen tepe formasyonu, tepelerde diplere göre çok daha sık görülür. İşlem yapmak için zor bir oluşumdur. Oluşum genellikle *ayrı piyasasının* habercisidir. Bu oluşumda piyasalar kontrolden çıktığı için, "*aklı olan işlem yapmaz*" sözü yaygın bir söylemdir. Oluşum sırasında, satışların çoğu erken aşamada büyük oyuncular tarafından tamamlanır. Fiyat öngörülemez biçimde salınmaya devam eder ve kimse bir sonraki hareketin nerede biteceğinden emin olamaz. Hisse senedi hacmine gelince, çok düzensizdir ve bir sonraki hareketin yönüne dair hiçbir ipucu vermez. Şekil 2.18

örneğinde görüldüğü gibi genişleyen tepe formasyonunda, *beş küçük çaplı dönüşü* büyük bir düşüş izler [65].



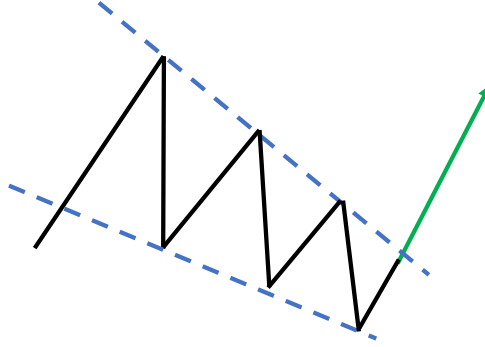
Şekil 2.18. Genişleyen Tepe formasyon görünüm.

Şekilde, hisselerin fiyatı beş kez dönmekte, dönüş noktası *d* geri dönüş noktası *b*'den daha alçak bir noktada meydana gelmekte, dönüş noktası *c* ve *e* ise sırası ile dönüş noktası *a*'dan daha yüksekte meydana gelmektedir. Fiyat iki noktanın (*b&d*) altında kırılım yapmadıkça ve düşmeye devam ettikçe kimse eğilimden emin olamaz. Şekilde de görüldüğü üzere genişleyen tepe doğrulanmaktadır [42].

Takoz deseni, fiyatlarda yükseliş eğilimi (yükselen takoz olarak bilinir) veya düşme eğilimi (düşen takoz olarak bilinir) ile birlikte daralma aralığı ile karakterize edilir. Takoz deseni birincil eğilimin geçici olarak duraksaması olarak kabul edilir. Alım satım faaliyetlerinin bir desen oluşturan yakınsak düz çizgilerle sınırlandırıldığı bir oluşum türüdür. Bu desen aynı yöne işaret eden, yükselen veya düşen bir eğime sahiptir.

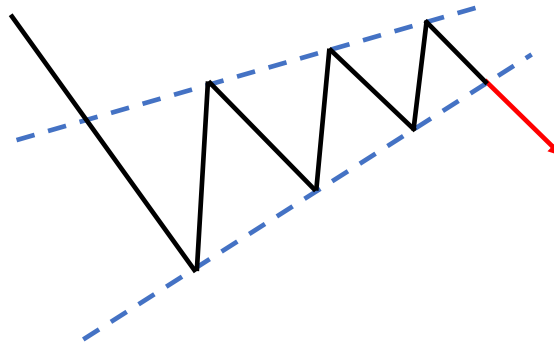
Düşen takoz deseni, piyasa bir daralma aralığı ile birlikte yeni alçak dipler ve yeni alçak tepeler yaptığında oluşan grafik deseni ile karakterize edilir. Bu desen bir düşüş eğiliminde görüldüğünde, aralığın daralması düşüş eğiliminin güç kaybettiğine işaret ettiği için, dönüş deseni olarak nitelendirilir. Bu desen bir yükseliş eğiliminde görüldüğünde ise, piyasa aralığı düzeltmeye doğru giderek daralırken, düşüş eğiliminin gücünü kaybettiğine ve yükseliş eğiliminin kaldığı yerden devam edeceğine işaret ettiği için boğa piyasası deseni olarak nitelendirilir. Düşen bir takozda, her iki sınır çizgisi soldan sağa doğru aşağı eğim yapar. Üst çizgi, alt çizgiden daha dik bir açıyla iner. Hacim azalmaya devam eder ve fiyatların daralması nedeniyle işlem hareketleri yavaşlar. Nihayetinde kırılma noktası gelir ve kırılımından sonra işlem hareketleri farklılaşır. Fiyatların, düşen takozun belirli sınır çizgilerinin dışına çıktıklarında, temel

eğilime devam etmeden önce yan taraflara ve çanağın dışına doğru hareket etme olasılıkları daha yüksektir.



Şekil 2.19. Düşen Takoz formasyon görünüm.

Yükselen takoz deseni, piyasa bir daralma aralığı ile birlikte yeni yüksek tepeler ve yeni yüksek dipler yaptığıında oluşan grafik deseni ile karakterize edilir. Bu desen bir yükseliş eğiliminde görüldüğünde, daralma aralığı yükseliş eğiliminin gücünü kaybettiğine işaret ettiği için, dönüş deseni olarak nitelendirilir. Bu desen bir düşüş eğiliminde görüldüğünde ise, piyasa aralığı düzelmeye doğru giderek daralırken, düzelmenin gücünü kaybettiğine ve düşüş eğiliminin kaldığı yerden devam edeceğine işaret ettiği için ayı piyasası deseni olarak ta nitelendirilir. Şekil 2.20’de de görüldüğü gibi yükselen takozda, her iki sınır çizgisi soldan sağa doğru yukarı eğim yapar. Her iki çizgi de aynı yöne işaret etse de alt çizgi üst çizgiye kıyasla yukarıya doğru daha dik bir açıyla yükselir. Fiyatlar genellikle alt sınır çizgisini aştıktan sonra düşer. Hacimlere gelince, her yeni fiyat artışı veya dalgalanmasıyla düşmeye devam ederler, bu da yüksek fiyat seviyesinde talebin azaldığına işaret eder [66].

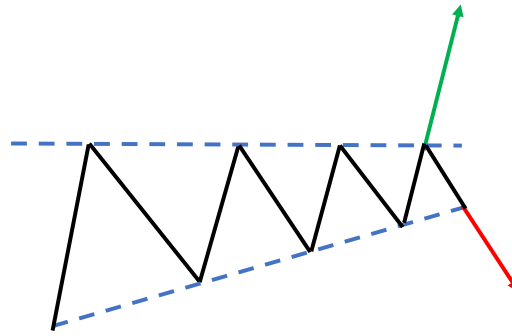


Şekil 2.20. Yükselen Takoz formasyon görünüm.

Üçgen Grafik deseni, ismini fiyat aralığında ve yakınsak eğilim çizgilerinde daralma olmasından, dolayısıyla üçgen şekline benzemesinden alır. Üçgen desenleri üç

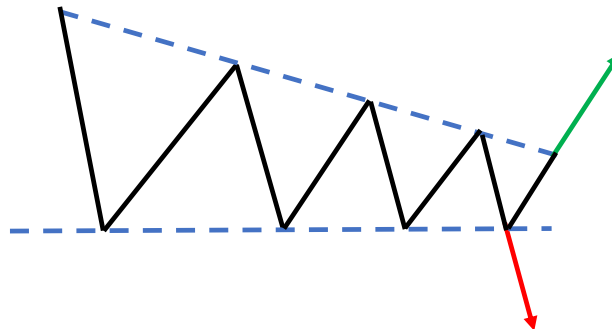
kategoriye ayrılır: yükselen üçgen, alçalan üçgen ve simetrik üçgen. Üçgenin şekli önemli olmakla birlikte, daha önemli olan, piyasanın üçgenin sınırları dışına çıktığında hareket ettiği yöndür. Son olarak, üçgenler bazen dönüş desenleri yani ana eğilimin terse dönüşü olarak görülebilirken, normalde devamlılık desenleri (yani ana eğilimin devamı) olarak görülürler.

Yükselen üçgen, piyasa yeni yüksek dipler ve aynı seviyede tepeler yaptığında oluşur. Bu desenler normalde yükseliş eğiliminde görülür ve satın alma talebi deseninin tepe direnç çizgisine doğru yükselerek kontrolü giderek daha fazla ele geçirdiği için devam deseni olarak nitelendirilir. Normalde bu desen bir yükseliş eğiliminde görülebilir, ancak bir düşüş eğiliminde görülür ise, güçlü bir dönüş sinyali olarak işlev görebileceğine dikkat edilmelidir. Şekil 2.21'deki grafikte yükselen üçgen örneği görülmektedir [42].



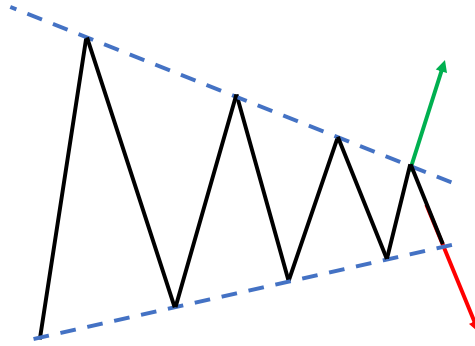
Şekil 2.21. Yükselen Üçgen formasyon görünümü.

Alçalan üçgen, piyasa yeni alçak tepeler ve aynı seviyede dipler yaptığında oluşur. Bu desenler normalde bir düşüş eğiliminde görülür ve ayılar deseninin dip direnç çizgisine doğru düşerek kontrolü giderek daha fazla ele geçirdiği için devam deseni olarak nitelendirilir. Normalde bu desen formunu bir düşüş eğiliminde görebilirsiniz, ancak bir yükseliş eğiliminde görürseniz, güçlü bir dönüş sinyali olabileceğine dikkat edilmelidir. Şekil 2.22'deki görselde örnek gösterim görülmektedir [42].



Şekil 2.22. Alçalan Üçgen formasyon görünüm.

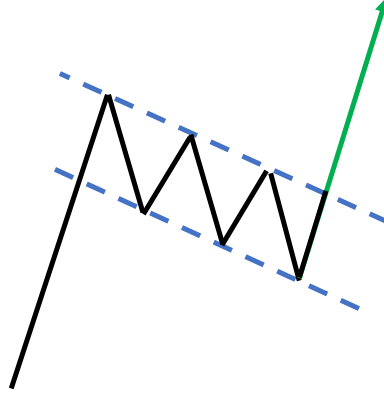
Simetrik üçgen, piyasa yeni alçak tepeler ve yeni yüksek dipler yaptığıında oluşur ve piyasa aralığının daralması ne boğaların ne de ayıların kontrolü elinde tutamadığına işaret ettiği için kontrolsüz piyasalar ile ilişkilendirilir. Bu desen bir yükseliş eğiliminde oluşursa, piyasanın yukarı yönde patlama yapması halinde devam, aşağı yönde patlama yapması halinde terse dönüş deseni olarak nitelendirilir. Benzer şekilde, eğer desen bir düşüş eğiliminde oluşursa, piyasanın aşağı yönde patlama yapması halinde devam deseni, yukarı yönde patlama yapması halinde terse dönüş deseni olarak nitelendirilir. Şekil 2.23'teki görselde örnek görülmektedir [42].



Şekil 2.23. Simetrik Üçgen formasyon görünüm.

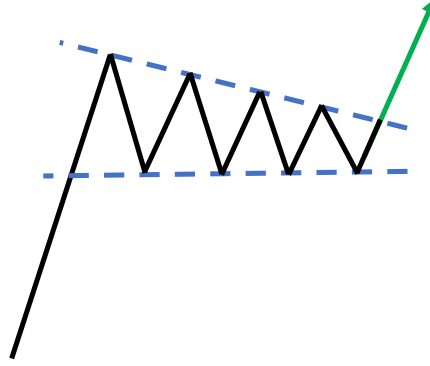
Bayrak ve flama desenleri, finansal olarak işlem gören aktiflerin (hisse senetleri, bonolar, vadeli işlemler, vb.) fiyat grafiklerinde yaygın olarak bulunan desenlerdir. Bu desenler, net bir fiyat eğilimi yönü, ardından bir konsolidasyon ve belirli bir aralıkta seyir hareketi, bunun da ardından eğilimin kaldığı yerden devam etmesi ile karakterize edilir.

Bayrak deseni, iki paralel çizgi ile çevrilidir. Bu çizgiler ya düz ya da birincil piyasa eğiliminin zıt yönünde olabilir. Piyasadaki birincil eğilimi temsil eden bir çizgi ile direk oluşur. Desen, büyük bir hareketin ardından birincil eğilimine devam etmeden önce piyasanın potansiyel olarak “mola” vermesi şeklinde nitelendirilmektedir. Şekil 2.24'te örnek grafik gösterim mevcuttur [67].



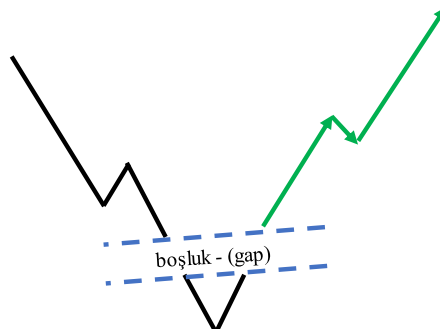
Şekil 2.24. Bayrak formasyon görünüm.

Flama deseni, oluşumu ve çıkarımları bakımından bayrak deseni ile aynıdır; tek fark flama deseninin konsolidasyon aşamasının paralel eğilim çizgileri yerine yakınsak eğilim çizgileri ile karakterize olmasıdır. Şekil 2.25’te örnek grafik gösterim mevcuttur [67].



Şekil 2.25. Flama formasyon görünüm.

Ada dönüş formasyonu, belli bir fiyat aralığı içindeki, kendisinden sonraki hareketten ayrı olarak gerçekleşen kompakt işlem hareketi olarak tanımlanabilir. Bu ayrılmanın sebebi “*Bitkinlik Boşluğu*”, zıt yöndeki müteakip hareketin ortaya çıkma sebebi ise “*Kopuş Boşluğu*” ’dur. Formasyona ait grafiksel gösterim Şekil 2.26’da görülmektedir.



Şekil 2.26. Ada Dönüş formasyon görünüm.

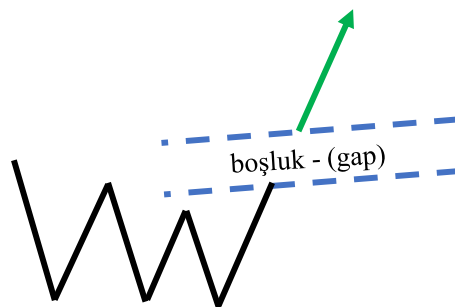
Ada dönüş formasyonlarına dikkatlice bakıldığında, bu formasyonların bir “*Bitkinlik Boşluğu*” ’ndan oluştuğu ve müteakip hareketin ardından “*Kopuş Boşluğu*” ’nun geldiği görülür. Nadiren, adayı tamamlayan *Kopuş Boşluğu*, reaksiyonun bir sonucu olarak gerçekleşen geri çekilme ile birkaç gün içinde doldurulur. Ada dönüş, omuz-baş-omuz formasyonlarının tepe noktasında veya dönüşünde ortaya çıkabilir.

Boşluk (Gap), doldurulmamış bir alan veya aralık olarak tanımlanır. Bir teknik analiz tablosunda, boşluk hiç işlem gerçekleşmeyen alanı temsil eder. Japon mum grafiğinde, pencere boşluk olarak yorumlanır. Yükseliş eğiliminde, bir günün en yüksek fiyatı ertesi günün en düşük fiyatından daha düşük olduğunda bir boşluk oluşur. Dolayısıyla düşüş eğiliminde, herhangi bir günün en düşük fiyatı ertesi günün en yüksek fiyatından daha yüksek olduğunda bir boşluk oluşur. Hareket başlangıcından önce fark edildiklerinde boşluklar önemli bir rol oynayabilir.

Boşluk türleri

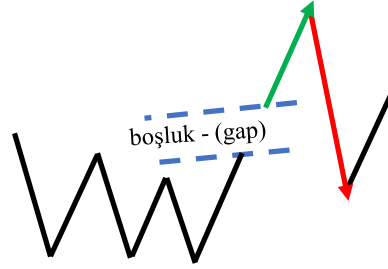
Bir hisse senedinin temettüsüz satışı sonucunda oluşan boşluk haricinde dört farklı boşluk türü vardır. Her boşluk türünün kendine özgü bir anlamı olduğu için, bu tür boşlukları ayırt edebilmek çok önemlidir.

- *Kopuş boşluğu*: Fiyatlar birikme alanından ayrıldığında ortaya çıkar. Fiyat bir üçgenden (yükselen veya alçalan üçgen) boşlukla ayrıldığında, duyarlılıktaki değişimin güçlü olduğu ve gelecek hareketin keskin olacağı söylenebilir. Hacme dikkat edilmelidir. Boşluk oluşuktan sonra hacim yüksekse, piyasanın *boşluğu doldurmak* için geri dönmemesi çok muhtemeldir. Fiyat düşük bir hacimde ayrılırsa, fiyatlar eğilimlerine kaldıkları yerden devam etmeden önce boşluğun doldurulması olasılığı vardır [68].



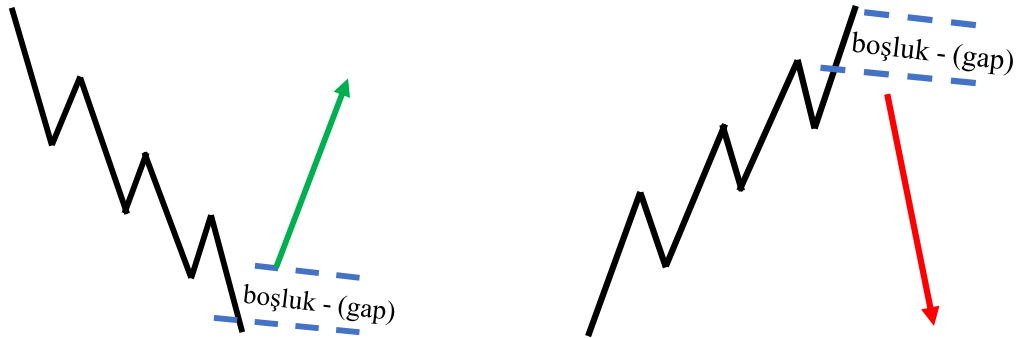
Şekil 2.27. Kopuş Boşluğu formasyon görünüm.

• *Standart boşluk*: alan boşluğu, desen boşluğu veya geçici boşluk olarak da bilinir. Bu boşluklar, işlemler kısa bir süre zarfında destek ve direnç seviyeleri arasında sınırlandırıldığında ve piyasa fiyatı yatay hareket ederken ortaya çıkma eğilimindedir. Bunları fiyat birikmesi alanında görmek de mümkündür. Genellikle, ilerleyen günlerde fiyat boşlukları doldurmak için geri hareket eder veya yükselir. Boşluk doldurulursa, o zaman pek fazla öngörülse değerleri kalmaz [69].



Şekil 2.28. Standart Boşluk formasyon görünüm.

• *Bitkinlik boşluğu* hareketin bittiğini bildirir. Bu boşluklar hızlı, düz çizgili yükseliş veya düşüş ile ilişkilendirilir. Yüksek hacimle tepede oluştuğlarında, piyasanın tükenmiş ve dönüşün yolda olma ihtimali çok yüksektir [70].



Şekil 2.29. Bitkinlik Boşluğu formasyon görünüm.

Mum Grafik Desenleri

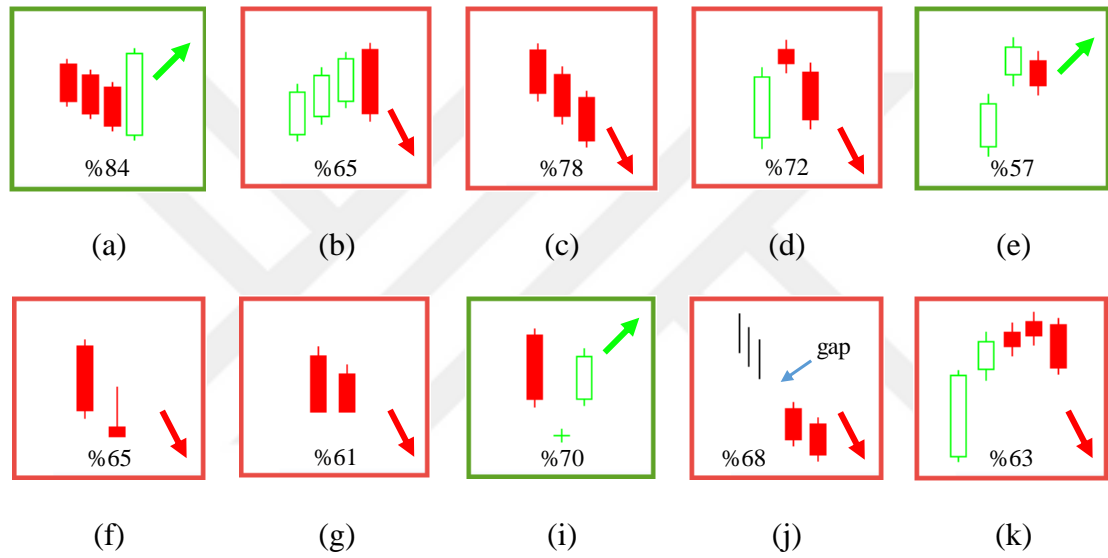
Hisse senedi analizlerinde en sık kullanılan yöntemlerden biri mum grafikleri ile teknik analizdir. Bu analiz yöntemi on yedinci yüzyılda pirinç ticaretinde kullanılmak üzere Japonlar tarafından geliştirilmiştir.

Ünlü yatırımcı Thomas Bulkowski mum grafiklerinde test ettiği fiyat yönünün ve deviniminin habercisi olarak son derece iyi performans gösteren birçok mum deseni vardır. Her biri, daha yüksek veya daha düşük fiyatları tahmin ederken çevresindeki fiyat çubukları bağlamında işlerlik gösterir.

Bu analiz, 2008 tarihli “Mum Grafik Ansiklopedisi” adlı kitabında mum desenleri için performans sıralaması yapan Thomas Bulkowski'nin çalışmalarına dayanmaktadır. Bulkowski beklenen iki tip desen sonucu için istatistik sunar:

- Dönüş - Mum dönüş desenleri fiyatın yönündeki değişikliği öngörürken,
- Devam - Devam desenleri mevcut fiyat yönündeki süreyi öngörür.

Analizlerde grafiklerin yukarı yönlü eğilimleri boğa, aşağı yönlü eğilimleri ise ayı formasyonları olarak isimlendirilirler. Bu bağlamda belirtilen desenlere ait bazı formasyon örnekleri yazının devamında sunulmuştur [42].



Şekil 2.30. Mum formasyon görünümleri.

Boğa formasyonlarından *düşen blok dönüş deseni*, Şekil 2.30.a'da olduğu gibi aşağı yönlü üç kırmızı mum ortaya çıkarır. Her çubuk yeni bir alçak dibi gösterir ve çubuk içindeki dibe yakın bir noktada kapanır. Dördüncü çubuk daha da düşük seviyede açılır, ancak serideki ilk mumun tepesinin üzerinde kapanan geniş aralıklı bir dış çubukta terse döner. Açılış fiyatı aynı zamanda dördüncü çubuğun dip noktasını işaret eder. Bulkowski'ye göre, bu dönüş daha yüksek fiyatları %84 doğruluk oranıyla öngörmektedir [71].

Ayı formasyonlarından *boşluk devam deseni*, Şekil 2.30.j'de görüldüğü gibi bir yükseliş eğiliminde göze çarpan bir tepeden sonra, yeni alçak dipleri belirten aşağı yönlü çubuklar ardından ortaya çıkartan bir boşluk meydana getirir. Bu desen, düşüşün yeni alçak diplere kadar bile devam edeceğini ve belki de daha geniş çaplı bir düşüşü

tetikleyeceğini öngörür. Bulkowski'ye göre, bu desen daha düşük fiyatları %68 doğruluk oranıyla öngörmektedir [72].

Ayı formasyonlarından *üç siyah karga dönüş deseni*, Şekil 2.30.c'deki gibi bir yükseliş eğiliminin tam tepesinde veya tepenin yakınında başlar; üç kırmızı çubuk, çubuk içi diplerin yakınında kapanan yeni alçak diplere işaret eder. Bu desen, düşüşün yeni alçak diplere kadar bile devam edeceğini ve belki de daha geniş çaplı bir düşüşü tetikleyeceğini öngörür. Ayı formasyonunu en iyi yansıtan versiyon yeni bir tepede başlar, çünkü hız oyunlarına girişerek alıcıları tuzağa düşürür. Bulkowski'ye göre, bu desen daha düşük fiyatları %78 doğruluk oranıyla öngörmektedir [73].

Ayı formasyonlarından *akşam yıldız dönüş deseni*, Şekil 2.30.d'de olduğu gibi bir yükseliş eğilimini yeni bir tepeye taşıyan uzun, beyaz bir çubukla başlar. Piyasa bir sonraki çubukta daha fazla boşluk açar, ancak ortalıkta yeni alıcılar belirmez ve dar aralıklı bir mum ortaya çıkar. Desen üçüncü çubuğun alt kısmındaki bir boşlukla tamamlanır; bu boşluk düşüşün yeni alçak diplere kadar bile devam edeceğini ve belki daha geniş çaplı bir düşüş eğilimini tetikleyeceğini öngörür. Bulkowski'ye göre, bu desen daha düşük fiyatları %72 doğruluk oranıyla öngörmektedir [74].

Boğa formasyonlarından *terk edilmiş bebek dönüş deseni*, Şekil 2.30.i'de görüldüğü gibi düşüş eğiliminin dibinde ortaya çıkar. Bir dizi siyah mum yeni alçak dipleri belirttikten sonra düşüş eğiliminin en altında görünür. Piyasa bir sonraki çubukta daha düşük seviyelerde boşluk oluşturur, ancak ortalıkta yeni satıcılar belirmez, böylece açılış ve kapanış fiyatları aynı olan, dar aralıklı doji mumu ortaya çıkar. Desen üçüncü çubuktaki yükseliş boşluğu ile tamamlanır; bu boşluk, toparlanmanın yeni yüksek tepelere kadar bile devam edeceğini ve belki de daha geniş çaplı bir yükseliş eğilimini tetikleyeceğini öngörür. Bulkowski'ye göre, bu desen daha yüksek fiyatları %70 doğruluk oranıyla öngörmektedir [75].

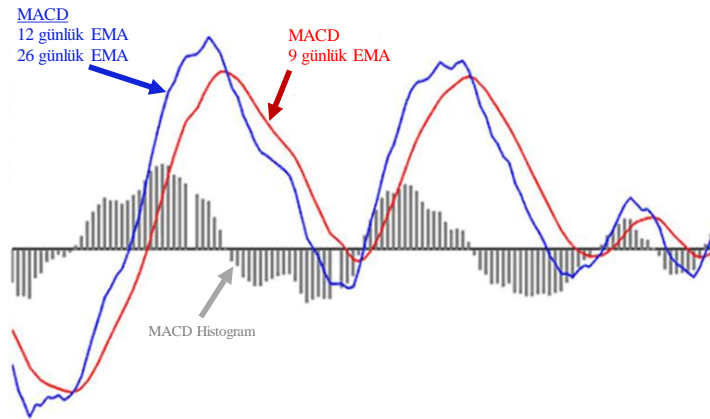
2.5.2.2. Teknik Göstergeler (Indicators)

MACD (Moving Average Convergence/Divergence - Hareketli Ortalama Yaklaşma/Uzaklaşma göstergesi), Gerald Appel tarafından 1970'lerin sonunda oluşturulan bir teknik analiz göstergesidir. Bir hisse senedinin fiyatındaki eğilimin gücü, yönü, hızı ve süresindeki değişiklikleri tespit etmek için kullanılır. MACD, kapanış fiyatlarının iki üssel hareketli ortalaması (EMAs - Exponential Moving Averages) arasındaki farkın hesaplanmasıdır [76].

Şekil 2.31'deki grafik, MACD göstergesi olan bir hisse senedini göstermektedir. Gösterge bir mavi çizgi, bir kırmızı bir çizgi ve iki çizgi arasındaki farkı hesaplayan bir histogram veya çubuk grafikten oluşmaktadır. Değerler, grafiğin ana bölümündeki hisse senedi fiyatı üzerinden hesaplanır. Şekildeki örnek için bu şu anlama gelir:

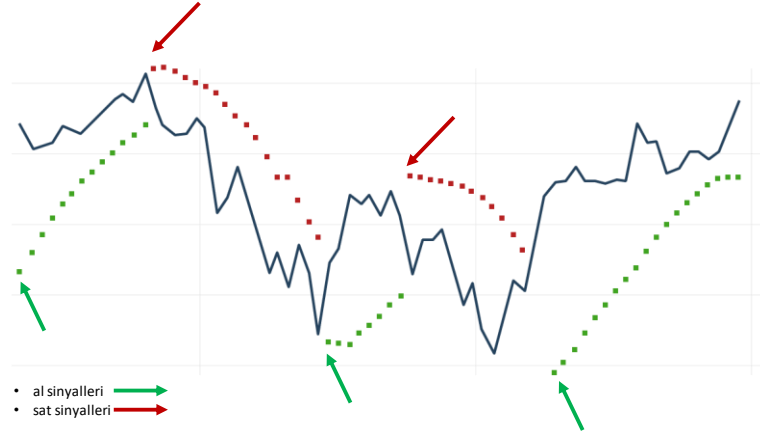
- MACD çizgisi (mavi çizgi): 12 ve 26 günlük EMA'lar arasındaki fark,
- Sinyal (kırmızı çizgi): mavi çizginin 9 günlük EMA'sı,
- Histogram (çubuk grafik): mavi ve kırmızı çizgiler arasındaki farkı göstermektedir.

Hareketli ortalama, kısa dönemli dalgalanmaları gidermek ve uzun dönemli eğilimleri veya döngüleri vurgulamak için çoğunlukla zaman dizisi verileriyle birlikte kullanılır. Kısa dönem ve uzun dönem arasındaki eşik uygulamaya bağlıdır ve hareketli ortalamanın parametreleri de buna göre belirlenir [42], [65].



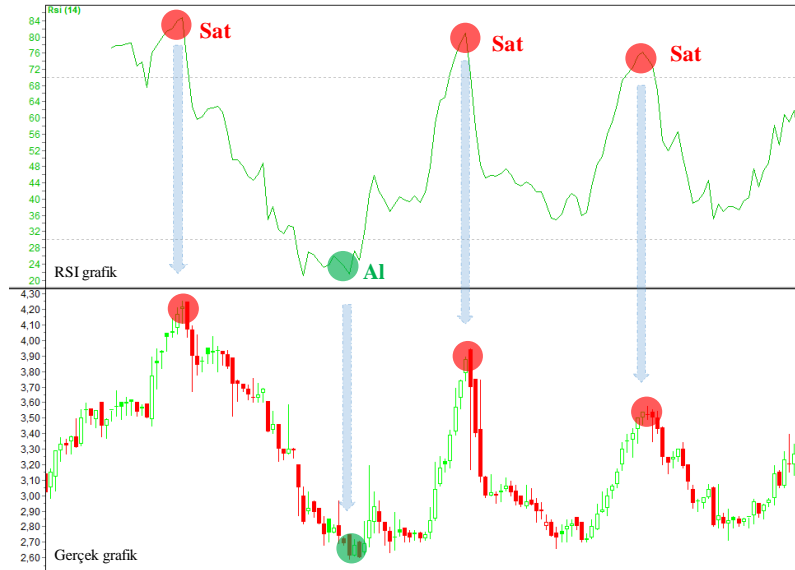
Şekil 2.31. MACD indikatör görünüm.

Parabolik SAR, Parabolik Bitiş ve Dönüş (SAR - Stop and Reverse) göstergesi, fiyat ve zaman bileşenlerini potansiyel alış ve satış sinyalleri üretmek amacıyla birleştirir. Parabolik SAR, dur emirlerinin ne zaman verileceğini belirlemede etkili bir araç olarak bilinmektedir. Şekil 2.32'deki örnek grafik gösterim, Parabolik Bitiş ve Dönüş (SAR) teknik göstergesi tarafından üretilen olası alım ve satım sinyallerini gösteren iyi bir örnektir [77].



Şekil 2.32. Parabolik SAR indikatör görünüm.

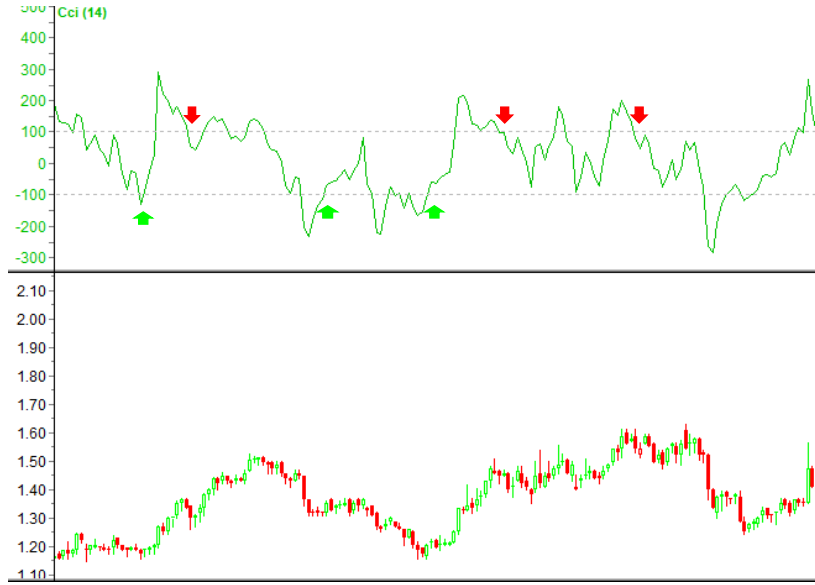
Göreceli Güç Endeksi (RSI - Relative Strength Index), hız (sürat) yanı sıra yönsel fiyat hareketlerinin değişimini (büyüklüğü) ölçmek için kullanılan oldukça ustalıkla bir momentum tabanlı osilatördür. RSI, grafiksel olarak gösterildiğinde, belirli bir piyasanın şimdiki ve tarihsel güç ve zayıflığını izlemek için görsel bir vasıta sağlar. Kapanış fiyatına dayanan güç veya zayıflık, belirtilen bir işlem dönemi boyunca fiyat ve momentumdaki değişiklikleri ölçen güvenilir bir metriktir. Enstrümanların (hisse senedi endeksleri) ve kaldıraçlı finansal ürünlerin (tüm türev ürünlerin) popüleritesi göz önüne alındığında; RSI, Şekil 2.33’de de görüldüğü üzere fiyat hareketlerine uygulanabilir bir gösterge olduğu kanıtlanmıştır [78].



Şekil 2.33. RSI indikatör görünüm.

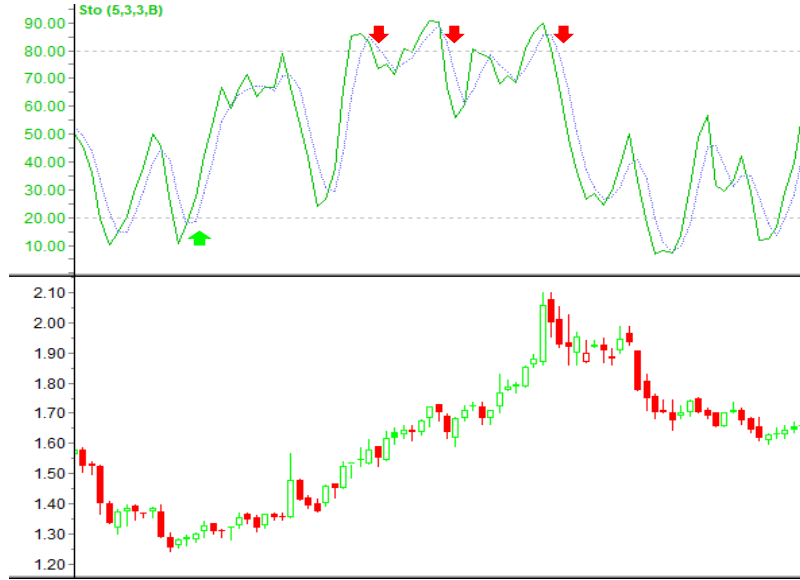
Emtia Kanal İndeksi (CCI - Commodity Channel Index), teknik analizde kullanılan ve bir enstrümanın istatistiksel ortalamasından gelen varyasyonları ölçen bir momentum

osilatördür. CCI, çok yönlülüğü nedeniyle hiç de az sayılamayacak bir oranda popülerlik kazanmış, çok iyi bilinen ve yaygın olarak kullanılan bir göstergedir. Tanımlı bir sınırı olmayan bir osilatördür, yani alt veya üst sınırları yoktur. CCI genellikle sapmaların yanı sıra dönüşleri bulmak için de kullanılır. Gösterge, başlangıçta emtiaların eğilimlerini belirlemek için kullanılmak üzere tasarlanmıştır ama şimdi çok çeşitli finansal araçlarda kullanılmaktadır. Şekil 2.34'te örnek bir hisse senedi üzerinde görünümü mevcuttur [79].



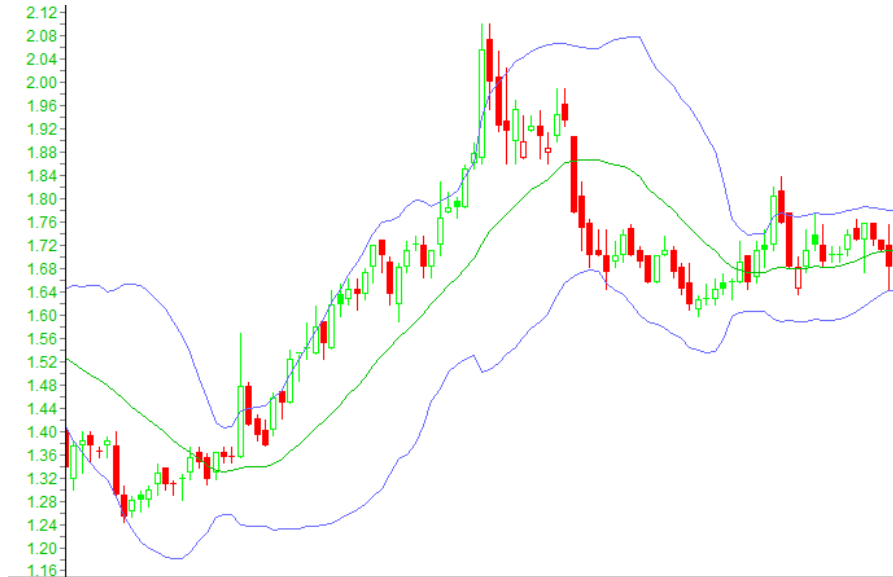
Şekil 2.34. CCI indikatör görünümü.

Stokastik Osilatör (Stochastic Oscillator), Şekil 2.35'te görüldüğü gibi menzili sınırlı bir momentum osilatördür. Stokastik göstergesi, kullanıcı tanımlı bir periyot üzerinden kapanış değeri konumunu yüksek/düşük aralığı ile karşılaştırmak için tasarlanmıştır. Genellikle, Stokastik Osilatör üç şey için kullanılır: Aşırı alım ve aşırı satım düzeylerini belirleme, farklılaşmaları tespit etme ve boğa/ayı oluşumları veya sinyallerini belirleme [80], [81], [82].



Şekil 2.35. SO indikatör görünüm.

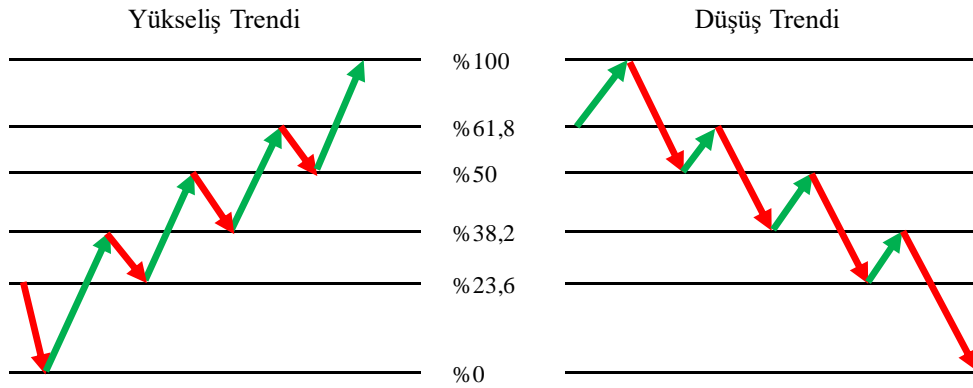
Bollinger Bantları (BB - Bollinger Bands), 1980'lerin başında John Bollinger tarafından oluşturulan yaygın bir teknik analiz aracıdır. Şekil 2.36'daki Bollinger Bantları, güvenlik fiyatlarına göre çizilen üç satırlık bir banttandır. Ortadaki çizgi genellikle 20 günlük bir süreye ayarlanmış Basit Hareketli Ortalamadır (SMA) (eğilim çizgisi ve dönem yatırımcı tarafından değiştirilebilir; ancak 20 günlük hareketli ortalama en popüler olanıdır). Daha sonra SMA, Bantlar ve fiyat arasındaki ilişkiyi gözlemleyerek oynaklığı ölçmenin bir yolu olarak kullanılan Üst ve Alt Bantlar için bir temel görev görür. Tipik olarak Üst ve Alt Bantlar SMA (Orta Bant) 'dan alınan iki standart sapmaya göre ayarlanır; ancak standart sapma sayısı yatırımcı tarafından da ayarlanabilir [83].



Şekil 2.36. Bollinger Bands indikatör görünüm.

2.5.2.3. Fibonacci Düzeltmeleri

Fibonacci düzeltmeleri, destek ve direnç seviyelerini belirlemeye yarayan bir teknik analiz yöntemidir. Fibonacci sayı dizisini kullanmalarının ardından bu şekilde adlandırılmışlardır. Fibonacci düzeltmesi, piyasaların bir hareketin öngörülebilir bir kısmını geri çekeceği ve daha sonra orijinal yönünde ilerlemeye devam edeceği fikrine dayanır. Fibonacci düzeltmesi, bir grafik üzerindeki iki uç nokta alınıp dikey mesafede anahtar Fibonacci oranlarına bölünerek oluşturulur. %0,0 düzeltmenin başlangıcı olarak kabul edilirken, %100,0 hareketin orijinal kısmına tam bir geri dönüştür. Bu seviyeler bir kez belirlendikten sonra, Şekil 2.37'deki olası destek ve direnç seviyelerini belirlemek için yatay çizgiler çizilir ve kullanılır [84].



Şekil 2.37. Fibonacci düzeltme oranları.

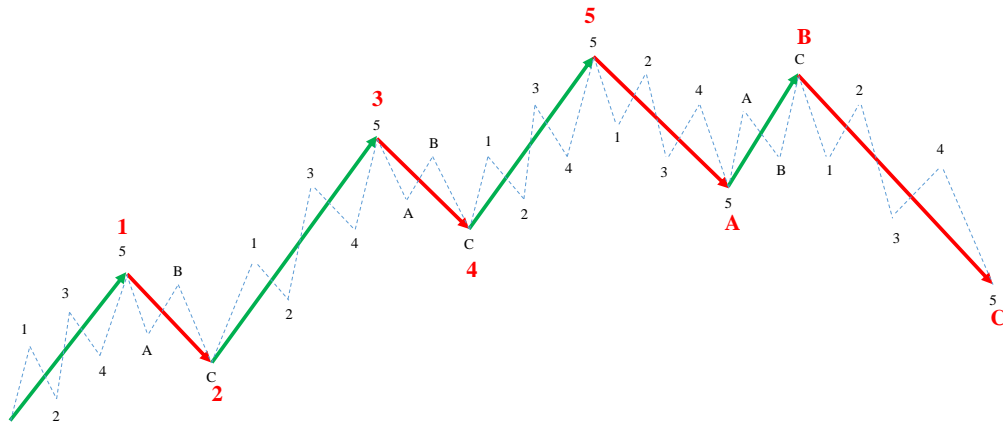
Fibonacci oranları, Fibonacci sayı dizisinden türetilen ve oranlar olarak ifade edilen matematiksel ilişkilerdir. Anahtar Fibonacci oranları %0, %23,6, %38,2, %50, %61,8 ve %100'dür.

2.5.2.4. Elliott Dalga Teorisi

Elliott Dalga Prensibi, yatırımcıların piyasa döngülerini analiz etmek ve yatırımcı psikolojisindeki aşırı uçları, fiyatlardaki dip ve tepeleri ve diğer kolektif faktörleri tespit etmek suretiyle piyasa eğilimlerini öngörmek için kullandıkları bir teknik analiz yöntemidir. Profesyonel hesap uzmanı Ralph Nelson Elliott (1871–1948), 1930'larda altta yatan sosyal ilkeleri keşfetmiş ve analitik araçları geliştirmiştir. Piyasa fiyatlarının, uygulayıcıların bugün Elliott desenleri ya da kısaca dalgalar dedikleri belirli örüntülerle ortaya çıktığını ileri sürmüştür [84].

Elliott'un modelinde, piyasa fiyatları *dürtüsel* veya *güdüsel* faz ile eğilimin tüm zaman ölçeklerindeki düzeltici faz arasında gidip gelir. Dürtüler daima, yine güdüsel ve düzeltici karakter arasında gidip gelen 5 adet düşük seviyeli dalgaya bölünmüştür, böylece 1, 3 ve 5. dalgalar dürtü, 2 ve 4. dalgalar ise 1. ve 3. dalgaların daha küçük düzeltmeleridir [85].

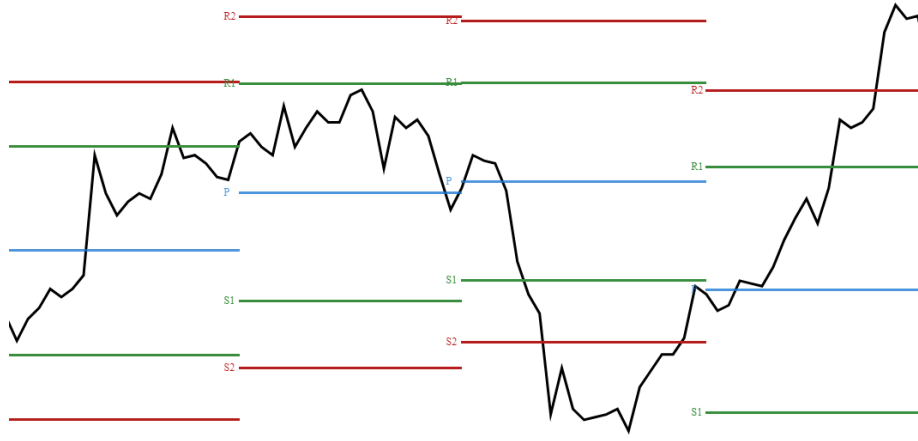
Bir finansal piyasadaki her bir deseninin ismi vardır. Şekil 2.38'de görüleceği üzere, her bir dalga için hem işlevi hem de dereceyi belirten semboller kullanılırken, motivasyon dalgaları için sayılar, düzeltici dalgalar için harfler (idealleştirilmiş üç dalga yapısı serisinin veya derecesinin en üstünde gösterilir) kullanılır [86].



Şekil 2.38. Elliott dalgaları.

2.5.2.5. Pivot Noktası

Pivot noktası, yatırımcılar tarafından piyasa hareketini öngörmek için kullanılan, bir piyasanın teknik analizindeki fiyat anlamlılık düzeyidir. Bir pivot noktası, bir önceki işlem döneminde bir piyasanın performansından çıkarılan anlamlı fiyatların (yüksek, düşük, yakın) ortalaması olarak hesaplanır. Sonraki dönemde piyasa pivot noktasının üzerinde işlem görürse, bu genellikle boğa piyasası duyarlılığı olarak değerlendirilirken, pivot noktasının altında işlem ayı piyasası olarak görülür. Piyasanın önceki işlem aralıklarından hesaplanan fiyat farklarını çıkararak veya ekleyerek, pivot noktasının altındaki ve üstündeki ek destek ve direnç seviyelerini hesaplanabilir. Şekil 2.39'da örnek gösterim mevcuttur [87].



Şekil 2.39. Pivot noktaları.

3. DERİN ÖĞRENME ALGORİTMALARI

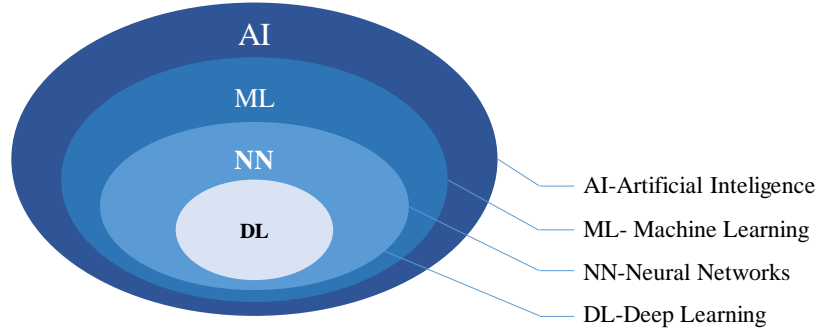
3.1. DERİN ÖĞRENME

1950'lerden bu yana, genellikle Makine Öğrenmesi (ML - Machine Learning) olarak adlandırılan küçük bir Yapay Zekâ (AI - Artificial Intelligence) alt kümesi, geçtiğimiz birkaç on yıl içinde pek çok alanda devrim gerçekleştirmiştir. Sinir ağları (NN - Neural Networks), ML'nin bir alt alanıdır ve bu alt alan DL'yi ortaya çıkarmıştır. Şekil 3.1 AI'nın taksonomisini göstermektedir. Öğrenmenin derin mimarisini veya hiyerarşik öğrenme yaklaşımlarını kullanan DL, 2006'dan itibaren büyük oranda geliştirilen ML'nin bir kategorisidir. Öğrenme, öğrenilen modelin (algoritmanın) belirli bir görevi yerine getirebilmesi için model parametrelerinin tahmin edilmesini kapsayan bir süreçtir. Örneğin, yapay sinir ağlarında (ANN-Artificial Neural Network) parametreler ağırlık matrisleridir. Öte yandan DL, girdi ve çıktı katmanı arasındaki, hiyerarşik mimarilere sahip doğrusal olmayan bilgi işleme birimlerinin, gelecekteki öğrenme ve desen sınıflandırması için kullanılan pek çok aşamasının var olmasına izin veren çeşitli katmanlardan oluşur [88], [89]. Verilerin gösterimi temelli öğrenme yöntemleri temsil öğrenmesi olarak da tanımlanabilir [90]. Yakın zamanlı literatür, DL tabanlı temsil öğrenmesinin, yüksek seviyeli kavramların düşük seviyeli kavramlar ile, düşük seviyeli kavramların ise yüksek seviyeli kavramlar ile tanımlanabildiği bir özellikler veya kavramlar hiyerarşisi içerdiğini belirtmektedir. Bazı makalelerde DL, farklı uygulama alanlarındaki neredeyse her türlü sorunu çözebilen evrensel bir öğrenme yaklaşımı olarak tanımlanmıştır [91], [92].

3.1.1. DERİN ÖĞRENME YAKLAŞIMLARI

Derin modeller derin yapılara sahip sinir ağları olarak adlandırılabilir. Şekil 3.1'de yapay zekâ kavramındaki konumu belirtilmiştir. Sinir ağlarının geçmişi 1940'lara kadar uzanır [93] ve geliştirilmelerindeki asıl amaç, genel öğrenme sorunlarını sistemli bir şekilde çözmek için insan beyninin işleyişini taklit etmektir. 1980'lerde ve 1990'larda, Hinton ve ark. tarafından geriye yayılım algoritmasının önerilmesiyle popülerlik kazanmıştır [94]. Bununla birlikte, eğitimin aşırı uyum göstermesi, büyük ölçekli eğitim verilerinin eksikliği, sınırlı ölçümlene gücü ve diğer makine öğrenmesi araçlarına

kıyasla düşük performans nedeniyle, sinir ağıları 2000'li yılların başlarında gözden düşmüştür.



Şekil 3.1. AI taksonomisi.

Derin öğrenme, ses tanıma teknolojisinde çığır açmasıyla birlikte, 2006'dan [95] itibaren popüler hale gelmiştir [96]. Derin öğrenmenin canlanması aşağıdaki faktörlere bağlanabilir.

- Çok büyük öğrenme kapasitesini tam olarak ortaya koyan, ImageNet [97] gibi büyük ölçekli açıklamalı eğitim verilerinin ortaya çıkması;
- GPU kümeleri gibi yüksek performanslı paralel hesaplama sistemlerinin hızlı gelişimi;
- Ağ yapılarının tasarımında ve eğitim stratejilerinde önemli ilerlemeler.

Otokodlayıcı (AE- Auto Encoder) [98] veya Sınırlı Boltzmann Makinesi (RBM- Restricted Boltzmann Machine) [99] tarafından yönlendirilen denetimsiz ve katmansal ön eğitim sayesinde iyi bir başlangıç yapılmıştır. Bırakma ve veri çoğaltma ile eğitimdeki aşırı uyum gösterme sorunu hafifletilmiştir [8], [25]. Parti normalizasyonu ile çok derin sinir ağlarının eğitimi oldukça verimli hale gelir [100]. Bu arada, AlexNet [8], Overfeat [101], GoogLeNet [102], VGG [103] ve ResNet [104] gibi çeşitli ağ yapıları, performansı artırmak için kapsamlı bir şekilde incelenmiştir. Derin öğrenmenin başarısı, büyük bir konvolüsyon sinir ağlarının (CNN-Convolutional Neural Network)'nin birkaç teknik kullanılarak [8] 1.2 milyon etiketli görüntü için eğitilmesinden kaynaklanmaktadır.

3.2. KONVÜLSİYON SİNİR AĞLARI (CNN)

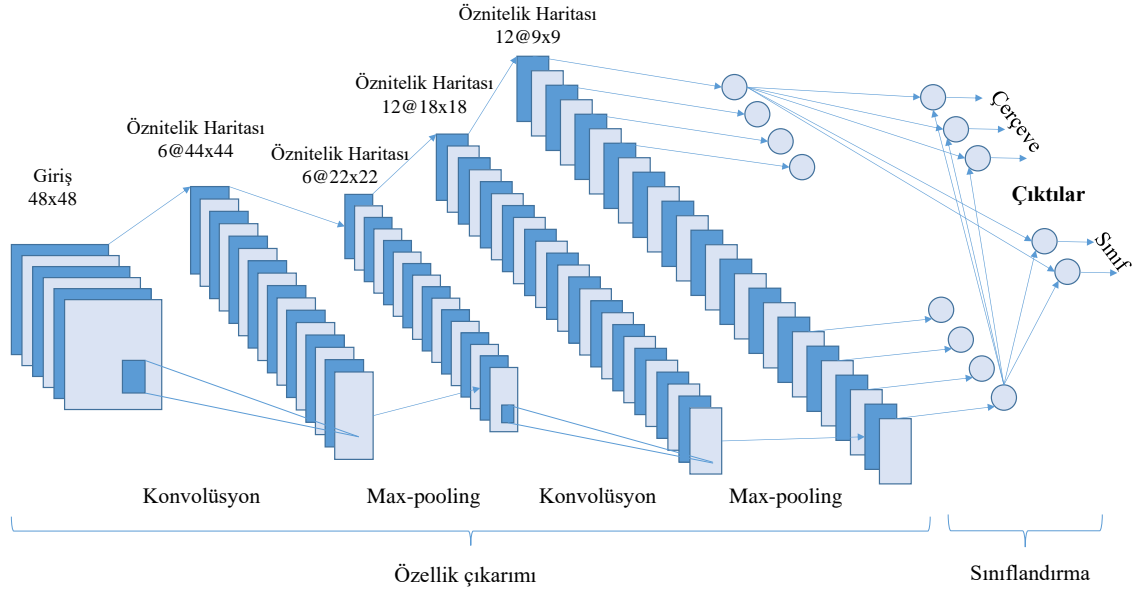
Görüntüler gibi ızgara yapılı girdi verileri üzerinde uzmanlaşan sinir ağı tiplerinden biri CNN'lerdir. Pratik uygulamalarda son derece başarılı oldukları kanıtlanmıştır. Adından

da anlaşılacağı gibi, katmanlarından en az bir tanesinde, genel matris çarpımı yerine konvolüsyon adı verilen matematiksel işlem kullanılmaktadır [105].

3.2.1. CNN'nin İşleyişi

CNN'ler normal sinir ağlarına çok benzer, ancak nöronları üç boyutlu olarak düzenlenir (genişlik, yükseklik ve derinlik). Bir katmanın içindeki bir nöron, kendisinden önce gelen katmanın, alıcı alan adı verilen, yalnızca küçük bir bölgesine bağlıdır; normal bir sinir ağında olduğu gibi tamamına bağlı değildir. CNN mimarisi, pek çok farklı tipte ardışık katmandan oluşur; bu katmanlardan bazıları tekrarlanır .

Şekil 3.2, öznelik çıkarıcılar ve sınıflandırıcı olmak üzere iki ana kısımdan oluşan CNN genel mimarisini göstermektedir. Öznelik çıkarıcı katmanlarda, ağın her bir katmanı, girdi olarak kendisinden hemen önce gelen katmandan gelen çıktıyı alır ve kendi çıktısını girdi olarak kendisinden sonra gelen katmana aktarır. CNN mimarisi, üç tip katmanın birleşiminden oluşur: Konvolüsyon, maksimum havuzlama ve sınıflandırma. Ağın düşük ve orta seviyesinde iki tip katmanı vardır: Konvolüsyon katmanları ve maksimum havuzlama katmanları. Çift sayılı katmanlar konvolüsyon için, tek sayılı katmanlar ise maksimum havuzlama işlemleri içindir. Konvolüsyon ve maksimum havuzlama katmanlarının çıkış düğümleri, öznelik haritalaması adı verilen iki boyutlu (2D) bir düzlemde gruplanır. Bir katmanın her bir düzlemi, genellikle önceki katmanların bir veya daha fazla düzleminin birleşiminden türetilir. Bir düzlemin düğümleri, bir önceki katmanın birbiri ile bağlantılı her bir düzleminin küçük bir bölgesine bağlıdır. Konvolüsyon katmanının her bir düğümü, girdi düğümlerindeki konvolüsyon işlemleri sayesinde girdi görüntülerinden öznelik çıkarır.



Şekil 3.2. Konvolüsyon sinir ağının (CNN) genel mimarisi [92].

Üst seviye özellikler, alt seviye katmanlardan nakledilen özniteliklerden türetilir. Öznitelikler en yüksek katmana veya seviyeye ilerledikçe, özniteliklerin boyutları sırasıyla konvolüsyon ve maksimum havuzlama işlemleri için çekirdeğin boyutuna bağlı olarak azalır. Bununla birlikte, sınıflandırma doğruluğunu sağlamak amacıyla girdi görüntülerinin daha iyi özniteliklerini temsil etmek için, öznitelik haritalarının sayısı genellikle artar. CNN'nin son katmanının çıktısı, sınıflandırma katmanı adı verilen, tam bağlantılı bir ağın girdisi olarak kullanılır. Beslemeli sinir ağları, daha iyi performansa sahip oldukları için sınıflandırma katmanı olarak kullanılmıştır [106], [107]. Sınıflandırma katmanında, çıkarılan öznitelikler son sinir ağının ağırlık matrisinin boyutuna göre girdi olarak alınır. Ancak, tam bağlantılı katmanlar ağ veya öğrenme parametreleri açısından maliyetlidir. Günümüzde, tam bağlantılı ağlara alternatif olarak kullanılan, ortalama havuzlama ve global ortalama havuzlama gibi çeşitli yeni teknikler bulunmaktadır. İlgili sınıfın skoru, bir softmax katmanı kullanılarak en üstteki sınıflandırma katmanında hesaplanır. En yüksek skora göre, sınıflandırıcı ilgili sınıflar için çıktı verir [92].

3.2.1.1. Konvolüsyon Katmanı

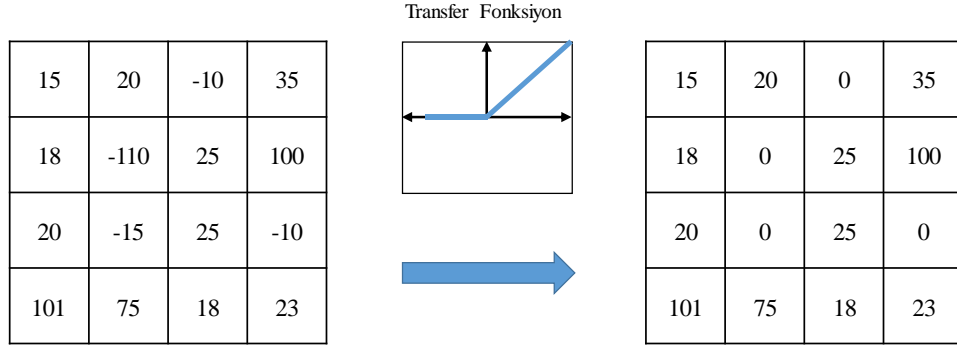
Adından da anlaşılacağı gibi, bu CNN'nin temel yapı taşıdır. Görüntünün genişlik ve yükseklik boyutlarında konvolse edilmiş bir dizi filtreden oluşur. Görüntünün konvolse edilmiş halindeki filtreler, görüntü ile aynı sayıda boyuta sahiptir; her bir boyutun derinliği aynıdır, fakat genişliği ve yüksekliği daha küçüktür. Yaygın olarak kullanılan

uzamsal boyutlar, 3x3 veya 5x5'tir. Çıktı genişliği ve yüksekliği filtrenin boyutuna, adım aralığına (filtrenin her bir hesaplama arasında hareket eden piksel sayısı, genellikle bir veya ikidir) ve görüntünün etrafındaki sıfır-doldurma miktarına bağlıdır. Çıktı derinliği uygulanan filtre sayısı ile aynı olacaktır. Konvolüsyon süreci, bir makine öğrenmesi sisteminin geliştirilmesine yardımcı olabilecek üç fikri destekler: seyrek etkileşimler, parametre paylaşımı ve eş değişkenli temsil [105]. Ayrıca, onu bir dereceye kadar kaymalara, ölçekleme ve bozulmalara karşı sabit kılar [108]. Bir girdinin ve bir filtrenin konvolüsyondan meydana gelen çıktıya öznitelik haritası veya bazen de aktivasyon haritası denir. Katmandaki her filtre tarafından bir öznitelik haritası oluşturulur ve bunlar birlikte çıktı derinliğini oluştururlar. Her öznitelik haritasının uzamsal boyutu girdi görüntüsünün boyutuna, doldurmaya, filtre boyutuna ve adım aralığına bağlıdır. Filtrenin girdiden daha küçük olması seyrek etkileşimlere yol açar. Bir öznitelik haritasındaki her birimin, girdinin alıcı alan adı verilen $n \times n$ bölgesine n^2 bağlantıları vardır. Örneğin, görüntü işlemede bu köşeler gibi küçük, anlamlı özniteliklerin saptanabileceği ve daha az parametrenin saklanmasına gerek olduğu anlamına gelir [105]. Öznitelik haritasındaki her birimin n^2 eğitilebilir ağırlıkları ve eğitilebilir bir sapması vardır. Bir öznitelik haritasındaki tüm birimler aynı parametreleri paylaşır, bu şu şekilde de yorumlanabilir; adından da anlaşılacağı gibi, bir öznitelik haritası yatay veya dikey köşeler gibi farklı öznitelikler tespit ederse, bu onu girdinin köşelerin tespit edildiği bölgesinden bağımsız kılar. Önemli olan bu özniteliklerin göreceli konumlarıdır. Bu parametre paylaşımı önemli miktarda bellek tasarrufu sağlar [108].

3.2.1.2. Doğrultulmuş Doğrusal Birimler Katmanı (ReLU-Rectified linear units layer)

Element bazında satüre olmayan etkinleştirme fonksiyonunu uygulayarak doğrusalsızlığı artırır $f(x) = \text{maksimum}(0, x)$. Bunun nasıl işlediğinin bir örneği Şekil 3.3'de görülebilir. Ağin, bu satüre etmeyen fonksiyonu kullanarak, $f(x) = \tanh(x)$ gibi satürasyon fonksiyonlarını veya $f(x) = \frac{1}{1+e^{-x}}$ sigmoid fonksiyonunu kullandığı duruma kıyasla çok daha hızlı eğitebildiği ortaya konulmuştur. [8]. Uzamsal boyut değişmeden kalır. Küçük, sıfır olmayan bir *gradyan*, α , denklem (3.1)'de olduğu gibi negatif sayılar için de kullanılabilir.

$$f = \begin{cases} x, & x > 0 \\ \alpha x, & \text{otherwise} \end{cases} \quad (3.1)$$

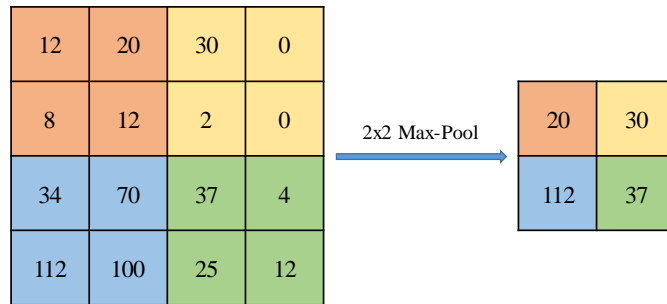


Şekil 3.3. Doğrultulmuş doğrusal birimler katmanının nasıl çalıştığına örnek.

Sol matristeki tüm eksi değerli sayılar, doğrultucu fonksiyonu uygulandıktan sonra sıfıra ayarlanır ve tüm diğer değerler değiştirilmeden bırakılarak yeni matris oluşturulur [109].

3.2.1.3. Havuzlama Katmanı

Küçük filtreler kullanılarak, bir önceki katmandan gelen çıktının dikdörtgen bir alanındaki maksimum veya ortalama değerleri örneklemek için hacmin doğrusal olmayan alt örneklemesinin oluşturulması işlemidir. Havuzlama, parametrelerin ve hesaplamaların miktarını azaltmak için uzamsal boyutu azaltır ve ayrıca aşırı uyum göstermeyi önler. Havuzlama katmanlarının nasıl işlediği Şekil 3.4'te gösterilmektedir. Havuzlama yöntemi olarak; maksimum havuzlama, ortalama havuzlama, toplam havuzlama yöntemleri kullanılmaktadır.



Şekil 3.4. Maksimum havuzlama örneği [110].

3.2.1.4. Tam Bağlantılı Katman

Bu katmandaki nöronlar, normal sinir ağlarında olduğu gibi, önceki katmanlardaki tüm aktivasyonlara tamamen bağlıdır. Bunlar genellikle ağın sonundadır ve sınıf olasılıklarını elde etmek için kullanılırlar [111].

3.2.1.5. Kayıp Katmanı

Genellikle ağdaki, örneğin *softmax* [109] fonksiyonunu uygulayarak, sınıflandırma gibi görevin amacını hesaplayan son katmandır. Bu hesaplama ait formül denklem (3.2)'de görülmektedir.

$$\sigma(z) = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ for } j = 1, \dots, K \quad (3.2)$$

Bahsedilen katmanların birleşimi bir CNN oluşturmak için kullanılabilir. Denklem (3.3)'te bir CNN'nin tipik mimari desen formülizasyonu görülmektedir [111].

$$\text{Input} \rightarrow [[\text{Conv} \rightarrow \text{ReLU}] * N \rightarrow \text{Pool?}] * M \rightarrow [\text{FC} \rightarrow \text{ReLU}] * K \rightarrow \text{FC} \quad (3.3)$$

“ \rightarrow ” göstergesi tekrarı temsil ederken, N , M , ve K sıfırdan büyük tam sayıları temsil etmektedir. N genellikle üçten küçük veya üçe eşittir ve K ise kesinlikle üçten küçük bir sayıyı temsil etmektedir. *Pool?* havuzlama katmanının opsiyonel olduğunu belirtir. Daha büyük ve daha derin ağlar için, havuzlama katmanından önce birden fazla konvolüsyon katmanı yığılmak genellikle iyi bir fikirdir. Çünkü konvolüsyon katmanı yıkıcı havuzlama işleminden önce girdi hacminin daha karmaşık özneliklerini tespit edebilir [111].

Tam bağlantılı katmanlarda eğitim sırasında bırakma uygulaması yaygındır. Bırakma oranı, aşırı uyum göstermeyi azaltmanın basit bir yoludur. Eğitim sırasında, bireysel düğümler belirli bir olasılıkla devre dışı bırakılır. Devre dışı bırakılan bir düğüme gelen ve giden bağlantılar da bırakılır. Bu, aşırı uyum göstermeyi azaltmanın yanı sıra, gereken hesaplama miktarını da azaltır ve daha iyi performans sağlar. Ancak, test sırasında tüm düğümler etkinleştirilir [112].

Ağın ağırlıklarını ilklendirirken, bunların hepsini sıfıra ayarlamamak önemlidir, fakat bu güncellemelerde istenmeyen simetriye yol açabilir. Bunun yerine, örneğin ağırlıkları Gauss dağılımından örnekleyerek küçük, rastgele sayılara ayarlamak genellikle iyi bir fikirdir.

Eğitimde, denklem (3.4)'te açıklandığı gibi, tahminler ve hedefler arasındaki kategorik çapraz entropinin hesaplanmasıyla, ihtiyaçları en aza indirmek için bir kayıp ifadesinin bulunması gerekir. Her i örneği için, tahmin olasılıkları arasındaki çapraz entropi p_i , örneğin bu softmax çıktısı olabilir ve hedef değer t_i hesaplanır. Amaç, ağın eğitimi sırasında bu kayıp ifadesini en aza indirmektir [111].

$$L_i = - \sum_j t_{i,j} \log(p_{i,j}) \quad (3.4)$$

3.2.1.6. Alt-Örnekleme Katmanı

Alt-örnekleme katmanı, girdi haritalarında alt-örnekleme işlemini gerçekleştirir. Bu genellikle havuzlama katmanı olarak ta bilinir. Bu katmanda, girdi ve çıktı öznitelik haritalarının sayısı değişmez. Örneğin, N kadar girdi haritası varsa, tam N kadar çıktı haritası olacaktır. Alt-örnekleme işlemi nedeniyle, alt-örnekleme maskesinin boyutuna bağlı olarak, çıktı haritalarının her boyutunun ebatı küçülecektir. Örneğin, 2×2 alt-örnekleme çekirdeği kullanılırsa, her çıktının boyutu tüm görüntüler için karşılık gelen girdi boyutunun yarısı kadar olacaktır. Bu işlem denklem (3.5)'teki şekilde formüle edilebilir:

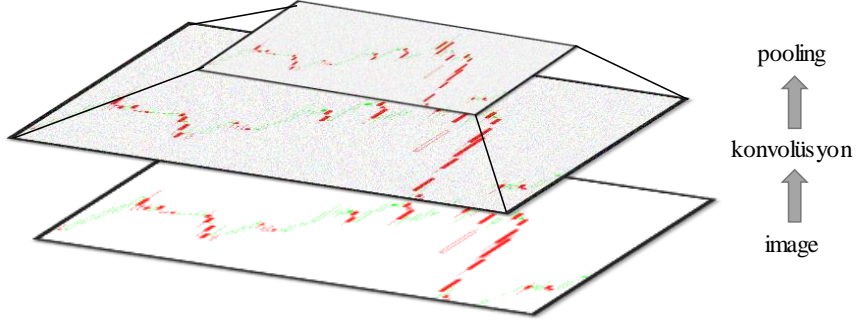
$$x_j^l = \text{down}(x_j^{l-1}) \quad (3.5)$$

$\text{down}()$ bir alt-örnekleme fonksiyonunu temsil eder. Bu katmanda çoğunlukla iki tür işlem gerçekleştirilir: Bu işlemler ortalama havuzlama veya maksimum havuzlama olarak ifade edilir. Ortalama havuzlama yaklaşımı söz konusu olduğunda, fonksiyon genellikle bir önceki katmandan gelen öznitelik haritalarının $N \times N$ yamalarını toplar ve ortalama değeri seçer. Diğer yandan, maksimum havuzlama durumunda, en yüksek değer öznitelik haritalarının $N \times N$ yamaları arasından seçilir. Bu nedenle, çıktı haritası boyutları n kat küçültülür. Bazı özel durumlarda, her çıktı haritası bir skaler ile çarpılır [111], [92].

3.2.1.7. Sınıflandırma Katmanı

Bu katman, önceki adımlarda bir konvolüsyon katmandan çıkarılan özniteliklerden gelen her bir sınıfın skorunu hesaplayan tam bağlantılı katmandır. Son katman öznitelik haritaları, tam bağlantılı katmanlara aktarılan, skaler değerleri olan vektörler olarak temsil edilir. Tam bağlantılı ileri beslemeli sinir katmanları, softmax sınıflandırma katmanı olarak kullanılır. Ağ modeline dâhil edilecek katman sayısı konusunda katı kurallar yoktur. Bununla birlikte, çoğu durumda LeNet [108], AlexNet [8] ve VGG Net [103] gibi farklı mimarilerde iki ila dört katman dâhil edildiği gözlemlenmiştir. Tam bağlantılı katmanlar hesaplama açısından maliyetli olduğundan, geçtiğimiz birkaç yılda alternatif yaklaşımlar önerilmiştir. Bunlar arasında, ağdaki parametre sayısını önemli ölçüde azaltmaya yardımcı olan global ortalama havuzlama katmanı ve ortalama havuzlama katmanı mevcuttur. CNN'ler aracılığıyla geriye doğru yayılımda, tam bağlantılı katman tam bağlantılı sinir ağlarının (FCNN - Fully Connected Neural

Networks) genel yaklaşımı izlenerek güncellenir. Konvolüsyon katmanların filtreleri, konvolüsyon katman ile ondan hemen önce gelen katman arasındaki öznelik haritalarında tam konvolüsyon işlem gerçekleştirilerek güncellenir. Şekil 3.5, bir girdi görüntüsünün konvolüsyonunu ve alt-örneklemedeki temel işlemleri göstermektedir [92].



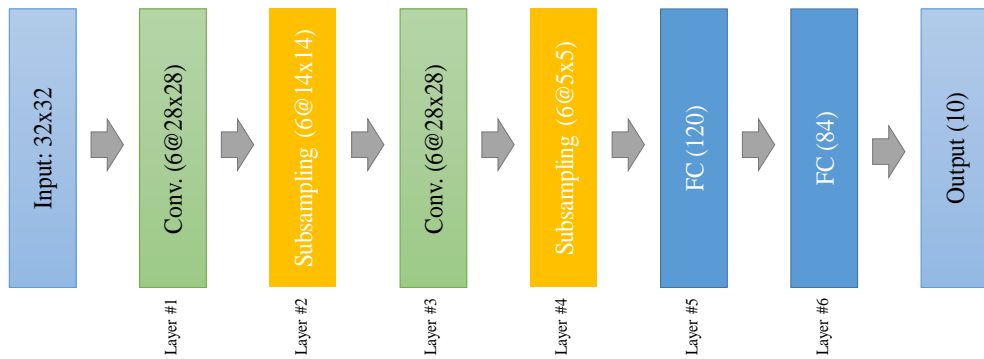
Şekil 3.5. Konvolüsyon ve havuzlama sonrası öznelik haritaları.

3.2.2. Popüler Ağ Mimarileri

Bu bölümde, çeşitli gelişmiş CNN mimarileri incelenecektir. Genel olarak, derin konvolüsyon sinir ağları bir temel katman grubundan oluşur; bu grup konvolüsyon katmanı, alt-örnekleme katmanı, yoğun katmanlar ve softmax katmanını içerir. Mimariler tipik olarak üst üste konulmuş birkaç konvolüsyon katman yığından ve maksimum havuzlama katmanından oluşur, bunları tam bağlantılı katman ve en sonda SoftMax katmanı izler. Bu modellere bazı örnekler LeNet [108], AlexNet [8], VGG Net [103], NiN [113] ve tam konvolüsyoneldir (All Conv) [114]. Başka alternatifler ve daha verimli gelişmiş mimariler de önerilmiştir; bunlar arasında DenseNet [115], FractalNet [116], giriş birimlerine sahip GoogLeNet [102], [117], [118] ve Artık Ağlar [104] vardır. Temel yapı bileşenleri (konvolüsyon ve havuzlama) bu mimarilerin hepsinde neredeyse aynıdır. Bununla birlikte, modern derin öğrenme mimarilerinde bazı topolojik farklılıklar gözlenmektedir. Birçok derin konvolüsyonel sinir ağı (DCNN-Deep Convolutional Neural Networks) mimarisi arasında AlexNet [8], VGG Net [103], GoogLeNet [102], [117], [118], DenseNet [115] ve FractalNet [116], nesne tanıma görevleri için farklı ölçütlerdeki gelişmiş performansları sebebiyle, genellikle en popüler mimariler olarak kabul edilir. Tüm bu yapılar arasında, bazı mimariler özellikle büyük ölçekli veri analizi için tasarlanırken (GoogLeNet ve ResNet gibi), VGG ağı genel bir mimari olarak kabul edilir.

3.2.2.1. LeNet (1998)

LeNet 1990'larda öne sürülmüş olmasına rağmen, sınırlı hesaplama becerisi ve bellek kapasitesi sebebiyle bu algoritmanın uygulanması 2010 yılına kadar mümkün olmamıştır [108]. LeCun ve ark. [108]geriye yayılım algoritmasına sahip CNN'leri önermiş ve en gelişmiş düzeyde doğruluğa ulaşmak için rakamları elle yazılmış veri seti üzerinde deneyler yapmıştır. Önerilen CNN mimarisi LeNet-5 olarak bilinmektedir [108]. LeNet-5'in temel konfigürasyonu Şekil 3.6'daki gibidir. İki konvolüsyon katmanı, iki alt-örnekleme katmanı, iki tam bağlantılı katman ve Gauss bağlantılı bir çıktı katmanından oluşmaktadır. Hesaplama donanımının kapasitesi gelişmeye başlayınca, CNN'ler bilgisayarla görme ve makine öğrenmesi camiasında etkili bir öğrenme yaklaşımı olarak popülerlik kazanmıştır.

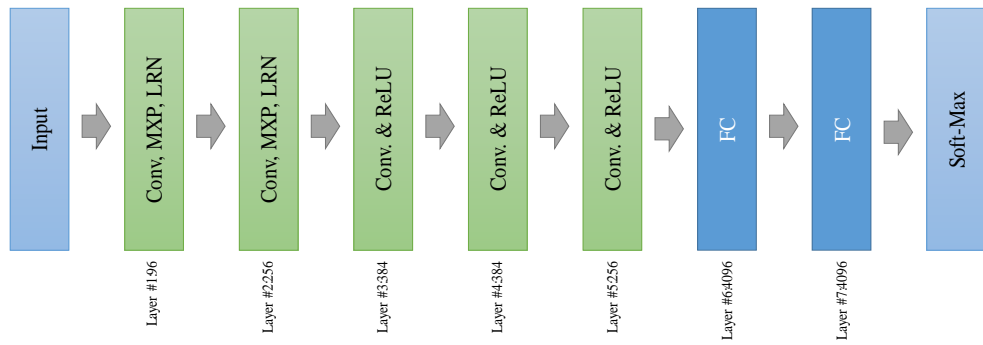


Şekil 3.6. LeNet mimarisi [108].

3.2.2.2. AlexNet (2012)

2012 yılında, Alex Krizhevsky ve diğerleri, LeNet'e kıyasla daha derin ve geniş bir CNN modeli önerdiler ve aynı yıl en zor ImageNet yarışması olan ImageNet Büyük Ölçekli Görsel Tanıma Yarışması (ILSVRC - ImageNet Large Scale Visual Recognition Challenge) adı verilen görsel nesne tanıma yarışmasını kazandılar [8]. AlexNet [8], tüm geleneksel makine öğrenmesi ve bilgisayarla görme yaklaşımları arasında en gelişmiş düzeyde tanıma doğruluğuna ulaşmıştır. Bu, makine öğrenmesi ve görsel tanıma ve sınıflandırma görevleri için bilgisayarla görme alanında devrim oluşturmuş ve derin öğrenmeye ilginin çok hızlı bir şekilde artmasını sağlayan bir başlangıç olarak kabul edilir. AlexNet'in mimarisi Şekil 3.7'de gösterilmektedir. İlk konvolüsyon katman, 11×11 boyutunda 96 farklı alıcı filtrenin kullanıldığı Yerel Yanıt Normalleştirme (LRN - Local Response Normalization) ile konvolüsyon ve maksimum havuzlamayı gerçekleştirir. Maksimum havuzlama işlemleri, adım aralığı boyutu 2 olan 3x3 filtre ile

gerçekleştirilir. Aynı işlemler ikinci katmanda 5x5 filtreler ile gerçekleştirilir. Sırasıyla 384, 384 ve 256 öznitelik haritalarına sahip üçüncü, dördüncü ve beşinci konvolüsyon katmanlarda 3x3 filtreler kullanılır. En sonda SoftMax katmanının takip ettiği bırakmada iki tam bağlantılı (FC - Fully Connected) katman kullanılır. Benzer yapıya ve aynı sayıda öznitelik haritasına sahip iki ağ, bu model için paralel olarak eğitilir. Bu ağda, LRN ve bırakma olmak üzere iki yeni kavram tanıtılmıştır. LRN iki farklı şekilde uygulanabilir: İlk olarak, bir NxN yamasının aynı öznitelik haritasından seçildiği ve komşu değerlere göre normalleştirildiği tek kanal veya öznitelik haritaların üzerinde uygulanır. İkinci olarak, LRN kanallar veya öznitelik haritaları arasında uygulanabilir [92].



Şekil 3.7. AlexNet mimarisi [8].

AlexNet'in üç adet konvolüsyon katmanı ve iki adet tam bağlantılı katmanı vardır. ImageNet veri kümesini işlerken, AlexNet için toplam parametre sayısı ilk katman için şöyle hesaplanabilir: Girdi örnekleri 224x224x3'tür, filtreler (çekirdekler veya maskeler) veya 11 boyutunda bir alıcı alan, adım aralığı 4'tür ve birinci konvolüsyon katmanının çıktısı 55x55x96'dır. Bu ilk katmanın 290.400 (55x55x96) nöronu ve 364 (11x11x3 = 363 + 1 sapma) ağırlığı olduğunu hesaplayabiliriz. İlk konvolüsyon tabakası için parametreler $290.400 \times 364 = 105.705.600$ nörondur.

3.2.2.3. ZFNet / Clarifai (2013)

2013 yılında, Matthew Zeiler ve Rob Fergus, 2013 ILSVRC'yi AlexNet'in eklentisi olan bir CNN mimarisiyle kazanmıştır. Ağ, yazarların isimlerine ithafen ZFNet [119] olarak adlandırılmıştır. CNN hesaplama açısından maliyetli olduğundan, modellerin karmaşıklığından yola çıkılarak, parametrelerin en etkin şekilde kullanılması gerekmektedir. AlexNet'in ağ parametrelerinde ufak değişiklikler yapılarak tasarlanan ZFNet mimarisi, AlexNet'in geliştirilmiş halidir. ZFNet, ağırlık sayısını önemli ölçüde

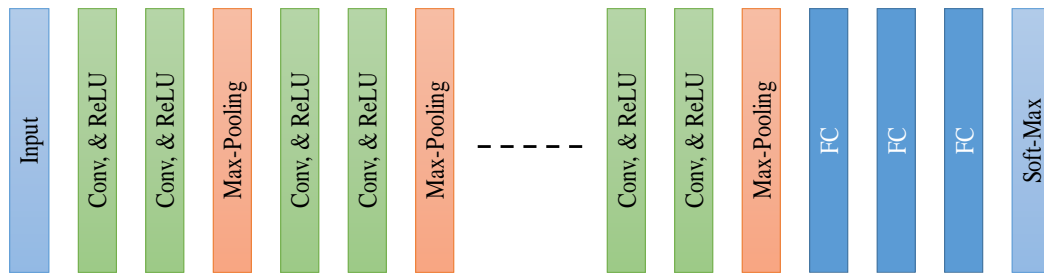
azaltmak için 11x11 çekirdekler yerine 7x7 çekirdek yapısı kullanmıştır. Bu, ağ parametrelerinin sayısını önemli ölçüde azaltır ve tanıma doğruluk düzeyini genel olarak arttırmaktadır.

3.2.2.4. Network in Network

Bu model, birkaç yeni kavram ortaya koymasıyla önceki modellerden bir parça farklıdır [113]. İlk kavram, modellere daha fazla doğrusalsızlık eklemek için konvolüsyonların 1x1 filtre ile uygulandığı çok katmanlı algı konvolüsyonu kullanmaktadır. Bu, ağın derinliğini artırmaya yardımcı olur, böylece derinlik bırakma ile düzenlenebilir. Bu kavram, genellikle derin öğrenme modelinin tıkanıklık katmanında kullanılır. İkinci kavram, global ortalama havuzlamayı (GAP - Global Average Pooling) tam bağlantılı katmanlara alternatif olarak kullanmaktır. Bu, ağ parametrelerinin sayısını önemli ölçüde azaltmaya yardımcı olur. GAP ağ yapısını büyük oranda değiştirir. GAP büyük bir öznitelik haritasında uygulayarak, öznitelik haritalarının boyutlarını küçültmeden son bir küçük boyutlu öznitelik vektörü oluşturulabilir [92].

3.2.2.5. VGGNET (2014)

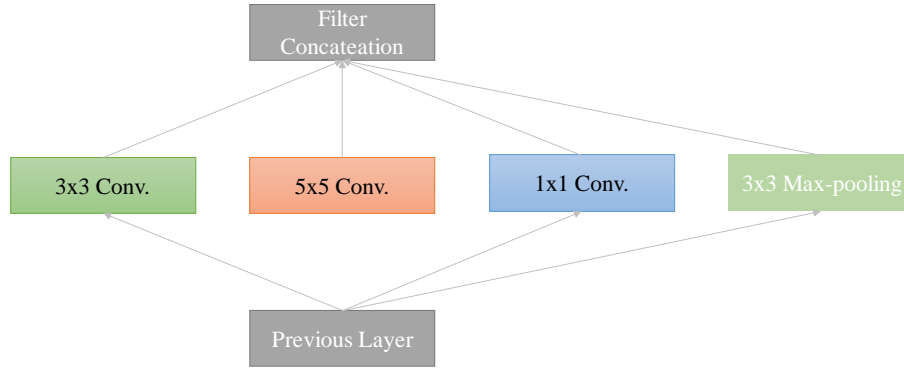
Görsel Geometri Grubu (VGG - Visual Geometry Group), 2014 ILSVRC ikincisidir [103]. Bu çalışmanın ana katkısı, CNN’de daha yüksek düzeyde tanıma ve sınıflandırma doğruluğu elde etmek için ağ derinliğinin kritik bir bileşen olduğunu göstermiş olmasıdır. VGG mimarisi, her ikisi de ReLU aktivasyon fonksiyonunu kullanan iki konvolüsyon katmanından oluşur. Aktivasyon fonksiyonunun arkasından tek bir maksimum havuzlama katmanı ve yine ReLU aktivasyon fonksiyonunu kullanan tam bağlantılı birkaç katman gelir. Modelin son katmanı, sınıflandırma için Softmax katmanıdır. VGG-E’de [103], konvolüsyon filtresi boyutu adım aralığı 2 olan 3x3 filtre ile değiştirilir. Üç VGG-E [103] modeli, VGG-11, VGG-16 ve VGG-19, modellerin sırasıyla 11, 16 ve 19 katmana sahip olmasını önermiştir. VGG ağ modeli Şekil 3.8’de gösterilmektedir.



Şekil 3.8. VGG ağ mimarisi [103].

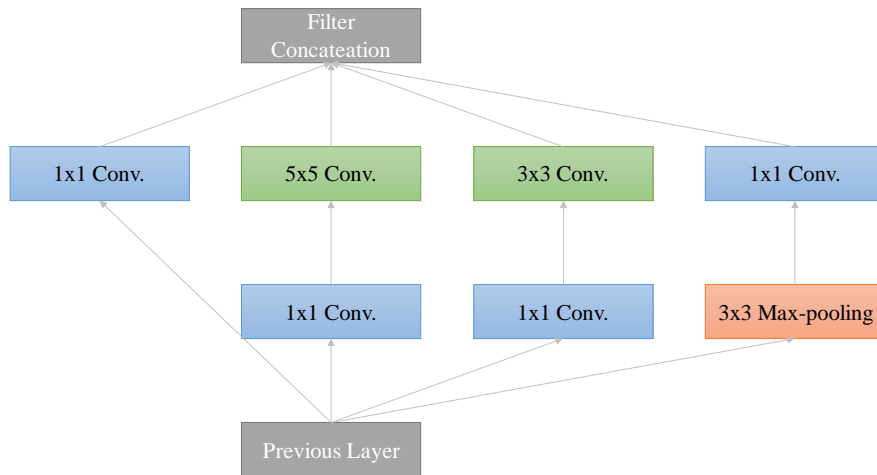
3.2.2.6. GoogLeNet (2014)

2014 ILSVRC [102] şampiyonu GoogLeNet, Google ekibinden Christian Szegedy tarafından, geleneksel CNN'ne kıyasla hesaplama karmaşıklığını azaltmak amacıyla geliştirilmiştir. Önerilen yöntemin amacı, farklı çekirdek boyutları tarafından oluşturulan değişken alıcı alanlara sahip giriş katmanlarını dâhil etmektir. Giriş katmanı kavramının ilk öne sürülen hali Şekil 3.9'da görülebilir.



Şekil 3.9. Giriş katmanı [102].

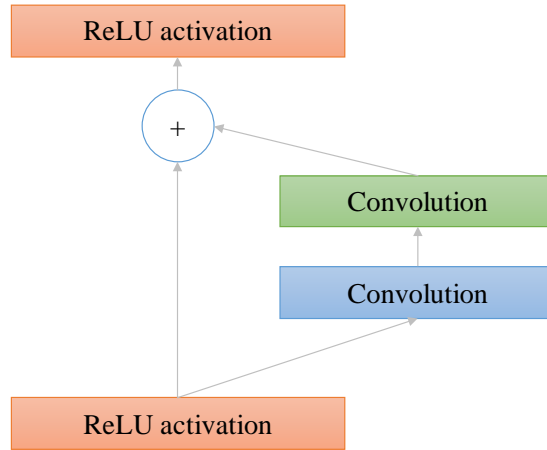
GoogLeNet, Şekil 3.10'da görüldüğü gibi, bir giriş katmanları yığını kullanılarak en gelişmiş seviyede tanıma doğruluğuna ulaşılmıştır. Başlangıç katmanı ve son başlangıç katmanı arasındaki fark, 1x1 konvolüsyon çekirdeklerinin eklenmesidir. Bu çekirdekler, hesaplama açısından maliyetli katmanlardan önce boyutsallığın azaltılmasına olanak tanımıştır. GoogLeNet, toplamda 22 katmandan oluşur; bu, kendisinden önceki tüm ağların sahip olduğu katman sayısından çok daha fazladır. Sonradan bu ağın geliştirilmiş versiyonu önerilmiştir [117]. Ancak, GoogLeNet'in kullandığı ağ parametresi sayısı kendisinden önce gelen AlexNet veya VGG'den çok daha azdır.



Şekil 3.10. Boyut küçültmeli giriş katmanı [102].

3.2.2.7. ResNet (2015)

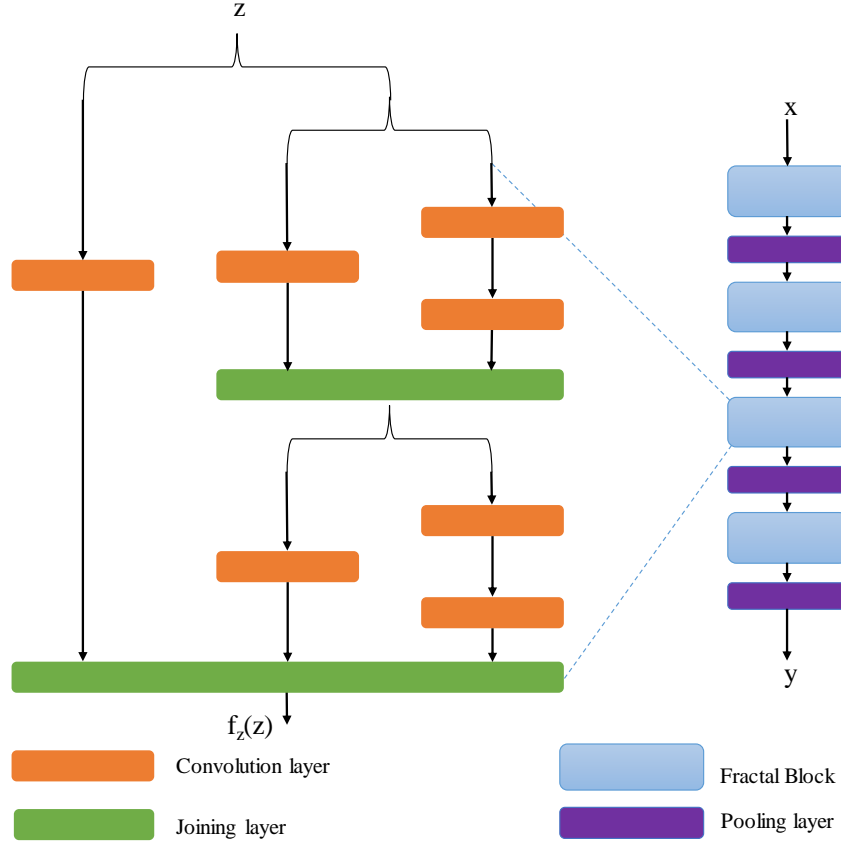
2015 ILSVRC galibi, ResNet (Residual Network (Artık Ağ)) mimarisi olmuştur [104]. ResNet, Kaiming He tarafından, önceki modellerin sahip olduğu gradyanların yok olması problemine sahip olmayan ultra derin ağlar tasarlamak niyetiyle geliştirilmiştir. ResNet pek çok farklı sayıda katmanla geliştirilmiştir: 34, 50, 101, 152 ve hatta 1202. Popüler ResNet50, ağın sonunda 49 konvolüsyon katmanı ve 1 tam bağlantılı katman içerir. ResNet mimarisinin temel blok diyagramı Şekil 3.11’de gösterilmektedir



Şekil 3.11. ResNet temel blok diyagramı [104].

3.2.2.8. FractalNet (2016)

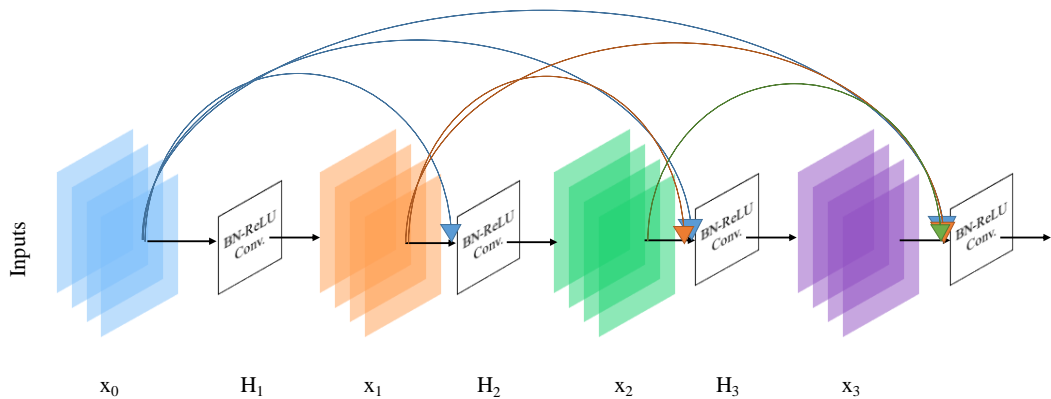
Bu mimari, ResNet modelinin gelişmiş, alternatif mimarisidir ve eğitim sırasında daha kısa yayılım yollarına sahip büyük modeller tasarlamakta etkilidir [116]. Bu kavram, büyük ağlar oluşturmak için bir başka düzenleme yaklaşımı olan bırakma yöntemine dayanmaktadır. Sonuç olarak, bu kavram eşitlik ve etkinlik dengesinin güçlendirilmesine yardımcı olur. FractalNet’in temel blok diyagramı Şekil 3.13’de gösterilmiştir.



Şekil 3.12. FractalNet modülü [116].

3.2.2.9. Yoğun Bağlantılı Ağ (DenseNet - Densely Connected Network) (2017)

Gao ve ark. tarafından 2017’de geliştirilen DenseNet [115], yoğun bağlantılı CNN katmanlarından oluşur; her bir katmanın çıktıları, bir yoğun bloktaki tüm ardıl katmanlarla bağlantılıdır [115]. Dolayısıyla, katmanlar yoğun bağlantı ile oluşturulduğu için, DenseNet adını almıştır. Bu kavram, gelecekte tekrar kullanım için uygundur, bu da ağ parametrelerini önemli ölçüde azaltır. DenseNet, iki bitişik yoğun blok arasına yerleştirilen birkaç yoğun blok ve geçiş bloklarından oluşur. Yoğun bloğun kavramsal diyagramı Şekil 3.13’de gösterilmiştir.



Şekil 3.13. DenseNet mimarisi [115].

3.2.3. Diğer DNN (Deep Neural Networks) Modelleri

Bilgisayarla görme camiasında popüler olan, hızlı bölge tabanlı CNN [120] ve Xception [121] gibi, birçok başka ağ mimarisi vardır. 2015 yılında, Tekrarlayan konvolüsyonlu Sinir Ağı (RCNN-Recurrent Convolution Neural Network) adı verilen, tekrarlayan konvolüsyon katmanlarını kullanan yeni bir model önerilmiştir [122]. Bu ağın geliştirilmiş versiyonu, başlangıç ağı ile tekrarlayan konvolüsyon ağındaki iki popüler mimarinin birleşimidir ve Giriş konvolüsyonlu Tekrarlayan Sinir Ağları ismiyle anılır (IRCNN-Inception Convolutional Recurrent Neural Networks) [123]. IRCNN, neredeyse aynı ağ parametrelerine sahip RCNN ve giriş ağına kıyasla daha yüksek düzeyde doğruluk sağlamıştır. Bağıntısal bileşenler arasında bağlantılar kurmak için, faz güdümlü mesaj içeren ve bir yapıdan geçen Görsel Faz Güdümlü CNN (ViP CNN - Visual Phase Guided) önerilmiştir ve sonuçta daha iyi hızlanma ve tanıma doğruluğu performansı göstermiştir [124]. Arama tabanlı CNN [125], verimli çıkarsama sağlayan hızlı, kompakt ve hatasız bir modeldir. 2016 yılında, bölütleme görevleri için, günümüzde yaygın olarak kullanılan ve tam konvolüsyonel ağ (FCN - Fully Convolutional Network) olarak bilinen mimari önerilmiştir [126]. Yakın zamanda önerilen diğer CNN modelleri arasında piksel ağı [127], skolastik derinliğe sahip derin bir ağ, derin denetimli ağlar ve merdiven ağı [128], [129], [130] bulunmaktadır. Ek olarak, CNN mimari modelleri [131]'da açıklanmaktadır. “Derin sinir ağları gerçekten derin olmak zorunda mı?” konulu bazı makaleler yayınlanmıştır [132], [133], [134]. FitNet isabetleri [135], başlatma yöntemi [136], derin ağa karşı geniş ağ [137], büyük eğitim setinde [138] DL eğitimi, grafik işleme [139], enerji tasarruflu ağ mimarileri [140], [141] hakkında yayınlanmış bazı makaleler bulunmaktadır.

3.2.4. CNN Uygulamaları

3.2.4.1. Grafik Sorunu Çözmek için CNN'ler

Grafik veri yapılarını öğrenmek, veri madenciliği ve makine öğrenmesi görevleri yerine getiren çeşitli uygulamalarda yaygın bir sorundur. DL teknikleri, makine öğrenmesi ve veri madenciliği grupları arasında bir köprü kurmuştur. 2016'da rastgele grafik işleme için etkili bir CNN önerilmiştir [142].

3.2.4.2. Görüntü İşleme ve Bilgisayarla Görme

Yukarıda tartışılan modellerin çoğu, görüntü sınıflandırması [8], [119], [103], [102], [104], saptama, bölütleme, yerelleştirme, altyazı, video sınıflandırması gibi pek çok farklı uygulama alanında hayata geçirilmiştir. Görüntü işleme ve bilgisayarla görme ile ilgili, görsel sınıflandırma, bölütleme ve saptamayı da içeren görevlere yönelik DL yaklaşımları üzerine yapılmış birçok araştırma mevcuttur [104]. Bu araştırmaların başlıcaları CNN yöntemi [143] kullanarak tek görüntü süper çözünürlüğü, blok eşleştirmeli CNN [144] kullanarak görüntüyü gürültüden arındırma A-Lamp (Adaptif Düzen-Duyarlı Çoklu Yama Derin CNN) [145], hiperspektral görüntüleme bölümlenmesi için DCNN [146], görüntü çakıştırma [147], hızlı sanatsal üslup transferi [148], DCNN kullanarak görüntü arka plan bölümlenmesi [149], el yazısı karakter tanıma [150], optik görüntü sınıflandırma [151], yüksek çözünürlüklü uydu görüntüleri [152] kullanarak ekin haritalama, hücresel eşzamanlı tekrarlayan ağlar ve CNN ile nesne tanıma [153] şeklinde isimlendirilen çalışmalardır. DL yaklaşımları, insan aktivitelerini tanıma görevlerinde yoğun olarak uygulanır ve mevcut yaklaşımlara kıyasla üst düzeyde performans göstermektedirler [154], [155], [156], [157], [158], [159]. Bununla birlikte, sınıflandırma, bölütlere ayırma ve tespit görevi için en gelişmiş modeller aşağıdaki gibidir:

(1) *Sınıflandırma sorunları için modeller:* sınıflandırma modellerinin mimarisine göre, girdi görüntüleri konvolüsyon ve alt örnekleme katmanları ile farklı bir adımda kodlanır ve son olarak sınıf olasılığını hesaplamak için SoftMax yaklaşımı kullanılır. Sınıflandırma modellerinin listesi şöyledir: AlexNet, VGGNet, GoogleNet, ResNet, DenseNet, FractalNet, CapsuleNet, IRCNN, DCRN vb.

(2) *Bölütleme sorunları için modeller:* Son birkaç yılda öne sürülen birkaç anlamsal bölütleme modeli vardır. Bölütleme modeli iki birimden oluşur: Kodlama ve kod çözme birimleri. Kodlama biriminde, kod çözme birimi gizli boşluktan gelen görüntüyü ters konvolüsyon ve alt-örnekleme işlemlerini uygulayarak kodlarken, alt boyuttaki gizli boşluğa kodlamak için konvolüsyon ve alt-örnekleme işlemleri gerçekleştirilir. İlk bölütleme modeli Tam konvolüsyon Ağı'dır (FCN - Fully Convolutional Network) [126], [160]. Daha sonra, bu ağın SegNet olarak adlandırılan geliştirilmiş versiyonu önerilmiştir [161]. Yakın zamanda RefineNet [162], PSPNet [163], DeepLab [160], UNet [164] ve R2U-Net [165] gibi pek çok yeni model önerilmiştir.

(3) *Saptama sorunları için modeller:* saptama sorunu, sınıflandırma ve bölütleme sorunlarına göre biraz farklıdır. Bu durumda, modelin hedef türlerini karşılık gelen pozisyonu ile tanımlamaktır. Bu model iki soruya cevap verir: Nesne nedir (sınıflandırma sorunu)? ve nesne nerede (regresyon sorunu)? Bu hedeflere ulaşmak için, öznelik çıkarma modülünün tepesindeki sınıflandırma ve regresyon modülü için iki adet kayıp hesaplanır ve model ağırlıkları her iki kayba göre güncellenir. İlk kez, nesne saptama görevi için bölge tabanlı CNN (RCNN - Region based CNN) önerilmiştir [166]. Geçtiğimiz yakın zamanda, yoğun nesne dedektörü için odak kaybı da dahil olmak üzere bazı daha gelişmiş saptama yaklaşımları önerilmiştir [167]. Sonradan bu ağırlık daha hızlı RCNN, hızlı RCNN [120], [168], maske R-CNN [169], You Only Look Once (YOLO) [170], SSD: Single Shot MultiBox Detector [171] ve patolojik görüntülerden doku tespiti için UD-Net [172] adı verilen farklı, geliştirilmiş versiyonları önerilmiştir.

3.2.4.3. Konuşma İşleme

CNN'ler ayrıca, çok derinli derin CNN [173] kullanarak konuşma geliştirme ve Konvolüsyonel Geçitli Tekrarlayan Ağ (CGRN - Convolutional Gated Recurrent Network) [174] kullanarak ses etiketleme gibi, konuşma işleme görevleri için de uygulanmaktadır.

3.2.4.4. Tıbbi Görüntüleme

Litjens ve ark. [175] sınıflandırma, saptama ve bölütleme görevlerini de içeren tıbbi görüntü işleme için DL üzerinde etkili bir araştırma ortaya koymuşlardır. Tıbbi görüntü analizi için birkaç popüler DL yöntemi geliştirilmiştir. Örneğin MDNet, görüntüleri ve karşılık gelen metin açıklamalarını [176] kullanarak tıbbi teşhis, kısa-eksen MRI kullanarak kalp bölütlemesi [177], CNN kullanarak optik disk ve retinal vaskülit [178], FCNN [179] ile öğrenilen özneliklere sahip rastgele ormanları kullanarak beyin tümörü bölütlemesi için geliştirilmiştir. Bu teknikler, hesaplamalı patoloji alanında uygulanmış ve en yüksek düzeyde performansa ulaşmıştır [172], [180], [181], [182].

3.3. TRANSFER ÖĞRENME

Transfer öğrenimi, T1 görevini çözebilmesi için büyük miktarda veri üzerinde önceden eğitilmiş bir CNN kullanma pratiğini ifade eder. T2 görevini çözmek için, CNN içerisindeki bir veya daha fazla katmanı yenisiyle değiştirmek ve daha sonra onu farklı,

daha küçük bir veri kümesi üzerinde eğitmek gerekir. Örnek olarak, CNN çok büyük bir veri kümesi kullanılarak eğitildikten sonra arabaları sınıflandırabileceğini varsayın. Çıktı katmanını doğrusal bir katmanla değiştirin ve onu kamyonları çok daha küçük bir veri kümesi kullanarak sınıflandırmak için eğitin. Örnek vermek gerekirse, ImageNet [183], yaklaşık 14 milyon görüntü içeren büyük bir veri kümesidir ve genellikle CNN'nin önceden eğitildiği veri kümesi olarak kullanılır. Transfer öğrenimi usulen aşağıdaki gibi tanımlanabilir:

Bu tür bir öğrenme yaklaşımının işe yaramasının temel nedeni, CNN'lerin sınıflandırma sorunları için evrensel olan unsurları erken öğrenmeleridir. Bu, bunlarla sınırlı olmamak kaydıyla ayırt saptama, renk saptama ve ışık değişikliklerini içerir. Transfer öğreniminden ilk kez, Pratt [184] tarafından kaleme alınan (Discriminability-Based Transfer between Neural Networks (Sinir Ağları Arasında Ayırt Edebilirlik Temelli Aktarım)) makalede bahsedilmiştir. Transfer öğrenimi, CNN'ni sıfırdan eğitmek için gereken iki şeyi, yani inanılmaz derecede büyük bir veri kümesini ve çok fazla bilgi işlem gücünü azaltmak için kullanılır. Transfer öğrenimi söz konusu olduğunda iki ana fikir vardır: ayarlama ve öznelik çıkarma. Her ikisi de eğitim sırasında ağ içerisindeki ağırlıkları değiştirme ile ilgilidir, ancak bunu gerçekleştirme şekilleri oldukça farklıdır. Özellik çıkarmayı kullanırken, ana fikir ağdaki ilk katmanları dondurmak ve yalnızca son katmanları eğitmektir. Buna karşılık ayarlamayı kullanırken, ağdaki tüm ağırlıkların değişmesine izin verilerek, normal şekilde eğitmek. Bu yaklaşımların her ikisi de eğitimden önce ağdaki bazı katmanları yenisiyle değiştirir.

3.4. BİLGİSAYAR GÖRÜŞÜ

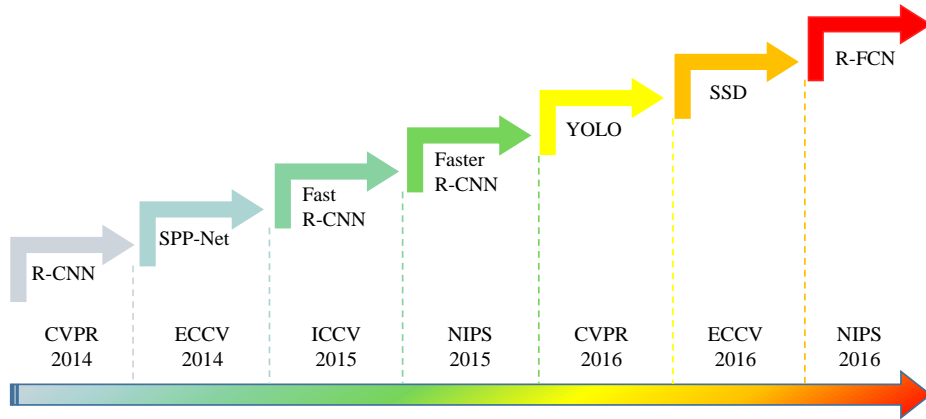
Bilgisayarla görme, 60'lı yıllardan beri bilgisayar bilimciler tarafından incelenen bir konudur ve bilgisayarların görüntü veya videolardan üst düzeyde bilgiler çıkarmasını sağlamayı amaçlar. Bu tür üst düzey bilgilere örnek olarak, bir görüntüde hangi nesnenin var olduğu veya nesnenin kameradan ne kadar uzakta olduğu gösterilebilir. Bu tür bir bilgi elde etme girişiminin hem algoritmaların hem de yapay zekânın kullanıldığı pek çok farklı yöntemi vardır. Kullanılan algoritmalara örnek olarak SIFT [185], SURF [186] ve BRIEF [187] verilebilir. Yakın zamanda, bilgisayarla görme alanı derin öğrenme modelleri tarafından devralınmıştır, çünkü her iki alandaki sorunların çoğu büyük oranda örtüşmektedir. Son yıllarda hesaplama gücünün artması da bilgisayarla görme alanında derin öğrenmenin yükselmesine katkıda bulunmuştur [188].

Bilgisayarla görme, bilgisayarların gördüklerini insanların gördüklerine benzer şekilde görselleştirme, tanıma ve işleme yetkisine sahip olacak şekilde güçlendirildikleri çalışma alanıdır [189]. Bilgisayarla görmenin temel amacı, dünya hakkında bir şeyler çıkarsamak için görüntü ve video verilerinden ilgili bilgiler üretmektir [190], [191]. Bilgisayarla görme, yapay zekâ ve makine öğrenmesinin bir alt alanı olarak sınıflandırılabilir. Bu, görsel bilginin manipüle edilmesini veya iyileştirilmesini içeren görüntü işlemeden oldukça farklıdır ve görüntünün içeriği ile ilgilenmez. Bilgisayarla görme uygulamaları arasında görüntü sınıflandırması, görsel saptama, 2D görüntülerden 3D sahne oluşturma, görüntü elde etme, artırılmış gerçeklik, yapay görme ve trafik otomasyonu bulunmaktadır [192], [193], [194], [195], [196], [197]. Bugün, makine öğrenmesi pek çok bilgisayarla görme algoritmasının gerekli bir bileşenidir [198]. Bu algoritmalar tipik olarak görüntü işleme ve makine öğrenmesi tekniklerinin kombinasyonudur. Bu algoritmaların temel gereksinimi, büyük miktarda görüntü/video verisini işlemek ve çok çeşitli uygulamalar için gerçek zamanlı hesaplama yapabilmektir.

4. NESNE TANIMA ALGORİTMALARI VE UYGULAMALARI

4.1. NESNE TANIMA ALGORİTMALARI

Bilgisayarla görme alanında uzun süredir devam eden, temel ve zorlu bir sorun olan nesne tespitinin üzerinde onlarca yıldır aktif şekilde araştırma yapılmaktadır. Nesne tespitinin amacı, belirli bir görüntüde belirlenen kategorilerden (insanlar, arabalar, bisikletler, köpekler ve kediler gibi) herhangi bir nesne örneğinin olup olmadığını tespit etmek ve varsa her bir nesne örneğinin uzamsal konumunu ve kapsamını bildirmektir (belirli bir sınırlayıcı kutu ile [199], [109]). Görüntü algılama ve bilgisayarla görmenin temel taşı olan nesne tespiti, bölütleme, sahne anlama, nesne izleme, görüntü alt yazılama, olay tespiti ve etkinlik tanıma gibi daha karmaşık veya üst düzey görüntü görevlerini çözenin temelini oluşturur. Nesne tespiti, yapay zekâ ve bilgi teknolojilerinin robot görüntüleme, tüketici elektroniği, güvenlik, otonom sürüş, insan bilgisayar etkileşimi, içeriğe dayalı görüntü elde etme, akıllı video gözetimi ve artırılmış gerçeklik gibi pek çok alanında geniş bir uygulama alanına sahiptir. Yakın zamanda, derin öğrenme teknikleri [95], [89], öznelik temsillerini verilerden otomatik olarak öğrenme için güçlü yöntemler olarak öne çıkmışlardır. Zamansal olarak nesne tespiti kapsamındaki algoritmalarının gelişimlerine ait tarihsel gösterim Şekil 4.1’te verilmiştir.

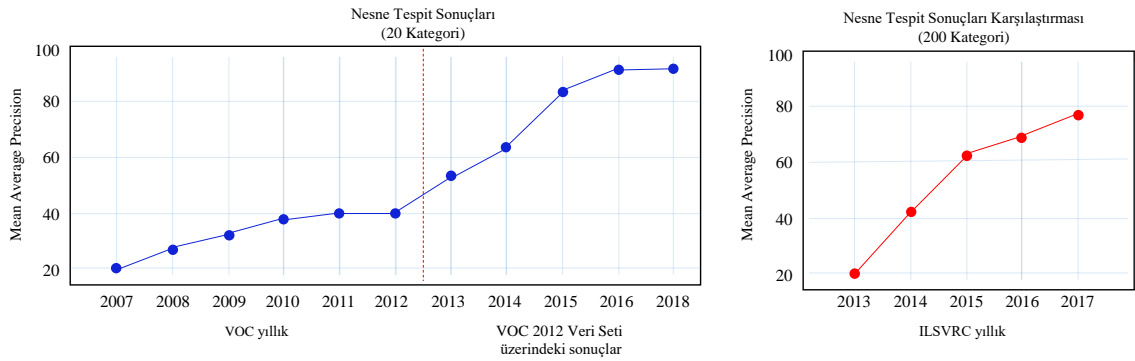


Şekil 4.1. Nesne tespit algoritmalarının tarihsel gelişimi [200].

Şekilde verilen algoritmalar genel isimli olup günümüzde halen sürekli olarak geliştirilen versiyonları mevcuttur. Bu teknikler özellikle, psikofizikçiler, sinirbilimciler ve mühendisler tarafından on yıllardır araştırılmasına rağmen son beş yılda çok daha

büyük ilgi gören nesne tespiti sorununda önemli ilerlemeler sağlamıştır. Tarihsel olarak bakılacak olursa, nesne tespiti alanındaki çalışmaların çoğu tek bir kategorinin (yüzler ve yayalar gibi) veya birkaç belirli kategorinin tespitine odaklanmıştır. Buna karşılık, son birkaç yılda araştırma camiası, nesne tespiti kabiliyeti açısından insanla yarışabilecek, genel amaçlı nesne tespiti sistemleri geliştirmek gibi zorlu bir hedefe doğru ilerlemeye başlamıştır. 2012’de Krizhevsky ve ark. [8], AlexNet adında, ILSVRC’de [109] görüntü sınıflandırma doğruluğunda rekor kıran bir Derin Konvolüsyon Sinir Ağı (DCNN) önermiştir. O zamandan bu yana, bilgisayarla görme uygulama alanlarının çoğunda araştırmaların odağı derin öğrenme yöntemleri olmuştur. Derin öğrenmeye dayalı yaklaşımların büyük çoğunluğu genelleyici nesne tespiti [166], [201], [202], [101], [203], [120] çalışmalarında ortaya çıkmıştır. Genelleyici nesne tespiti sorununun kendisi şu şekilde tanımlanmaktadır: Rasgele bir görüntü verildiğinde, önceden tanımlanmış kategorilerden herhangi bir anlamsal nesne örneği olup olmadığını belirlemek ve varsa eğer uzamsal konum ve kapsamı bildirmektir. Nesne, görülebilen ve dokunabilen, maddi bir şeyi ifade eder. Nesne sınıfı tespiti ile büyük ölçüde eş anlamlı olmasına rağmen, genelleyici nesne algılama nesne örnekleri veya özelleşmiş kategoriler (ör., yüzler, yayalar veya arabalar) yerine geniş yelpazedeki doğal kategorileri tespit etmeyi hedefleyen yaklaşımlara daha fazla vurgu yapar. 2006’dan 2012’ye kadar düzenlenen PASCAL VOC tespit yarışması [199], 2013’ten beri düzenlenen ILSVRC büyük ölçekli tespit yarışması [109] ve 2015’ten beri düzenlenen MS COCO büyük ölçekli tespit yarışması [204] gibi nesne tespiti yarışmalarında son yıllarda görülen artıştan da anlaşılacağı üzere, genelleyici nesne tespiti çok büyük oranda ilgi görmektedir. Son yıllardaki gelişmeler Şekil 4.2’de gösterilmektedir.

Turing Point in 2012: Deep Learning Achieved Record Breaking Image Classification Result



Şekil 4.2. Nesne tespiti performans gelişmeleri [205].

Derin öğrenmenin 2012’de sahneye girişinden bu yana, gözle görülür oranda performans (ortalama kesinlik) iyileşmesi gözlemlenmiştir. En iyi dedektörün performansı her yıl istikrarlı bir şekilde, önemli miktarda artmaktadır.

(a) PASCAL VOC veri setlerine ilişkin sonuçlar: 2007-2012 VOC yarışmalarında kazanan adayların tespit sonuçları (yalnızca sağlanan eğitim verilerini kullanarak).

(b) 2013-2017 ILSVRC’deki en iyi nesne tespiti sonuçları (yalnızca sağlanan eğitim verilerini kullanarak).

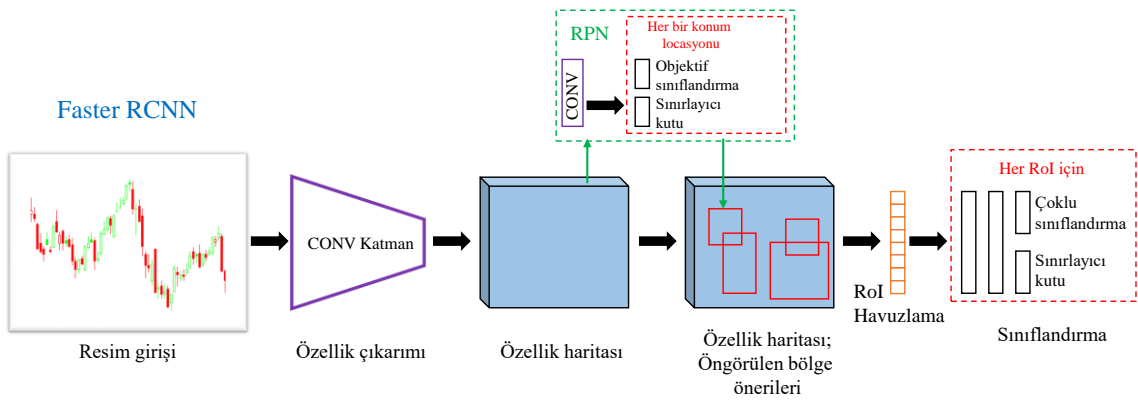
Gerçek zamanlı nesne tespitinin amacı nesnelere gerçek zamanlı olarak tespit etmek ve sınıflandırmak olan bilgisayar görmesinin bir alt kategorisidir. Bu, derin öğrenme modellerinin çıkarsama süresinin, video gibi gerçek zaman kısıtlamaları altındaki bir girdiyle uğraşırken, aynı zamanda yüksek bir doğruluk düzeyini muhafaza edebilecek kadar hızlı olması gerektiği anlamına gelir. Faster R-CNN: Bölge Önerisi Ağları ile Gerçek Zamanlı Nesne Tespitine Doğru [120] başlıklı makale, 15 FPS’ye ulaşacak kadar hızlı çıkarsama yapma olasılığının olduğunu göstermiştir. Birkaç ay sonra, ilk YOLO makalesi [170] yayınlanmıştır. YOLO son derece hızlı ve 63.4 mAP skoruna ulaşırken 45 FPS’ye izin veren çıkarsama hızlarına çıkabilmiştir. Gerçek zamanlı nesne dedektörleri SSD [171], YOLOv3 [206] ve YOLOv4 [207] tarafından aşamalı olarak geliştirilmişlerdir.

4.1.1. Faster R-CNN

Fast R-CNN algılama sürecini önemli ölçüde hızlandırırsa da hala dış bölge önerilerine dayanmaktadır. Bölge önerisi hesaplaması, Fast R-CNN’de yeni darboğaz olarak ortaya çıkmaktadır. Son zamanlarda yapılan çalışmalar CNN’lerin konvolüsyon katmanlarında [208], [209], [210], [211], [212] nesnelere yerleştirme konusunda dikkate değer bir beceriye sahip olduklarını göstermiştir; bu yeti FC katmanlarda zayıflar. Bu nedenle, bölge önerisi üretme sırasında seçici arama CNN tarafından değiştirilebilir. Ren ve ark. tarafından önerilen Faster R-CNN çerçevesi [120], bölge önerileri üretmek için etkili ve doğruluğu yüksek bir Bölge Önerisi Ağı (RPN) önermiştir. Ren ve ark., bölge önerisi için RPN, bölge sınıflandırması için Hızlı RCNN görevini yerine getirmek için tek bir ağ kullanırlar. Faster R-CNN’de, RPN ve Fast RCNN çok sayıda konvolüsyon katmanını paylaşır. Son paylaşılan konvolüsyon katmanının özellikleri, bölge önerisi ve ayrı dallardan bölge sınıflandırması için kullanılır. Bölge önerisi ağı (RPN) ilk olarak, her bir konvolüsyon öznetelik haritası lokasyonunda farklı ölçeklere ve en-boy oranına

sahip $k \times n$ referans kutularını (ör., tespit noktası denilen kısımları) başlatır. Her $n \times n$ tespit noktası, iki FC kardeş katmanın bir nesne kategorisi sınıflandırması ve bir regresyon kutusu içine beslenen alt boyutlu bir vektör ile (ZF için 256 ve VGG için 512 gibi) eşlenir. Fast R-CNN'den farklı olarak, RPN'de regresyon için kullanılan öznelikler aynı boyuta sahiptir. RPN, Fast R-CNN ile konvolüsyon özneliklerini paylaşır, böylece yüksek randımanlı bölge önerisi hesaplamasına olanak tanır. RPN aslında bir çeşit FCN'dir [126]; bu durumda Faster R-CNN, el işi öznelikler kullanılmadan oluşturulan, tamamen CNN tabanlı bir çerçevedir. Çok derin VGG16 modelinde [213], Faster R-CNN, GPU üzerinde 5 FPS'de (tüm adımlar dâhil) test edebilirken, PASCAL VOC 2007'de görüntü başına 300 öneri kullanarak üst düzeyde nesne tespiti doğruluğuna ulaşabilir. [120]'teki ilk Fast R-CNN, birkaç alternatif eğitim adımı içerir. Daha sonradan [214]'deki çalışma tek adımda müşterek eğitim ile basitleştirilmiştir.

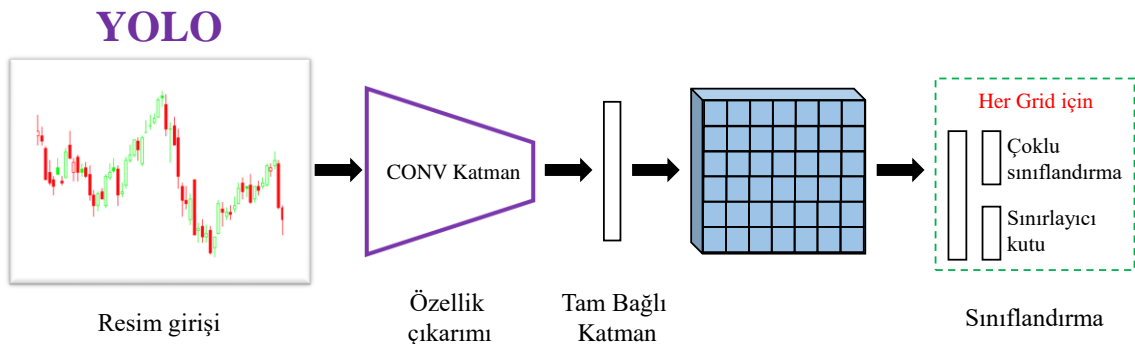
Faster RCNN ile eş zamanlı olarak, Lenc ve Vedaldi [215] seçici arama gibi bölge önerisi oluşturma yöntemlerinin rolüne meydan okuyarak CNN tabanlı dedektörlerde bölge önerisi oluşturma rolünü incelemiş ve CNN'lerin tam bağlantılı katmanlardan ziyade konvolüsyon katmanlarında doğru nesne tespiti için yeterli düzeyde geometrik bilgi içerdiğini bulgulamıştır. Lenc ve Vedaldi, seçici arama gibi bölge önerisi oluşturma yöntemlerini bertaraf ederek, yalnızca konvolüsyon sinir ağlarına (CNN) dayanan, entegre, daha basit ve daha hızlı nesne dedektörleri oluşturma mümkün olduğunu kanıtlamışlardır.



Şekil 4.3. Faster R-CNN mimari [205].

4.1.2. YOLO (You Only Look Once)

Redmon ve ark., [216] nesne tespitini görüntü piksellerinden kaynaklanan bir regresyon problemi olmaktan uzamsal olarak ayrılmış sınırlayıcı kutulara ve ilişkili sınıf olasılıklarına taşıyan birleştirilmiş bir dedektör olan YOLO'yu önermiştir. YOLO'nun tasarımı Şekil 4.4'de gösterilmiştir. Bölge önerisi oluşturma aşaması tamamen devre dışı bırakıldığından, YOLO tespitleri küçük bir aday bölge grubu kullanarak doğrudan tahmin etmektedir. Tespitleri yerel bölgeden gelen özniteliklere göre tahmin eden, Faster R-CNN gibi bölge tabanlı yaklaşımların aksine, YOLO görüntünün bütününden gelen global öznitelikleri kullanır. Özellikle belirtmek gerekirse, YOLO görüntüyü bir $S \times S$ ızgarasına böler. Her bir ızgara C sınıfı olasılıkları, B sınırlayıcı kutu konumlarını ve bu kutular için güven skorlarını tahmin eder. Bu tahminler bir $S \times S \times (5B + C)$ tensörü olarak kodlanır. YOLO, bölge önerisi oluşturma adımını tamamen devre dışı bırakarak, tasarımı daha hızlı konuma geçmiştir; 45 FPS'de gerçek zamanlı olarak çalışır ve hızlı bir versiyonu, yani 155 FPS'de çalışan Fast YOLO [216] da mevcuttur. YOLO, tahminler yaparken görüntünün bütününe gördüğünden, nesne sınıfları hakkındaki bağlamsal bilgileri dolaylı olarak kodlar ve arka plandaki yanlış pozitifleri tahmin etme olasılığı daha düşüktür. YOLO, sınırlayıcı kutu lokasyonunun, ölçeğin ve en-boy oranının kaba ayırımından dolayı daha fazla yerleştirme hatası yapar. [216]'de tartışıldığı üzere, YOLO bazı nesnelere, özellikle küçük olanları yerleştirmede başarısız olabilir, bunun muhtemel sebebi ızgara bölünmesinin oldukça kaba olması ve yapısı gereği her ızgara hücresinin yalnızca bir nesne içerebilmesidir.



Şekil 4.4. YOLO mimari [205].

4.1.3. SSD (Single Shot Detector)

Algılama doğruluğundan çok fazla ödün vermeden gerçek zamanlı hızı korumak için, Liu ve ark. [171], YOLO'dan [216] daha hızlı olan SSD'yi önermiş ve Faster R-CNN

karşılaştırılarak değerlendirilmiştir.

[218]'de düşük kontrastlı ve değişken, gürültülü arka plana sahip uzun evrimli görüntüler için nesne tespitinde yüksek verimliliğe ulaşmak için derin konvolüsyon sinir ağları ile birlikte çalışan, adaptif bulanık tabanlı bir ağ topolojisi önerilmiştir.

[219]'da LiDAR verilerini ve RGBD bulutlanma noktaları kullanılarak doğru ve etkili nesne tespiti için 3D CNN mimarisi VoxNet'i sunulmuştur. Yaklaşımlar kamuya açık son teknoloji ölçütler ile değerlendirilmiş ve bu yaklaşımın, nesnelere gerçek zamanlı sınıflandırırken söz konusu ölçütlerin ötesine geçen bir doğruluk performansı sergilediği bulgulanmıştır.

[166]'da Fast R-CNN olarak adlandırılan bölge tabanlı bir konvolüsyon sinir ağı sunulmuştur. Bu ağ, düşük hesaplama hızıyla ödünleşme yaparken nesnelere yüksek doğruluk düzeyinde algılama yetisine sahiptir. Bu nedenle bu ağ, doğruluk açısından yüksek bir performans sergilemesine rağmen gerçek zamanlı nesne tespiti ve nesne tanıma için uygun kabul edilmez.

[214]'te Fast R-CNN'nin "Faster R-CNN" olarak bilinen güncellenmiş bir versiyonu sunulmuştur. Adından da anlaşılacağı gibi, bölge tabanlı konvolüsyon sinir ağının güncellenmiş versiyonu, önceki versiyona ve diğer son teknoloji ağların birçoğuna göre daha iyi hesaplama hızı ve doğruluk sergilemiştir. Öznitelik üretmek ve bunları nihai tespiti gerçekleştirmekten sorumlu tespit ağı ile paylaşarak ağın hesaplama hızını artıran bir RPN eklenmiştir. Faster R-CNN modelleri, gerçek zamanlı tespit yapma yetisine sahiptir, ancak daha küçük boyutlu nesnelere tespit etmekte zorlanırlar.

Faster R-CNN, büyüklük bakımından Fast R-CNN'den daha hızlı olmasına rağmen, Faster R-CNN ağının ilk ve ikinci aşamaları olan CNN öznitelik çıkarma ve bölge-başına hesaplama maliyeti ağın hızına ket vurmaktadır. Bu sorunu ele alan Kim ve ark., en son teknolojik yenilikleri kullanarak öznitelik çıkarma aşamasında değişiklikler yapmış ve PVANET olarak bilinen daha yeni bir ağ sunmuşlardır [220]. Bu ağ, hesaplama maliyetini azaltırken, birden fazla kategorideki nesnelere muadilleri ile eşit düzeyde bir doğrulukla tespit etme kapasitesine sahiptir.

[221]'de nesne tespiti söz konusu olduğunda üst düzey performans sergileyen mevcut ResNet'i benimserken, R-FCN olarak adlandırılan tam konvolüsyon bir ağ kurulmuştur. Çalışmada nesne tespiti doğruluğunu artırmak amacıyla, Fast R-CNN'deki tam bağlantılı katmanların yerine konuma duyarlı ve uzamsal bilgileri kodlayabilen bir dizi

skor haritası koymuşlardır. Sonuç olarak, R-FCN Faster R-CNN ile benzer doğruluk performansını daha iyi hesaplama hızlarında göstermiştir.

[222]'de HyperNet olarak adlandırılan ve çoklu çıktı katmanlarında tespit yaparak nesnelere çoklu ölçeklerde tespit edebilen bir ağ sunulmuştur. Bu ağ, [34]'te önerilen MS-CNN ile benzerlik gösterir ve nesnelere çoklu ölçeklerde tespit etmek için etkili bir çerçeve sağlar.

[223]'te özel GoogLeNet [102] ağının daha basit Dark-Net19 ile değiştirildiği, YOLO'nun geliştirilmiş bir versiyonu olan YOLOv2'yi önerilmiş ve ayrıca parti normalizasyonu [224], tam bağlantılı katmanların bertaraf edilmesi, k-means algoritması ve çok ölçekli eğitim ile öğrenilen iyi tespit noktası kutularını kullanma gibi mevcut işlerden alınan bir dizi stratejiden faydalanılmıştır. YOLOv2 de, PASCAL VOC ve MS COCO gibi, standart tespit görevlerinde üst düzey performansa ulaşmıştır. Buna ek olarak, Redmon ve Farhadi, çoklu kaynaklardan gelen verileri birleştirmek için ImageNet ve kelime ağacına sahip COCO üzerinde eş zamanlı eğitim yapmak için birleşik bir optimizasyon yöntemi önermek suretiyle, YOLO9000'i sunmuşlardır.

Kavramsal basitlik, verimlilik ve esneklik ruhunu takip eden He ve ark. [169]'da Faster R-CNN'i genişleterek, piksel bazlı nesne örneği bölütlemesi sorununun üstesinden gelmek için Mask RCNN'i önermişlerdir. Maske RCNN, aynı ilk aşamaya RPN sahip iki aşamalı veri hattını izler. İkinci aşamada, sınıfı ve kutunun göreceli konumunu tahmin etmenin paralelinde, Maske RCNN her bir ilgi bölgesi için ikili maske üreten bir dal ekler. Yeni dal, CNN öznitelik haritasının en tepesinde bulunan FCN'dir [126]. Orijinal ilgi bölgesi (RoI) havuzlama (RoIPool) katmanının neden olduğu yanlış hizalamalardan kaçınmak adına, piksel seviyesinde uzamsal örtüşmeyi korumak için ilgi bölgesi hizalama (RoIAlign) katmanı önerilmiştir. Destek ağı ResNeXt101-FPN [225][226] ile, Mask RCNN COCO'da nesne örneği bölütlemesi ve sınırlayıcı kutu nesne tespiti açısından en iyi sonuçları elde etmiştir. Eğitilmesi kolaydır, iyi düzeyde genelleme yapar ve 5 FPS'de çalışan Faster RCNN'ye yalnızca küçük bir ek yük getirir [169].

Light-Head RCNN: RFCN'nin tespit hızını daha da artırmak için [221], Li ve ark., [227]'de light-Head RCNN'yi önermiştir. Özellikle, Li ve ark., küçük kanal sayısına sahip ince öznitelik haritaları ve düşük hesaplama maliyetli RCNN alt ağı üretebilmek için büyük çekirdekte ayrılabilir konvolüsyon uygulayarak mükemmel bir ödünleşme hızına ve doğruluğuna ulaşmışlardır.

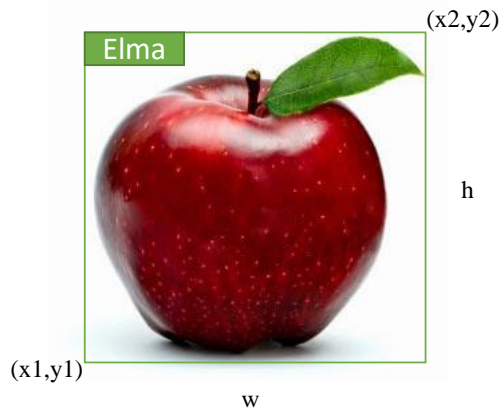
CNN tarafından elde edilen ıgır aıcı grnt sınıflandırma sonularından ve el yapımı znetelikler iin blge nerisinde seici aramanın bařarısından esinlenen Girshick ve ark., [228]’de genelleiyici nesne tespiti iin CNN’yi ilk keřfedenler arasındadırlar ve AlexNet’i [8] blge nerisi ynteminde seici aramayla entegre eden RCNN’yi [166], [229] geliřtirmiřlerdir.

4.2. VERİ ETİKETLEME

Veri etiketleme, denetlenen bir makine ğrenimi grevinde en nemli adımdır. Eđitim verilerinin kalitesinin modelin kalitesini belirlemek olduđu yaygın bir makine ğrenme topluluđunda kullanılan en nemli kavram olarak “Garbage In Garbage Out” řeklinde ifade edilmektedir. Yani eđer eđitim iin hazırlanan veriniz kt ise bu modelinizin sonucunun da kt olacađı anlamına gelmektedir. Aynı ifade veri etiketleme iin kullanılacak ek aıklamalar iin de geerlidir. Yani bir ocuđa bir domates gsterip onun bir patates olduđunu sylerseniz, ocuk bir daha ki seferde grdđ bir domatesi artık patates olarak sınıflandıracaktır. Bir makine ğrenimi modeli, rneklere bakarak benzer řekilde ğrenirken, modelin sonucu, eđitim ařamasında beslediđimiz etiketlere bađlıdır. Veri etiketleme ok fazla manuel alıřma gerektiren bir grevdir. Bir nesne sınıflandırma model alıřmasındaki en nemli ve en yorucu blm oluřturmaktadır. Grnt etiketleme veya grnt aıklama , bir grntdeki farklı birimleri tanımlama veya tanıma srecidir. Bu sre, grntleri bilgisayar grř iin okunabilir hale getirilmesine yardımcı olur. Bilgisayarla gr iin en yaygın grnt aıklama trlerinden bazıları sınırlayıcı kutular, okgen blmlleme, anlamsal blmlleme, 3D kpler, anahtar nokta ve dnm noktası aıklama ve izgi ve spline’ler olarak sıralanabilirler [230].

4.2.1. Sınırlayıcı Kutular

Sınırlayıcı kutular, bilgisayarla grmede en yaygın kullanılan aıklama trdr. Sınırlayıcı kutular, hedef nesnenin konumunu tanımlamak iin kullanılan dikdrtgen kutulardır. Sol st kředeki x ve y eksen koordinatları ve dikdrtgenin sađ alt křesindeki x ve y eksen koordinatları ile belirlenebilirler. rnek gsterimi řekil 4.6’da verilen sınırlayıcı kutular nesne algılama ve yerelleřtirme grevlerinde kullanılır. Sınırlayıcı kutular genellikle sınırlayıcı kutunun iki koordinatı (x_1, y_1) ve (x_2, y_2) veya bir koordinatı (x_1, y_1) ve geniřliđi (w) ve yksekliđi (h) ile temsil edilirler [230].



Şekil 4.6. Sınırlayıcı kutu örnek gösterim [230].

4.2.2. Çokgen Bölümleme (Segmentasyon)

Nesneler her zaman dikdörtgen şeklinde değildirler. Bu fikirle, çokgen bölümlenmeler, nesnenin şeklini ve konumunu çok daha kesin bir şekilde tanımlamak için dikdörtgenler yerine Şekil 4.7’de görüldüğü gibi karmaşık çokgenlerin kullanıldığı başka bir veri açıklama türü ile etiketlenebilirler [207].



Şekil 4.7. Çokgen segmentasyon gösterim [204], [231].

4.2.3. Anlamsal Bölümleme

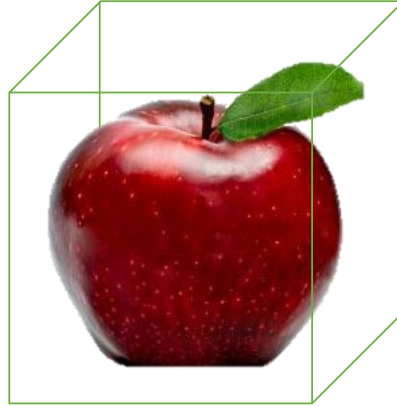
Anlamsal bölümleme, Şekil 4.8’de görüldüğü gibi her pikselin bir sınıfa atandığı piksel bazlı bir etiketleme türüdür. Bu sınıflar yaya, araba, otobüs, yol, kaldırım vb. olabilirler. Her piksel bir anlam taşımaktadır. Anlamsal Bölümleme, öncelikle çevresel bağlamın çok önemli olduğu tüm nesnelerin bir ifade karşılığının olabileceği otonom araç sistemlerinde ve robotik gibi alanlarda çalıştıkları ortamı daha net anlamaları için kullanılırlar [207], [234].



Şekil 4.8. Anlamsal bölümlenme örnek gösterim [204], [231].

4.2.4. 3D Küpoidler

3D küpler, nesne hakkında ek derinlik bilgileri içeren sınırlayıcı kutulara benzer etiketleme türleridir. 3D küpoid etiketleme yöntemi ile nesnenin bir alanda hacim ve konum gibi özellikleri ayırt etmesini sağlayarak 3D temsilini elde edilebilir. Özellikle otonom sistemlerde nesnelerin araçlardan uzaklık mesafelerini ölçmek için derinlik mesafelerine ihtiyaç duyulan durumlarda kullanımı oldukça yaygındır [230].



Şekil 4.9. 3D Küpoid örnek gösterim [230].

4.2.5. Anahtar Nokta ve Yer İşareti

Görüntü üzerinde noktalar oluşturarak küçük nesnelere ve şekil farklılıklarını tespit etmek için kullanılırlar. Bu tür etiketleme, yüz özelliklerini, yüz ifadelerini, duyguları, insan vücudunun parçalarını ve pozlarını algılamak için oldukça kullanışlıdır [230].



Şekil 4.10. Anahtar nokta ve yer işareti örnek gösterim [230].

4.2.6. Çizgiler ve Spline'lar

Adından da anlaşılacağı gibi, bu tür etiketleme modeli, çizgiler ve spline'lar kullanılarak oluşturulurlar. Otonom araçlarda şerit algılama ve tanıma için yaygın olarak kullanılırlar [230].



Şekil 4.11. Çizgiler ve Spline'lar örnek gösterim [230].

4.2.7. Görüntü Ek Açıklama Formatları

Görüntü açıklama formatları etiketleme işlemlerinin ardından sistemin eğitim aşamasında etiket ifadelerini sistemin anladığı dilde ifade etme biçimidir. Görüntü ek açıklama olarak tek bir standart format yoktur. En yaygın olarak bilinen iki tür açıklama formatı aşağıda ifade edilmiştir.

4.2.7.1. Pascal VOC

Pascal VOC, açıklamayı XML dosyasında saklar. Etiketlenmiş resim verisi ile aynı ismi taşıyan bir .xml dosyası etiketleme işlemi ardından oluşturulur. Bu dosya içeriğinde etiketlenen nesneye ait sınıf etiketini, koordinatlarını, yükseklik ve genişlik bilgilerini saklamaktadır. nesne algılama için bir Pascal VOC açıklama dosyası örneği Şekil 4.12'te bulunmaktadır [232].

```

<annotation>
  <folder>Train</folder>
  <filename>arclk_08-09.jpg</filename>
  <path>arclk_08-09.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>1800</width>
    <height>650</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>sat</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <occluded>0</occluded>
    <bndbox>
      <xmin>11</xmin>
      <xmax>12</xmax>
      <ymin>39</ymin>
      <ymax>63</ymax>
    </bndbox>
  </object>
</annotation>

```

Şekil 4.12. Pascal VOC ek açıklama dosya içeriği.

4.2.7.2. YOLO

YOLO etiketleme biçiminde, Pascal VOC'taki gibi aynı dizinde her bir görüntü dosyası için aynı ada sahip bir .txt dosyası oluşturulur. Her .txt dosyası, ilgili görüntü dosyası için ek açıklamalar olarak, nesne sınıfı, nesne koordinatları, nesne yükseklik ve genişlik bilgilerini içerir. Oluşturulan bu ".txt" dosyalarının içeriği formül 4.1'deki yapıdadır.

$$\langle object - class \rangle \langle x - center \rangle \langle y - center \rangle \langle width \rangle \langle height \rangle \quad (4.1)$$

Formüle göre sınıf değeri 0-∞ arasında bir tam sayı değeridir. 2 nesneli bir yapı için sınıf değerleri 0 ve 1 olarak belirlenir. Width ve Height değerleri sınırlar içine alınan nesnenin yükseklik ve genişlik değeri olarak belirlenirken, x-y-center değerleri nesneye ait sınırların merkezini ifade etmektedir. Bu değişkenler 0.0 ile 1,0 arasında değer alırlar. Buna göre etiketlenen bir resim için oluşturulan ".txt" dosya içeriği Şekil 4.13'da gösterildiği gibidir [233].

```

0 0.175722 0.624819 0.027731 0.036284
0 0.292303 0.382438 0.025467 0.184325
1 0.055178 0.360668 0.018676 0.152395
1 0.395303 0.113208 0.024335 0.136430

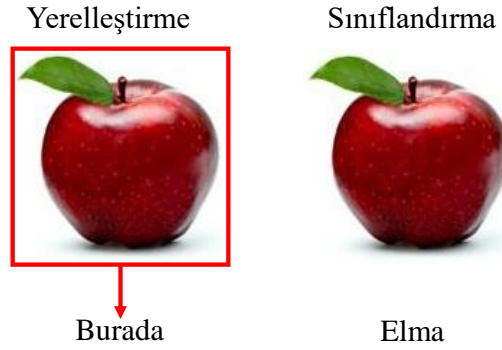
```

Şekil 4.13. YOLO ek açıklama dosya içeriği.

4.3. ALGORİTMALARIN METRİK PERFORMANS ÖLÇÜTLERİ

Nesne algılama ölçümleri, modelin bir nesne algılama görevinde ne kadar iyi performans gösterdiğini değerlendirmek için kullanılır. Ayrıca, birden çok algılama sisteminin nesnel olarak karşılaştırılmasını veya aralarında bir kıyaslama yapılabilmesini sağlar.

Faster R-CNN, MobileNet SSD ve YOLO gibi birçok nesne algılama algoritmaları modellerini değerlendirmek için Ortalama ortalama kesinlik (mAP - Mean Average Precision) metrik değerlendirme yöntemini kullanırlar. mAP bilgisayarla görmede, nesne tespiti yerleştirme ve sınıflandırma hesaplamaları için kullanılan popüler bir değerlendirme metriğidir. Şekil 4.14'deki örnekte de görüleceği üzere yerleştirme bir nesnenin konumunu etrafındaki sınırlayıcı kutunun koordinatlarını belirlerken, sınıflandırma kutu içerisindeki nesnenin sınıfını belirler.




Şekil 4.14. Görüntü sınıflandırma ve yerleştirme.

Bu tarz çalışmalarda metrik değerler ölçülürken hem sınıflandırma performansı hem de görüntüde sınırlayıcı kutuların yerleştirilmesi değerlendirilir. Bu değerlendirme işlemi için belirtilen görüntü üzerinde kaç nesnenin doğru kaç nesnenin yanlış algılandığının belirlenmesi gerekmektedir. Bu değerlendirme işlemi IoU (Birlik Üzerinden Kesişim) metriği ile gerçekleştirilmektedir.

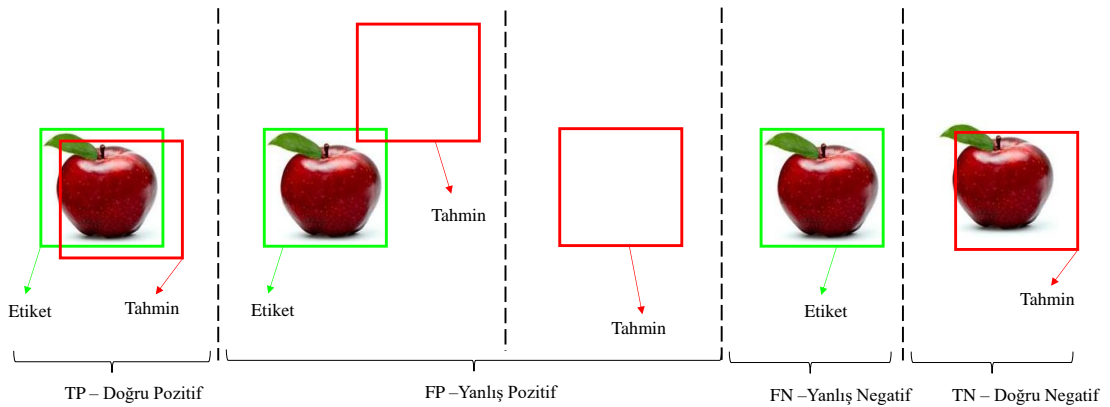
IoU, öngörülen kutunun ne kadar iyi olduğunu değerlendirmek için yer gerçeği sınırlayıcı kutusu (test veri seti kümesindeki sınırlayıcı kutularla belirtilmiş etiketler) ve öngörülen sınırlayıcı kutu arasındaki benzerliği ölçen bir değerlendirme ölçüsüdür. Diğer bir deyişle algılama sınırlayıcı kutusu ile yer gerçeği sınırlayıcı kutusunun birleşimi üzerindeki kesişim olarak tanımlanır. Formül gösterimi Şekil 4.15'de verilmiştir. IoU puanı 0 ile 1 arasında değişir, iki kutu ne kadar yakınsa IoU puanı o

kadar yüksek olur. PASCAL VOC veri seti için, 0.5 IoU eşiği olan bir metrik değer kullanılırken, MS COCO, IoU eşikleri için (0.5, 0.55,, 0.8, 0.85, 0.9, 0.95) 0.05 adımında, farklı metrik değerlerini kullanmaktadır.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Şekil 4.15. IoU hesaplama örneği [234].

Her bir nesne tespitini sınıflandırmalara dönüştürmek için genel kabul olarak 0.5 gerçek değerli bir eşik değeri belirlenir. Bu eşik değerine göre tespit edilen nesnenin $IoU \geq 0.5$ ise, nesne algılama *Doğru Pozitif (TP - True Positives)* olarak sınıflandırılırken, $IoU < 0.5$ ise bu yanlış bir tespittir ve *Yanlış Pozitif (FP - False Positive)* olarak sınıflandırılır. Görüntüde bir etiketleme mevcut olduğunda ve model nesneyi tespit edemediğinde, bu durum *Yanlış Negatif (FN - False Negatives)* olarak sınıflandırılır. İfadelere ait görseller Şekil 4.16’de verilmiştir.



Şekil 4.16. Etiket ve tahmin ilişkisi (IoU).

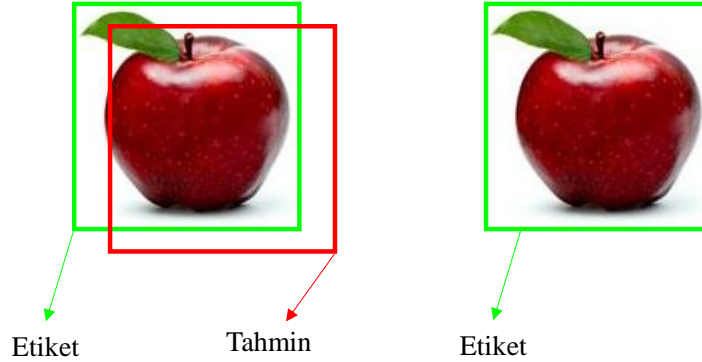
Bu hesaplamalarında ardından hassasiyet (*Precision*) ve geri çağırma (*Recall*) hesaplanması gerekmektedir. *Precision* tahminlerin ne kadar doğru olduğunu ölçerken, *Recall* modelin tahmin etmesini gerekenlerin ne kadarını tahmin ettiğini ölçmektedir.

Bu hesaplama yöntemlerine ait denklemler 4.2 ve 4.3'te verilmiştir.

$$Precision = \frac{TP}{TP+FP} \quad (4.2)$$

$$Recall = \frac{TP}{TP+FN} \quad (4.3)$$

Bu ifadelere göre Şekil 4.17'da verilen örnek bir tahminleme görüntüsü için mAP hesaplaması gerçekleştirilmiştir.



Şekil 4.17. Nesne algılama örneği.

Şekilde iki adet etiketlenmiş nesne olup model sadece bir etiketli nesneyi tahmin etmiştir. Bu durumda $TP=1$ olarak hesaplanır. Diğer taraftan etiketli olan bir nesnenin model tarafından tahmin edilememesi durumu $FP=0$ olarak hesaplanır. Bu göre;

$Precision = \frac{TP}{TP+FP} = \frac{1}{1+0} = 1$ olarak ifade edilir. Bu görüntü için tek bir değere sahip olduğundan hassasiyet ortalaması da 1 olacaktır. Hesaplanması gereken diğer önemli ifade de $Recall$ 'dur. $Recall$ için $TP=1$, $FP=0$ iken $FN=1$ 'dir. Buna göre;

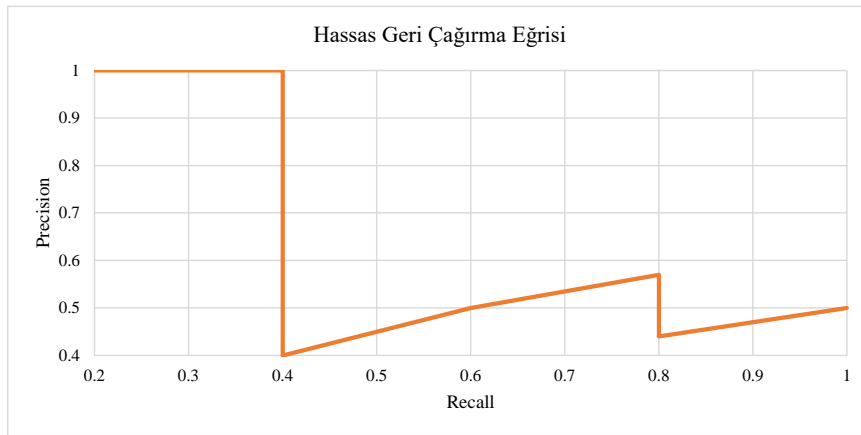
$$Recall = \frac{TP}{TP+FN} = \frac{1}{1+1} = 0.5 \text{ olarak ifade edilir.}$$

mAP puanı, mevcut farklı algılama zorluklarına bağlı olarak tüm sınıflar ve / veya genel IoU eşikleri üzerinden ortalama AP alınarak hesaplanır. AP hesaplanmasını göstermek için aşırı basitleştirilmiş bir örnek için etiketlenmiş 5 adet elma görüntüsü olsun. Görseldeki elmaların etiketleri üzerindeki tahminleri sıra ile doğru/yanlış olarak Çizelge 4.1'de verilerek gerekli hesaplamalar çizelgede sunulmuştur.

Çizelge 4.1. AP hesaplama tablosu.

Sıra (Etiket)	Tahmin (Doğru/Yanlış)	Precision	Recall
1	Doğru	1/1=1.0	1/5=0.2
2	Doğru	2/2=1.0	2/5=0.4
3	Yanlış	2/3=0.67	2/5=0.4
4	Yanlış	2/4=0.5	2/5=0.4
5	Yanlış	2/5=0.4	2/5=0.4
6	Doğru	3/6=0.5	3/5=0.6
7	Doğru	4/7=0.57	4/5=0.8
8	Yanlış	4/8=0.5	4/5=0.8
9	Yanlış	4/9=0.44	4/5=0.8
10	Doğru	5/10=0.5	5/5=1.0

Çizelgeye göre üçüncü satırın örnek olarak hesaplanmasında *Precision* ve *Recall* değerleri sırası ile $2/3=0.67$ ve $2/5=0,4$ olarak hesaplandığı görülebilir. Tahmin sıralamasında aşağıya inildikçe *Recall* değerleri artmaktadır. Bu sonuçlara göre *Precision* bir zig-zag grafik modeli oluşturur. Grafiğe ait görünüm Şekil 4.18'de verilmiştir.



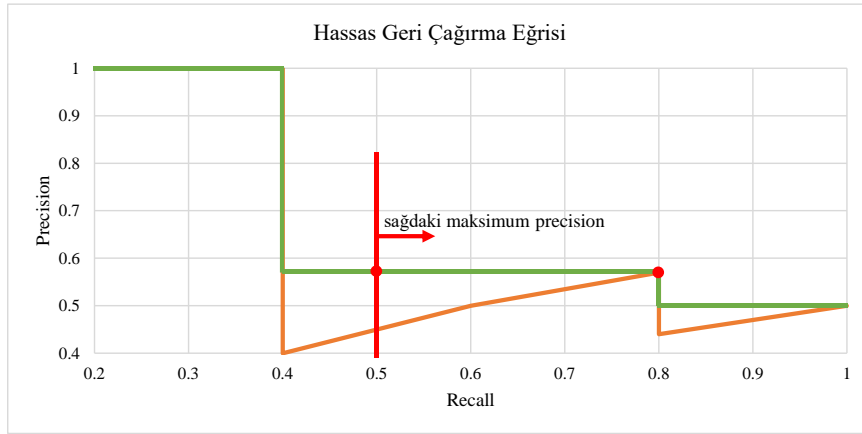
Şekil 4.18. Hassas geri çağırma eğrisi.

AP, verilen grafik eğrisi altındaki alanın hesaplanması ile bulunur. Bu hesaplama ait formül denklem 4.4'te verilmiştir.

$$AP = \int_0^1 p(r) dr \quad (4.4)$$

Bu hesaplama gerçekleştirilmeden önce grafik üzerindeki zig-zag desenlerde düzenleme yapılır. Bu işlem grafik üzerinde, her geri çağırma seviyesinde, her bir hassasiyet değerini, o geri çağırma seviyesinin sağındaki maksimum hassasiyet değeriyle

değiştirilir. Buna göre oluşan yeni grafik görünüm Şekil 4.19’de verilmiştir.



Şekil 4.19. Düzeltilmiş hassas geri çağırma eğrisi.

Bu işlem adımıyla turuncu çizgiler yeşil çizgilere dönüştürülerek eğri, zig-zag desen yerine basamak haline getirilmiştir. Böylece AP hesaplaması ufak hatalar göz ardı edilerek daha basit ve net olarak gerçekleştirilebilecektir. Nesne algılama için mAP, tüm sınıflar için hesaplanan AP'nin ortalamasıdır. Bu işlemler tamamen model algoritmalar tarafından yürütülerek hesaplanmaktadır.

4.4. ALGORİTMALARIN EĞİTİMİ İÇİN DONANIM SEÇİMİ

Bir derin öğrenme algoritmasının içerebileceği parametrelerin boyutu ve çokluğu göz önüne alındığında, eğitim genellikle çok uzun sürer ve bu sebeple donanım seçimi son derece önemlidir. Bu sebeple bu bölümde, mevcut seçeneklere ışık tutmayı amaçlamakta ve CPU ile GPU karşılaştırması yapılmaktadır [111].

4.4.1. Merkezi İşlem Birimi (MIB)

Geleneksel olarak, sinir ağları öncelikle bir bilgisayarın Merkezi İşlem Birimi (CPU) üzerinde eğitilmiştir. CPU genellikle bir bilgisayarın “beyni” olarak tanımlanır ve ona gönderilen hesaplamaları sıralı olarak gerçekleştirme işlevi görür. Bu, bir bilgisayarın temel çalışma prensibini oluşturur ve bir CPU hesaplamalarını ne kadar hızlı gerçekleştirirse, verilen görevleri o kadar hızlı tamamlar. Bazen bir programın birbirinden bağımsız olarak hesaplanabilecek farklı görevleri olur. Tüm görevleri bitirmek ve gereken süreyi optimize etmek için, pek çok CPU’nun paralel olarak hesaplamalar yapabilen birden fazla çekirdeği vardır. Bu, görevlerin daha hızlı bitirilmesini sağlar, çünkü CPU’lar tek bir çekirdeğin boşluğunu beklemek zorunda

kalmazlar [111].

4.4.2. Grafik İşlem Birimi (GIB)

Grafik İşlem Birimi (GPU-Graphics Processing Unit), paralel hesaplamada mükemmel performans sergileyen, özel bir işlemci türüdür. Günümüz bilgisayarlarındaki CPU'lar genellikle bir ila 16 çekirdeğe sahipken, ileri teknoloji sunucu CPU'ların çekirdek sayısı 32 ye kadar çıkabilmektedir. Birinci sınıf GPU'larda binlerce çekirdek olduğu için, GPU'lar bu sayıları gölgede bırakmaktadır. GPU'lar, sıralı işlemlerde CPU muadillerine göre daha yavaştır, ancak paralel olarak yürütülebilen görevler söz konusu olduğunda çok başarılıdır. Bu, bellekte saklanan bir 3D peyzajın ekranda gösterilebilmesi için 2D bir görüntüye yansıtılmasını gerektiren 3D grafiklerde çok geçerlidir. Adından da anlaşılacağı gibi, GPU'lar ilk olarak 3D grafikleri görüntülemek üzere tasarlanmışlardır ve hala öyledirler [111].

Bir derin öğrenme algoritması eğitiminde gerekli işlemler paralel olarak gerçekleştirilebileceğinden, eğitimi yalnızca CPU kullanılan durumlara kıyasla birkaç kat daha hızlı hale getirdikleri için GPU'lar oldukça değerli araçlar haline gelmişlerdir. Ancak bu, GPU'yu farklı kodlar çalıştıracak şekilde programlama yetisi gerektirir ve bunun üreticiler tarafından desteklenmesi gerekir. Bugün birinci sınıf grafik kartı pazarında, gerçekçi olmak gerekirse sadece iki büyük üretici vardır ve bunlar AMD ve NVIDIA firmalarıdır. Bu ikisinden NVIDIA, kodun GPU'larında çalıştırılmasına izin vermek için tasarlanan CUDA dilinin geliştirilmesine oldukça çok yatırım yapmıştır. Alternatif olarak, birçok üreticinin en yaygın işlemci tiplerinde paralel hesaplama için kullanılan OpenCL kullanmak mümkündür. OpenCL, açık standart yazılım programlama arayüzleri (API) geliştiren ve kâr amacı gütmeyen bir birlik olan Khronos Grubu tarafından idame ettirilmektedir [235]. OpenCL genele uygulanabilirliği sebebiyle çok tercih edilirken, CUDA daha uzun süredir piyasada olup makine öğrenmesi yazılım desteğinde daha ileridedir. Bu sebeple NVIDIA kartları, makine öğrenmesi algoritmalarıyla çalışırken açık ara en yaygın kullanılan grafik kartı haline gelmiştir [236]. Son yıllarda artan ilgi göz önüne alındığında, birçok oyuncu bu yeni büyüyen pazarın bir parçası olma hedefiyle çalışmaktadır. Diğer taraftan bezer şekilde Intel, makine öğrenmesinde daha iyi performans gösterebilmek için sunucu CPU'larına ince ayar yapmaktadır [237]. Google, makine öğrenmesi görevlerini enerji tasarruflu bir şekilde yerine getirecek bir çip geliştirmektedir [238]. AMD, makine öğrenmesi için

özel olarak tasarlanan yeni bir GPU serisi ile birlikte daha iyi performans sağlayacak yazılım araçlarının piyasaya sürüleceğini duyurmuştur [239]. Bu, makine öğrenmesi için uygun donanım seçeneklerinin hızla genişlediğini ve gelecekte hangi seçimin en iyi olacağına dair spekülasyonların çıkacağına işaret etmektedir.

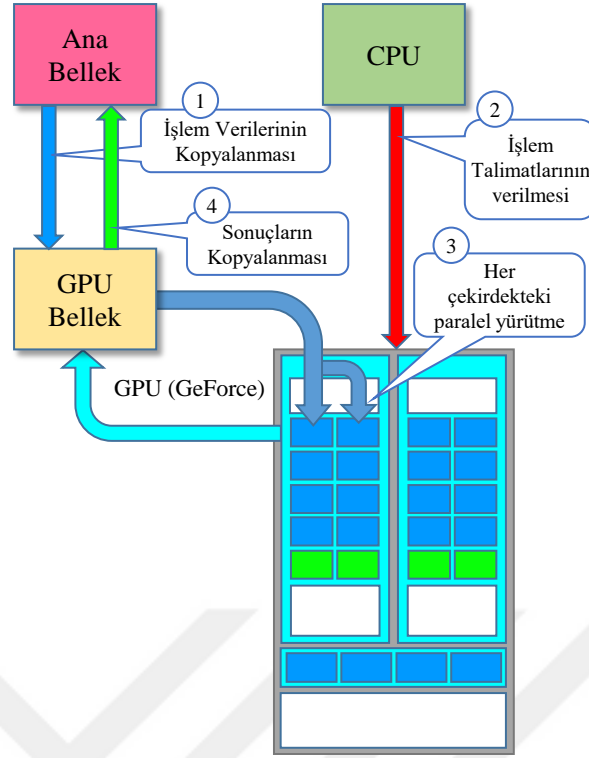
4.5. ALGORİTMALARIN EĞİTİMİNDE KULLANILAN KÜTÜPHANELER

Makine öğrenmesi algoritmalarını uygulama sürecini basitleştirmek için çeşitli yazılım kütüphaneleri geliştirilmiştir. Bu kütüphaneler işlevsel bir sinir ağı kurmak için gereken görevlerin çoğunu otomatik olarak yürüten araçlar sağlarlar. Kütüphaneler aynı teorik makine öğrenmesi modellerine dayanmakla birlikte, bunların nasıl uygulanacağı konusundaki yaklaşımlarında farklılık gösterirler. Aşağıda, bu çalışmada ihtiyaç duyulan ve kullanılan kütüphanelerin kısa özetleri verilmiştir.

4.5.1. CUDA

CUDA (Compute Unified Device Architecture), Nvidia tarafından geliştirilen bir paralel hesaplama platformu ve uygulama programlama arayüzü (API) modelidir [240]. Yazılım geliştiricilerin ve yazılım mühendislerinin genel amaçlı işleme GPGPU (Grafik İşleme Birimlerinde Genel Amaçlı Hesaplama) olarak adlandırılan bir yaklaşım için CUDA etkin bir grafik işleme birimi (GPU) kullanmalarına olanak tanır. CUDA platformu, hesaplama çekirdeklerinin çalışması için GPU'nun sanal komut kümesine ve paralel hesaplama öğelerine doğrudan erişim sağlayan bir yazılım katmanıdır [241].

CUDA platformu, C, C ++ ve Python gibi programlama dilleriyle çalışacak şekilde tasarlanmıştır. Bu erişilebilirlik, paralel programlama yapan uzmanların GPU kaynaklarını kullanmalarını, grafik programlamada ileri düzey beceriler gerektiren Direct3D ve OpenGL, OpenCV gibi daha önceki API'lere kıyasla kolaylaştırmaktadır [242].



Şekil 4.20. CUDA işlem akış diyagramı [241].

4.5.2. cuDNN

NVIDIA CUDA® Derin Sinir Ağı kütüphanesi (cuDNN), derin sinir ağlarının temelleri için GPU hızlandırmalı bir kütüphanedir. cuDNN, ileri ve geri konvüsyon, havuzlama, normalleştirme ve aktivasyon katmanları gibi standart görevler için yüksek düzeyde ayarlanmış uygulamalar sağlar.

Dünyanın her yerindeki derin öğrenme araştırmacıları ve çerçeve geliştiriciler, yüksek performanslı GPU hızlandırma için cuDNN'e güvenmektedir. Bu onların, düşük seviyeli GPU performans ayarına zaman harcamak yerine sinir ağlarını eğitmeye ve yazılım uygulamaları geliştirmeye odaklanmalarını sağlamaktadır. cuDNN, Caffe2, Chainer, Keras, MATLAB, MxNet, PyTorch ve TensorFlow gibi yaygın olarak kullanılan derin öğrenme çerçevelerini hızlandırır [243].

4.5.3. Tensorflow

Başlangıçta Google tarafından Google Beyin projesinin bir parçası olarak geliştirilen Tensorflow, 2015'in sonlarında açık kaynak haline getirilmiştir. Tensorflow bir Python kütüphanesidir ve büyük bir şirket tarafından sıfırdan geliştirilen tek büyük kütüphane olarak öne çıkar; diğer kütüphaneler araştırma camiası tarafından geliştirilmiştir.

Tensorflow, kullanıcının öğrenme oranı, model ağırlıkları, kayıp fonksiyonları vb. gibi şeyleri görselleştirerek kolaylıkla grafikler oluşturmaya izin veren, TensorBoard gibi bir takım entegre araçlarına sahiptir. Tensorflow, aynı zamanda iş yükünü sadece aynı cihazdaki GPU'lara değil, birbirine bağlı pek çok cihaza dağıtarak hesaplamada büyük avantaj sağlayan tek kütüphanedir [244].

4.5.4. OpenCV

OpenCV (Açık Kaynak Kodlu Bilgisayarla Görme Kütüphanesi), açık kaynak kodlu bir bilgisayarla görme ve makine öğrenmesi yazılımı kütüphanesidir. OpenCV, bilgisayarla görme uygulamaları için ortak bir altyapı sağlamak ve ticari ürünlerde makine algısının kullanımını hızlandırmak için oluşturulmuştur [245].

Kütüphane hem klasik hem de son teknoloji bilgisayarla görme ve makine öğrenmesi algoritmalarının kapsamlı bir setini içeren 2500'den fazla, optimize edilmiş algoritmaya sahiptir. Bu algoritmalar yüzleri saptamak ve tanımak, nesnelere tanımlamak, videolardaki insan eylemlerini sınıflandırmak, kamera hareketlerini izlemek, hareketli nesnelere izlemek, nesnelere 3D modellerini çıkarmak, stereo kameralardan 3D bulutsallık noktaları üretmek, bütün bir sahnenin yüksek çözünürlüklü görüntüsünü elde etmek için görüntüleri birbirine iliştmek, görüntü veri bankasından benzer görüntüleri bulmak, flaş kullanarak çekilen görüntülerde kırmızı gözü gidermek, göz hareketlerini izlemek, sahne tanımak ve onu artırılmış gerçeklik ile bindirmek için imleyici oluşturmak vb. için kullanılabilir [246], [247].

Kütüphane C ++, Python, Java ve MATLAB arayüzlerine sahiptir ve Windows, Linux, Android ve Mac OS'yi desteklemektedir. OpenCV çoğunlukla gerçek zamanlı görüntü uygulamalarına eğilmekte ve mümkün olduğunda MMX ve SSE direktiflerinden yararlanmaktadır. Sınırsız özellikli CUDA ve OpenCL arayüzleri halihazırda geliştirilmektedir. Bu algoritmaları oluşturan veya destekleyen 500'den fazla algoritma ve bunun yaklaşık 10 katından daha fazla fonksiyon vardır. OpenCV aslen C ++ ile yazılmıştır ve STL konteynerleri ile sorunsuz çalışan bir arayüz şablonuna sahiptir [246].

4.5.5. NumPy

NumPy Python programlama dili için oluşturulan bir kütüphanedir. Büyük, çok boyutlu diziler ve matrislerin yanı sıra geniş bir yüksek seviye matematiksel fonksiyon

koleksiyonuna bu diziler üzerinde çalışabilmeleri için ek destek verir. NumPy'nin atası Numeric, aslen Jim Hugunin tarafından diğer geliştiricilerin katkılarıyla geliştirilmiştir. 2005 yılında, Travis Oliphant rakip Numarray'in özelliklerini kapsamlı değişikliklerle Numeric'e dahil etmek suretiyle NumPy'yi geliştirmiştir. NumPy açık kaynak kodlu bir yazılımdır ve birçok kişinin katkısıyla meydana gelmiştir [248].

4.5.6. Matplotlib

Matplotlib, Python programlama dili ve sayısal matematik uzantısı NumPy için bir grafik kütüphanesidir. Grafikleri Tkinter, wxPython, Qt veya GTK + gibi genel amaçlı GUI araç takımlarını kullanarak uygulamalara gömmek için nesne yönelimli bir API sağlar. Ayrıca, MATLAB'inkine çok benzer şekilde tasarlanmış bir sonlu otomata (OpenGL gibi) dayalı, yöntemsel bir "pylab" arayüzü vardır [249], [250].

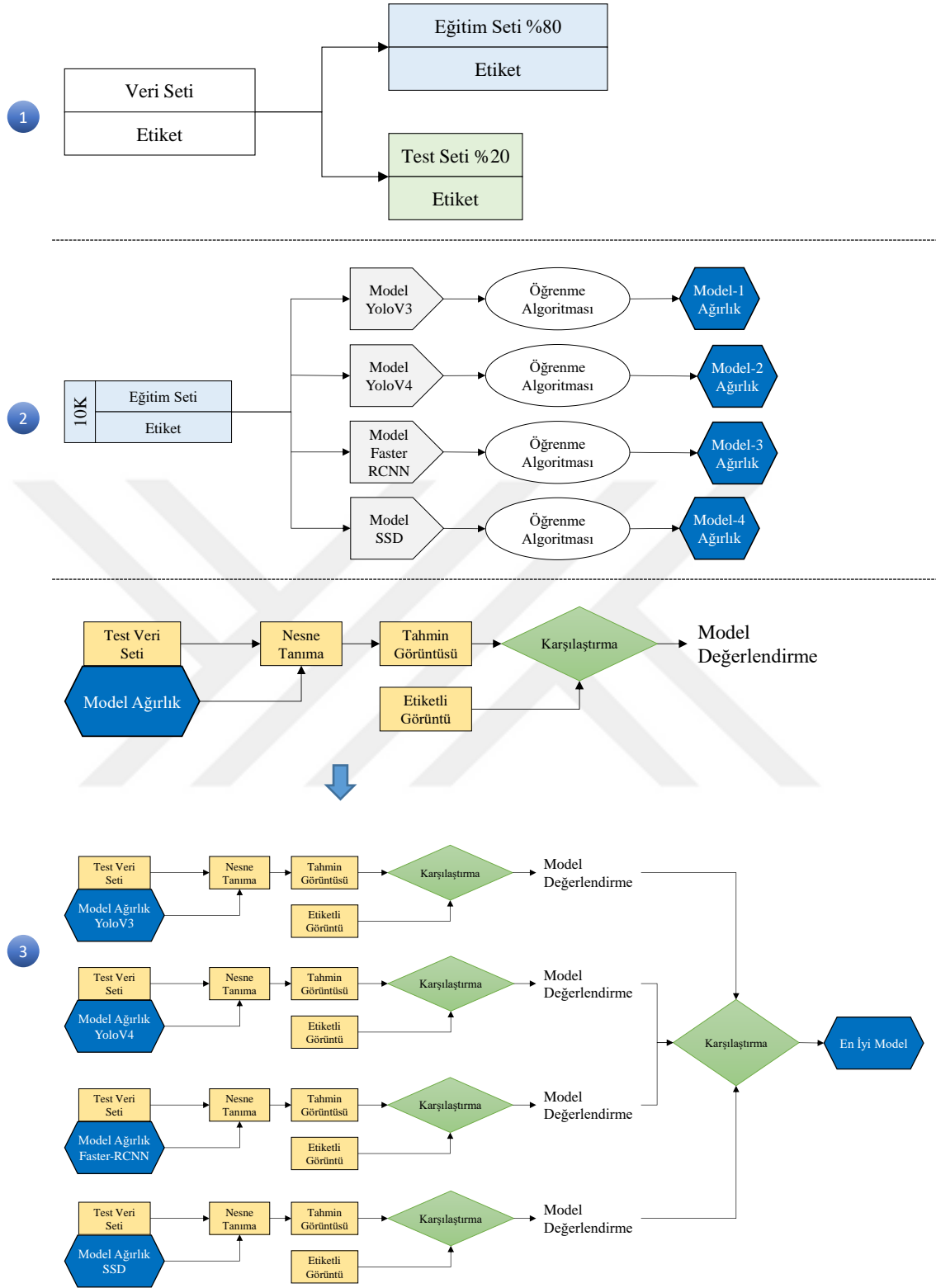


5. NESNE TESPİT ALGORİTMALARI İLE HİSSE SENEDİ AL-SAT KARAR DESTEK MODELİNİN OLUŞTURULMASI

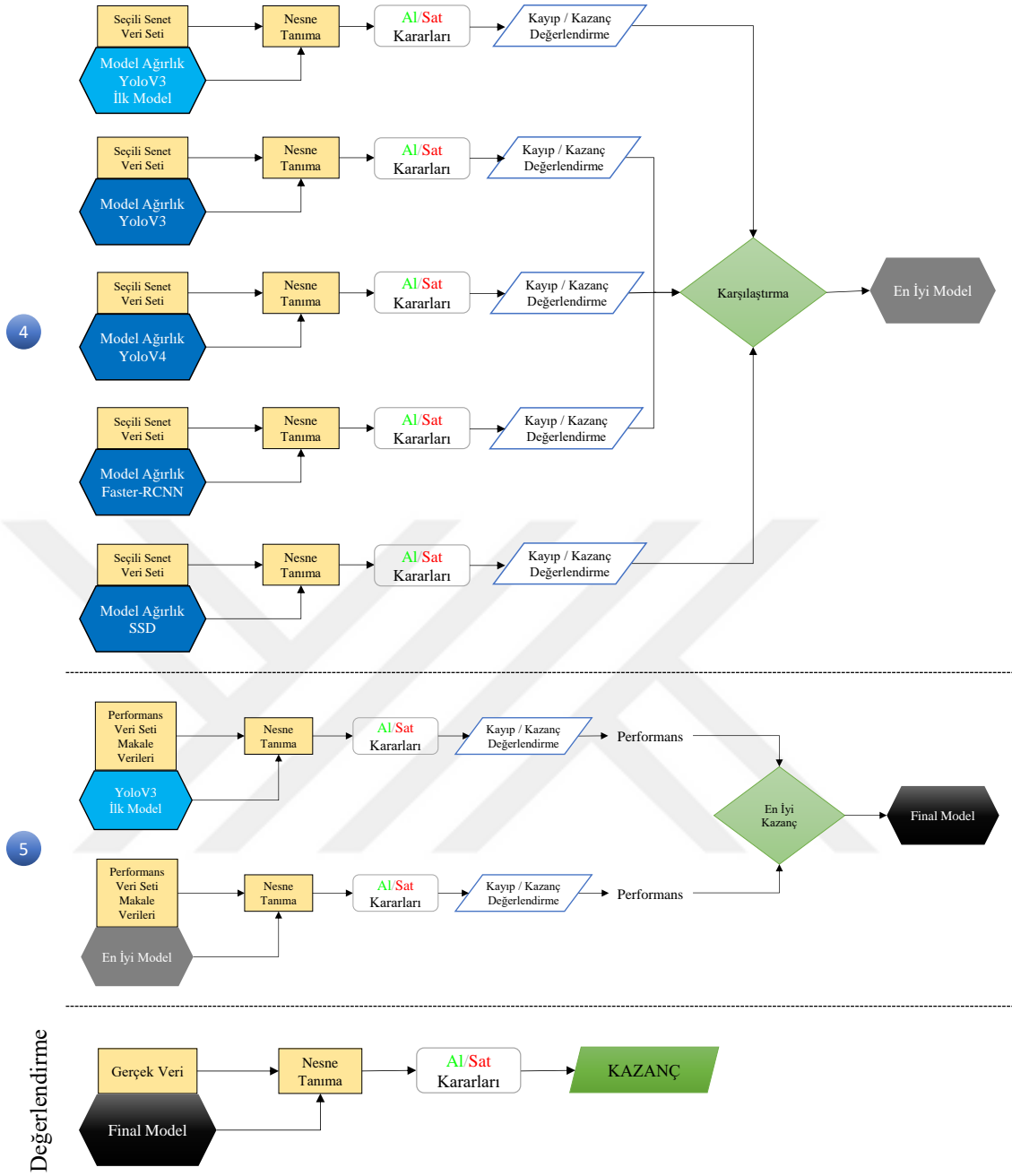
5.1. ÖNERİLEN MODEL

Önerilen model, alternatif finansal piyasa araçlarına ait mum grafikleri üzerinde yatırımcılara teknik ve temel analiz tekniklerinin verdiği alım satım kararlarını destekleyecek nitelikte bir “*Al-Sat*” karar modelidir. Bu model yukarıda anlatılan nesne tanıma algoritmaları ve finansal tahmin yöntemlerinden mum grafikleri yöntemlerinin bir araya getirilerek oluşturulmuş bir model yaklaşımıdır. Bu model tasarımına ait akış diyagramı Şekil 5.1 ve Şekil 5.2’de adım adım verilmiştir.





Şekil 5.1. Çalışmaya ait akış diyagramı -1.



Şekil 5.2. Çalışmaya ait akış diyagramı-2.

Akış diyagramına ait aşamalar başlıklar halinde çalışmanın devamında sunulmuştur.

5.2. VERİ HAZIRLAMA

5.2.1. Veri Seti Oluşturma

Şekil 5.1 akış diyagramı ilk aşamasında belirtildiği üzere önerilen modellerin eğitim ve test işlemleri için bir veri setine ihtiyaç duyulmaktadır. İhtiyaç duyulan bu veri seti için

BIST'e kayıtlı güncel hisse senetlerine ait fiyat bilgilerinin 2D mum grafik görüntüleri kullanılmıştır. Mum grafiklerinin oluşumu için finansal veriler akademik tabanlı çalışmalar kapsamında BIST'ten talep edilmiş, aynı zamanda güncel grafikler üzerinde test çalışmaları yapmak adına açık kaynak olarak *Yahoo.Finans* üzerinden güncel fiyat bilgilerine erişim sağlanmıştır. Elde edilen veriler .csv dosya türünde olup günlük açılış, en düşük, en yüksek ve kapanış fiyat verilerinden oluşmaktadır. Bu veriler 2D mum grafik görünümüne dönüştürülmüştür. Elde edilen grafik gösterime ait örnek Şekil 5.3'te verilmiştir.



Şekil 5.3. Bir hisse senedine ait bir yıllık 2D mum grafik.

Spekülatif ve manipülatif grafiklere sahip hisseler dışında herhangi bir endeks (BIST-30, BIST-50, vb.) ayrımı yapılmaksızın bir sıra halinde 2000-2018 yılına ait hisse senetlerinin görsel 2D mum grafikleri yıllık olarak *hisseadi_ay_yil* ismi ile 1800x650 boyutunda resim formatında kaydedilerek veri seti oluşturulmuştur. Veri setine dâhil edilen senetlerin bazılarının 2000 yılı başlangıcı olmadığından hangi yılda BIST'te işlem görmeye başlamış ise grafik görüntüsü o yıl itibari ile kaydedilmiştir. Buradaki farklı senetlerin kullanılarak veri seti oluşturulmasındaki amaç, farklı eğilim ve mum grafik görünümlerine sahip senetlerin zamansal davranışlarının etiket varyasyonumuzu arttıracakı düşüncesindedir. Gerçekleştirilen çalışma için ilk olarak 550 adet 2D mum grafik görünümü Şekil 5.3'te verildiği gibi bir resim üzerinde 330-370 adet günlük mum grafiği temsil edecek şekilde kaydedilmiştir. İlk aşamada hazırlanan VG-1, hedeflenen sonuçların alınabilmesi adına tek algoritma ile eğitilmiş ve etkileyici sonuçlar alınmıştır [1]. Ardından çalışmayı genişletmek adına veri setine 709 adet 2D mum grafik görüntüsü eklenerek veri seti 1259 adete çıkarılarak VG-2 oluşturulmuştur. Veri seti

(eđitim ve test grafikleri) gn sonu fiyatları sonucunda Őekillenen mum grafik grntlerinden oluŐmaktadır. Bilindiđi zere bu grafikler, yatırım araçlarının seans sonu veya 4 saatlik, 1 saatlik gibi sonuçlarından da elde edilebilmektedir. Burada dikkat edilecek tek kriter model eđitiminde yıllık olarak toplam 360 ($\pm 10\%$) gnlk mum çubuđuna ait grafik kullanıldıđından seans grafik grntsnn kullanılması halinde mum sayısının 180 gne indirilmesi gerekmektedir. Yine 180 gnlk olan seanslık grafik, iŐlem seansının iki seans olmasından dolayı toplamda 360 adet mumdan oluŐan grafik grntsne ulaŐılacaktır.

Őekil 5.3'te grleceđi zere yatırım enstrmanlarına ait grafikler, dođası geređi iniŐli çıkıŐlı yapıları oluŐturan eđilimlere sahiptirler. Bu eđilim yapıları içerisindeki dip-tepe seviyelerini etiketleyerek anlamlı kutucuklar oluŐturmak, çalıŐmanın temel yapısıdır Őeklinde ifade edilebilir.

5.2.2. Veri Etiketleme

Bilgisayarla grme çalıŐmaları için blm 4.2'da da açıklandıđı zere bir dizi grnt etiketleme aracı mevcuttur. Bu çalıŐmada 2D grseller zerinde iŐlemlerin yrtldđ sınırlayıcı kutular zelliđinde çalıŐmaya imkn sađlayan "LabelImg" grnt etiketleme aracı kullanılmıŐtır [251]. LabelImg, en popler grnt etiketleme araçlarından biri olup, Python ile yazılmıŐ açık kaynak kodlu bir yazılımdır. Bu yazılım grafik arayz için Python PyQt kullanır. Etiketlemede oluŐturulan ek açıklamaları XML dosya biçiminde kaydetmek için *Pascal VOC* formatı kullanılırken, *TXT* dosya biçiminde kaydetmek için *YOLO* formatı kullanılmaktadır. Format deđiŐimini Őekil 5.4 zerinde iŐaretili alanda da grlebileceđi gibi bir "click" iŐlemi ile ayarlanabilmektedir.



Şekil 5.4. LabelIMG arayüz görünümü (a).



Şekil 5.5. LabelIMG arayüz görünümü (b).

Etiketleme işlemi gerçekleştirilen her görüntü için ayrı ayrı ek açıklama dosyaları mevcuttur. Çalışma kapsamında etiketlemeler ayrı ayrı format için iki kez etiketlenmemiştir. İlk aşamada 550 görüntü üzerinden etiketlenen 10000 etiket ve ikinci aşamada 1259 adet görüntü üzerinden etiketlenen 20000 etiket YOLO format etiketlenmiş ve ek açıklama dosyaları bir klasör içerisine kaydedilmiştir. Klasör içeriğine ait görünüm Şekil 5.6'te verilmiştir.

> Bu bilgisayar > Yeni Birim (D:) > gunay_tez > data_asil

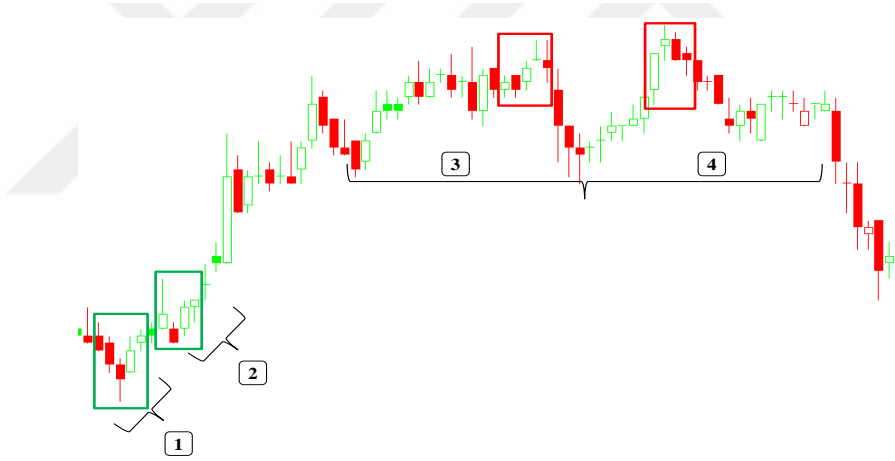
Ad	Tarih	Tür
acsel_13-14.jpg	28.03.2019 00:40	JPG Dosyası
acsel_13-14.txt	28.04.2019 00:57	Metin Belgesi
acsel_14-15.jpg	28.03.2019 00:42	JPG Dosyası
acsel_14-15.txt	28.04.2019 01:00	Metin Belgesi
acsel_15-16.jpg	28.03.2019 00:44	JPG Dosyası
acsel_15-16.txt	28.04.2019 01:05	Metin Belgesi

Şekil 5.6. Etiket klasörü içeriği.

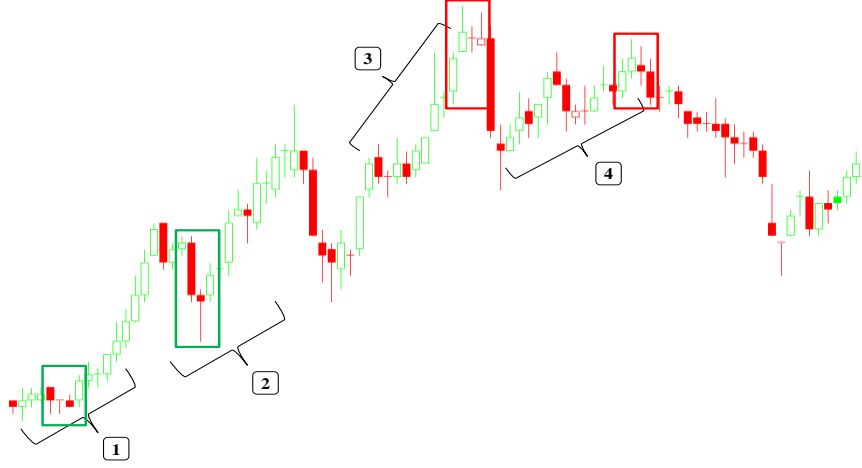
YOLO formatında etiketlemenin tamamlanmasının ardından diğer format olan Pascal VOC için yeniden bir etiketleme işlemi gerçekleştirilmemiştir. Bu tez çalışmasının en zorlu, teferruatlı ve zaman alan bölümü etiketleme gerçekleştirip ek açıklamaların oluşturulması aşamasıdır. Ek açıklamalar algoritmalar için insan gözü ile algılanan nesnelere makine gözü ile algılanmasını sağlayan en önemli bölümdür. Özellikle bu çalışmada nesne tanıma algoritmaları için standart nesnelere dışında bir nesne ile

çalışma gerçekleştirildiğinden etiketlemenin önemi iki kat daha da artmaktadır. Bir kez etiketlendikten sonra oluşturulan ek açıklamalar için aynı şekilde tekrar aynı etiketlemeleri oluşturmak oldukça zordur. Bu sebeple *donusturt.py* [252] adlı kodlama ile YOLO'dan elde edilen ek açıklamalar Pascal VOC ek açıklama türüne dönüştürülmüştür. Bu dönüştürme işlemi gerçekleştirilen kod bloğu EK'ler bölümünde verilmiştir.

Modellerin eğitimi için hazırlanan veri seti etiketleme işlemlerinin tümü bu çalışmada yapılmış olup, etiketleme sırasında Dow Kuramı, Japon mum formasyonları ve teknik analiz bölümünde tanımlanan formasyonlara ait eğilim dip-tepe kavramları dikkate alınmıştır. Şekil 5.8 ve Şekil 5.8'de görüldüğü üzere bir yatırım elemanın fiyatının 1 numaralı ve 2 numaralı bölgede olması etkileyici bir biçimde alım bölgesinde olduğunu göstermektedir. 1 ve 2 numaralı bölgenin bir eğilimin dip seviyesini oluşturduğu açıkça görülmektedir. Aynı zamanda 3 ve 4 numaralı bölgelerinde devam eden eğilim için tepe noktalarını oluşturduğu görülmektedir.

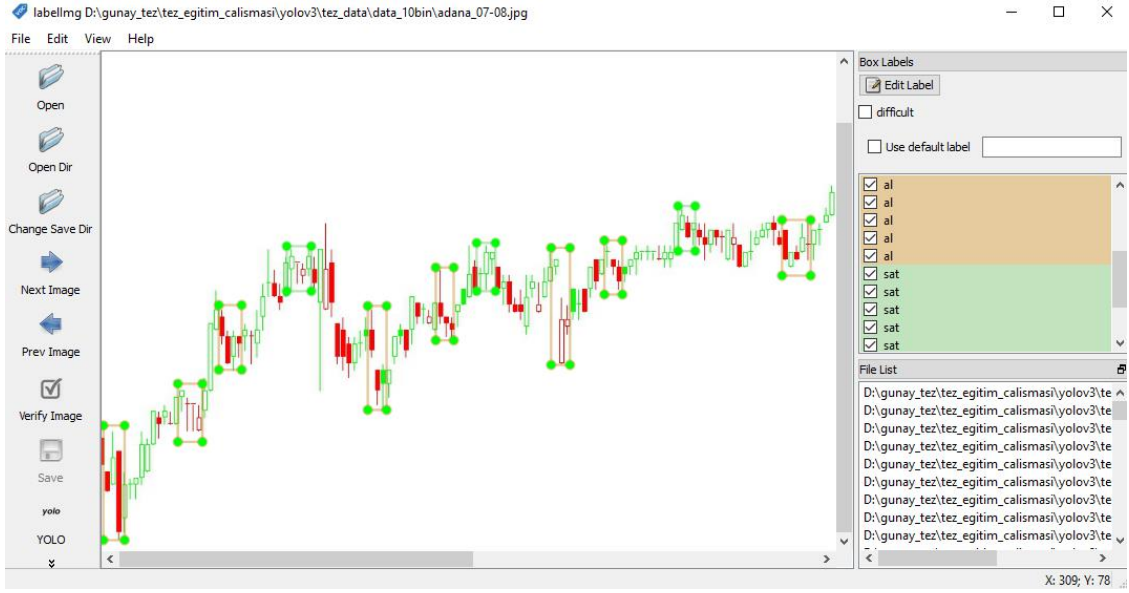


Şekil 5.7. Mum grafiği etiketleme stratejisi-1.



Şekil 5.8. Mum grafiği etiketleme stratejisi-2.

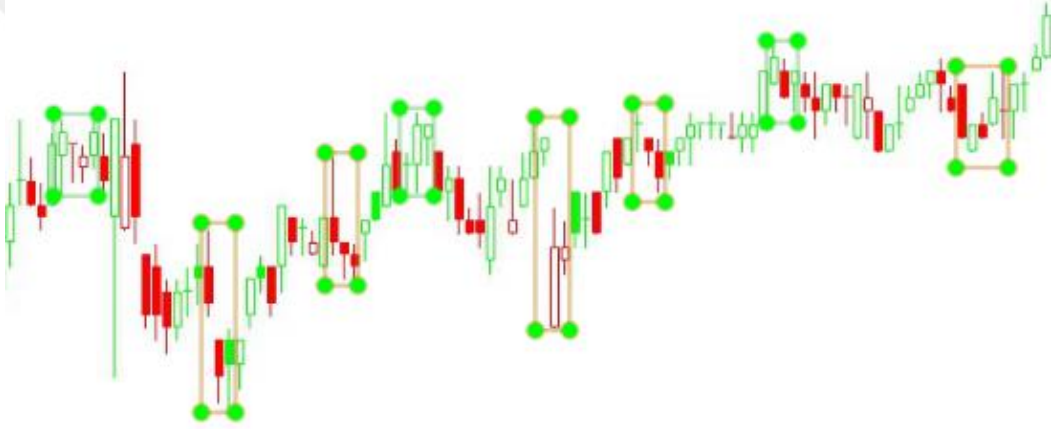
Her yatırımcı grafiğe göre 1 numaralı bölgeden alım yapıp, 3 veya 4 numaralı bölgeden satış yapmayı arzulamaktadır. Dow'a göre piyasada o an geçerli olan "fiyat", yatırımcılara fiyat düşüşüne veya yükselişine neden olan durumlardan, özellikle haberlerden daha fazla ipucu sunmaktadır (*fiyatlar önden gider, haberler onu takip eder*). Bu bakış açısı grafik bazlı analiz ve işlemleri daha da ön plana çıkartmaktadır. Yani piyasadaki yatırımcıların ihtiyacı olan bütün bilgi zaten fiyatın içindedir. Bu yüzden haberler yerine, fiyat düzeylerine yani grafiklere odaklanmak daha mantıklıdır. Bu sebeple veri setindeki eğitim grafiklerinin 1 numaralı olarak değerlendirilen dip seviyedeki mum çubuk-çubukları görüntülerinin, gelecek eğilimin yükselen habercisi olabileceğinden "Al" sinyali olarak, 3 ve 4 numaralı bölgelerde kırmızı kutular ile işaretli alanlar ise satım bölgesi ve eğilim tepe seviyeleri olduğundan düşen eğilimin habercisi niteliğinde "Sat" sinyali olarak etiketlenmişlerdir. **Veri ön işlem aşamasında ilk aşamada 550 adet yıllık grafik üzerinden 5212 "Al" ve 4870 adet "Sat" sinyali etiketlenirken, daha sonra eklenen 709 adet yıllık grafik ile birlikte 1259 görüntü üzerinde, 10128 adet "Al" ve 9935 adet "Sat" sinyali ile toplamda 20063 adet etiket oluşturulmuştur.** Bu etiketleme işlemlerinin program üzerinde gerçekleştirilmesi Şekil 5.9'da, etiketleme işlemlerine ait birkaç farklı örnek Şekil 5.10, Şekil 5.11 ve Şekil 5.12'de sunulmuştur.



Şekil 5.9. Program üzerinde etiketleme gösterimi.



Şekil 5.10. Etiketleme işlemi örnek-1.



Şekil 5.11. Etiketleme işlemi örnek-2.



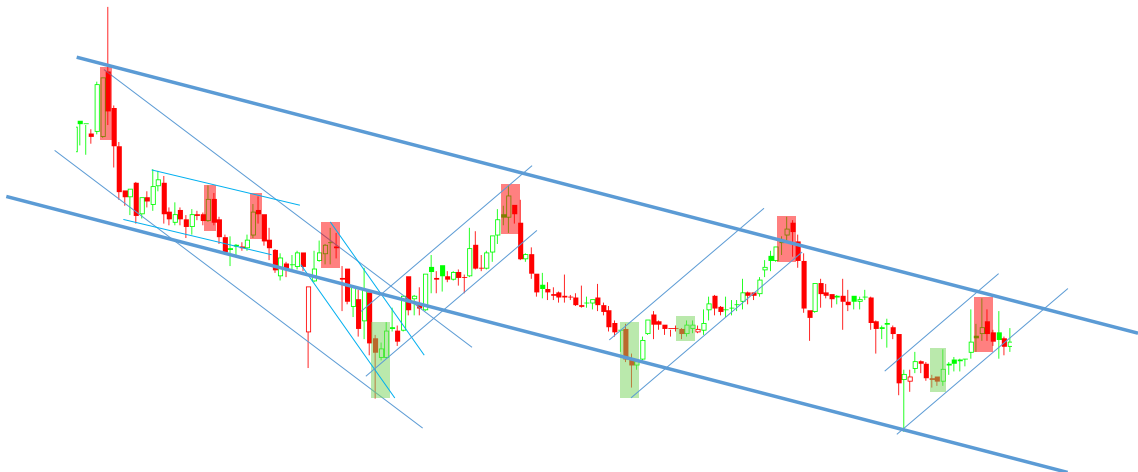
Şekil 5.12. Etiketleme işlemi örnek-3.

5.2.2.1. Etiketleme Aşamaları

Etiketleme işlemleri tamamen gözlemlene yöntemle gerçekleştirilmiştir. Grafikler üzerine çizilen yapay eğilimler ile etiketleme işlemleri kolaylaştırılmaya ve anlamlı hale getirilmeye çalışılmıştır. Bu bölümün diğer bir özelliği; mümkün olduğunca her dip ve tepenin etiket olarak belirlenmesi ile hedeflenen modelin piyasa eğilimlerini takip etmede ve saklı fırsatları yakalamada önemli araç olması düşüncesidir. Bölüm 2.4'te detaylı bir şekilde anlatılmış olan grafikler üzerinde formasyon ve indikatör analiz çalışmaları tüm dünyaca bilinen aktif bir şekilde kullanılan yöntemlerdir. Bu yöntemlerin temel hedefi bu tepe ve dip seviyelerini yatırımcılara bildirmektir. Bu sebeple olabildiğince çok sayıda etiketleme çalışması yaparak, bilinen formasyon ve indikatörlerin etiketlenmiş olmasının yanında belki de varlığı bile bilinmeyen birçok formasyon ve indikatörün etiketlenmesini sağlamaktır. Bu işlemler için bir mum grafik görüntüsü üzerinde gerçekleştirilen yapay eğilim çalışmasına örnek Şekil 5.13 ve Şekil 5.14'te verilmiştir.



Şekil 5.13. Etiketleme öncesi yapay eğilim çalışması-1.



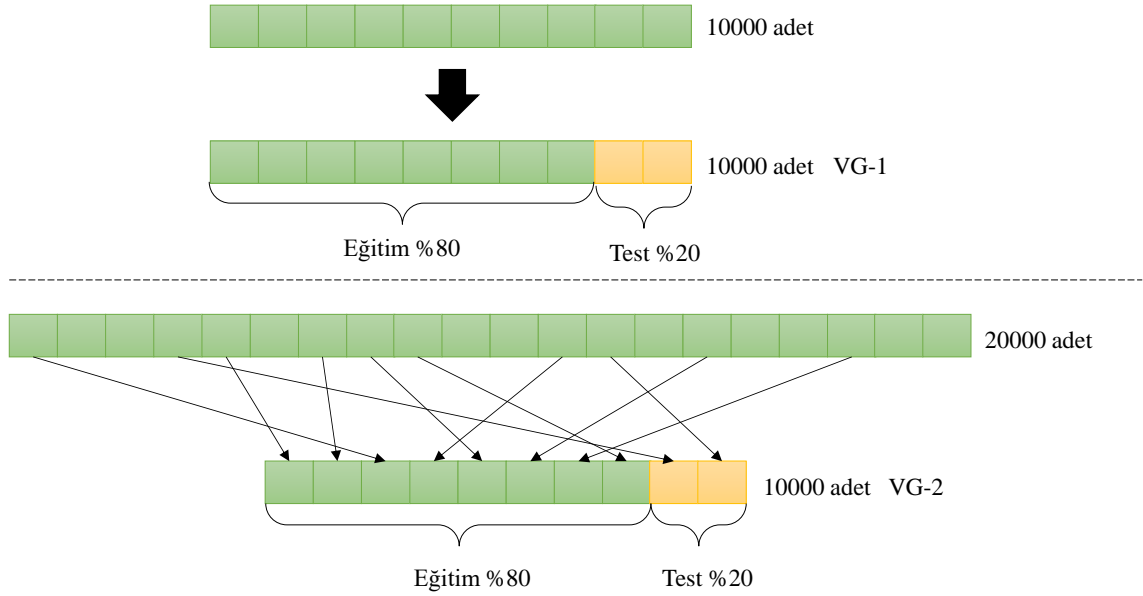
Şekil 5.14. Etiketleme öncesi yapay eğilim çalışması-2.

Bu örneklerde de görüldüğü üzere mümkün olduğunca eğilim dip ve zirveleri belirlenerek etiketler oluşturulmaya çalışılmıştır. Bilindiği üzere tüm yatırım enstrümanları zamansal veri grafiklerine sahiptirler ve hiçbir grafik sürekli yukarıya veya sürekli olarak aşağıya hareketler gerçekleştirmezler (Spekülatif davranışlar hariç). Bu sebeple finans dünyasında bilinen bütün tahminsel çalışmalar için hazırlanmış teknik analiz yöntem ve algoritmalar bu eğilim dip ve zirvelerini belirlemeye çalışırlar. Şekilde de görüleceği üzere sadece bir resim üzerinde bile dip ve zirvede oluşmuş kümesel mum görüntüleri birbirlerinden oldukça farklı davranışlar göstermektedirler. Bu farklılıklar çalışmanın kuvvetliliği açısından da önem arz etmektedir. Bu sayede oldukça fazla farklı etiket grupları elde edilmiştir. Bu noktada çalışma kapsamında bilinen tüm yöntemlerin dışında matematiksel olmayan sadece görüntü temelli bir teknik analiz çalışmasının ortaya çıktığı düşünülmektedir. Etiketleme aşamasında özellikle indikatör ve formasyon kavramlarının oluşturduğu görüntülerden kaçınılmıştır. Bunun sebebinin hali hazırda bu yöntemlerin zaten var olması ve çalışmanın bu sınırlamanın dışında kalınmasının istenmesidir.

5.2.3. Veri Gruplandırma

Şekil 5.1'de verilen akış diyagramının ilk aşamasını oluşturma kısmında tez çalışması için öncelikle bir veri seti oluşturulmuş ardından bu veri seti etiketleme işleminden geçmiştir. Son olarak model için belirlenmiş algoritmalarda eğitilmek üzere veriler gruplandırılmıştır. Daha önceden de bahsedildiği üzere bu işlem iki temel yapıda gerçekleştirilmiştir. İlk olarak VG-1'i oluşturan 550 görüntüden elde edilen yaklaşık 10000 adet etiketli veri %20 test %80 eğitim verisi olarak gruplandırılmıştır. Ardından bu veri eğitilmiş, gerekli testler yapılmış ve bir yayın gerçekleştirilmiştir. Bu test sonuçlarına ait özet gösterim Çizelge 6.8'te geniş test sonuçları ise Ek'ler bölümünde sunulmuştur [1]. Daha sonra çalışmanın genişletilmesi adına 709 adet veri seti eklenerek toplam 1259 adet veri seti oluşturulmuş ve bu veri seti üzerinde 20000 adet veri etiketlemesi gerçekleştirilmiştir. Gerçekleştirilen bu 20000 adet etiketli veri üzerinden önceki çalışma referans alınarak rastgele 10000 etiketli veri seçilmiştir. Bu seçim bir rastgele komut içeren kodlama ile gerçekleştirilmiştir. Böylelikle VG-1 veri seti büyüklüğüne bağlı kalınarak yine aynı sayıda fakat daha geniş bir veri seti üzerinden 10000 etiketli veri ile çalışma imkânı sağlanmıştır. Yeni oluşturulmuş 10000

adet etiketli veri seti de yine VG-1'de olduđu gibi %20 test %80 eğitim verisi olarak gruplandırılarak oluşturulmuş ve VG-2 olarak isimlendirilmiştir. Bu aşamaya ait görsel anlatım Şekil 5.15'te verilmiştir.



Şekil 5.15. Eğitim ve test veri seti gruplandırma.

Bu test ve eğitim veri seti gruplarının oluşturulması da yine *paylas.py* adlı bir kodlama ile rastgele olarak gerçekleştirilmiş olup kodlamalar EK'ler bölümde sunulmuştur.

6. BULGULAR VE TARTIŞMA

6.1. DONANIM ALT YAPISI VE YAZILIM KURULUMU

Tüm deneyler, Windows işletim sistemi ile çalışan, özel olarak kurulmuş bir masaüstü bilgisayarda gerçekleştirilmiştir. Bilgisayarın teknik özellikleri Çizelge 6.1’de, kullanılan yazılım kütüphanelerinin versiyonları Çizelge 6.2’de verilmiştir. Makine öğrenmesi ile ilgili yazılım kütüphanelerinin çalışmalarına yönelik kısa açıklamaları da bölüm 4.5’te verilmiştir.

Çizelge 6.1. Tez uygulamasında kullanılan bilgisayar konfigürasyonu.

Donanım	Özellik
CPU	AMD Ryzen 7 2700X Eight-Core-Processor 3.70 GHz
RAM	32 GB DDR3
GPU	NVIDIA GeForce RTX 2070 8 GB
M.2 SSD	Force MP300 256 GB
OS	Windows 10 x64

Çizelge 6.2. Tez uygulamasında kullanılan yazılım ve yazılım kütüphaneleri.

Yazılım	Versiyon
Python	3.7
Anaconda	3.0
CUDA	10.0
cuDNN	10.0
Tensorflow	1.15
OpenCV	3.30

6.2. MODEL İÇİN ALGORİTMALARIN BELİRLENMESİ VE EĞİTİMİ

Bu bölümde hedeflenen problem alanına ait çözümler için model belirleme ve modeller ile ilgili eğitim aşamaları anlatılmıştır. Gerekli eğitim işlemleri için Şekil 5.1’deki akış diyagramı ikinci aşamada belirtildiği üzere literatürde bilinen dört farklı nesne tanıma algoritması kullanılmıştır. Bu algoritmalar popüleritesi yüksek olan ve geliştirildikleri

yöntem bakımından farklı algoritmalarıdır.

Nesne tanıma algoritmaları Region Based ve Unified Pipeline şeklinde isimlendirilen iki farklı yapıda çalışmaktadırlar. Region based algoritmalar her ne kadar daha iyi sonuçlar verdiği söylene de mobil ve giyilebilir cihazlar için hesaplama işlemleri oldukça maliyetli ve yavaşlardır. Bu grubun liderliğini Faster RCNN oluştururken, YOLO algoritmasının liderliğini oluşturduğu Unified Pipeline grubu algoritmalar son zamanlarda geliştirilen karmaşık bölge tespit stratejileri ile hem başarılarını arttırmış hem de oldukça hızlı sonuçlar elde edebilmektedirler. Nesne tanıma alanında bilinen en temel algoritmalar aşağıda listelendiği gibidirler [253].

- R-CNN Networks
- SPP
- Fast R-CNN
- Faster R-CNN
- RetinaNet (Focal loss)
- Yolo Framework — YoloV1, YoloV2, YoloV3, YoloV4
- SSD
- Feature Pyramid

Bu çalışmada da Region Based tabanlı algoritmaların en hızlısı olan Faster RCNN ve Unified Pipeline tabanlı algoritmalarından YOLO v3, v4 versiyonu ile SSD algoritmaları ile tasarımlar gerçekleştirilmiştir. Seçilen bu dört algoritma için eğitimler gerçekleştirilmiş ve her bir algoritmanın başarımları sonuçları değerlendirilmiştir.

6.2.1. Eğitimlerin Dosya Yapılanması

Eğitim işlemleri iki farklı dosya yapılanmasında gerçekleştirilmiştir. YOLO versiyonları ek açıklamalar için YOLO formatı kullanırken, Faster RCNN ve SSD algoritmaları Pascal VOC ek açıklama formatlarını kullanmaktadırlar. Bu sebeple test ve eğitim setleri iki farklı ek açıklama içerikli olarak iki grupta dosyalanmışlardır. Bu dosya yapılanma işlemleri YOLO versiyonlar için Şekil 6.17 ve Faster RCNN ve SSD için Şekil 6.18’de görüldüğü gibidir. Dosya yapılanmasının ardından veri seti eğitimi için her iki grup algoritmada da ayarlanması gereken bazı parametreler mevcuttur. Bu parametrelerin başında etiket isimleri yer alır. Bunlar YOLO grubu için “.names” adlı dosya ve Faster RCNN, SSD için “labelmap.pbtxt” isimli dosyalardır. Bu dosyalar içeriğinde etiketleme sırasında oluşturduğumuz etiket isimlerini barındırırlar. Bu etiketler algoritmaların anlayabileceği şekilde rakamsal değerlerin karşılığı olarak ifade edilmişlerdir. Dosyaların içeriğine ait gösterim Çizelge 6.3’te verilmiştir.

Çizelge 6.3. Etiket dosyaları içeriği.

YOLO (tez.names)	Faster RCNN – SSD (label.pbtxt)
al sat	<pre> item { id: 1 name: 'al' } item { id: 2 name: 'sat' } </pre>

Bu dosyaların haricinde yine bu çalışma kapsamında oluşturulan veri seti için eğitim aşamasında ayarlanması gereken bazı parametreler mevcut olup, bu parametreler iki grup için de konfigürasyon dosyaları içerisinde yer almaktadır. Konfigürasyon dosyaları içerisinde gerçekleştirilen parametre değişikliklerine ait ayarlamalar Çizelge 6.4'te yer verilmiştir. Çizelgede bulunan veri değerleri bu çalışmadaki modeller için belirlenmiş değerler olup, “-“ ile belirtilmiş alanlar dosya için girilecek parametre olmadığını göstermektedir.

Çizelge 6.4. Konfigürasyon dosyaları içeriği.

Parametre	YOLO (tez.cfg)	Faster RCNN – SSD (faster_rcnn_inception_v2_10bin.config) (ssd_mobilenet_v2_coco.config)
Sınıf Adedi	2	2
Filtre Sayısı	21	-
Öğrenme Oranı	0.001	0.001
Test Örnek Sayısı	-	110
Eğitim Süresi	~90 saat	~16 saat

Bu parametrik değerlerin ayarlanması dışında eğitim-test verilerinin ve konfigürasyon dosyalarının transfer öğrenim için kullanılacak olan önceden eğitilmiş ağırlık değerlerinin sürücü adres/yol bilgilerinin saklı tutulduğu dosyalar yapılandırılmıştır.

Tensorflow, iyi bir nesne tespit aracı eğitmek için bir nesnenin yüzlerce görüntüsüne ihtiyaç duyar. Önceki aşamalarda elde ettiğimiz etiketlemelerin ardından kullanılacak tüm algoritmaların tensorflow ile eğitilmesi aşamasına geçilmiştir. Daha öncede belirtildiği üzere bu eğitim adımları iki farklı grup şeklinde gerçekleştirilmiştir. Birinci grup Faster R-CNN ve SSD grubu olup ikinci grup Yolo versiyonlarıdır. Faster R-CNN grubu için bu aşamada ilk olarak etiketleme sonucu elde edilen görüntülerin ek açıklama

dosyalarına ait eğitim ve test grubu olarak ayrılmış .xml verileri, bir .csv dosya türüne dönüştürülerek listelenmiştir. Bu işlem adımı Şekil 6.1’de verilen komut istemi ile gerçekleştirilmiştir. Bu komut istemini çalıştırılması sonucu *images* klasörü içerisinde *images/train_label.csv* ve *images/test_label.csv* dosyaları oluşturulmuş olup elde edilen .csv dosyalarına ait içerik gösterim Şekil 6.2’de verilmiştir.

```
Administrator: Anaconda Prompt
(tensorflow) D:\gunay_tez\tez_egitim_calismasi\faster_cnn\models_fcnn\research\object_detection>python xml_to_csv.py
```

Şekil 6.1. Eğitim ve test ek açıklamaları .xml to .csv komut istemi.

	A	B	C	D	E
1	filename,width,height,class,xmin,ymin,xmax,ymax				
2	acsel_13-14.jpg,1767,688,al,171,383,207,458				
3	acsel_13-14.jpg,1767,688,al,516,343,553,413				
4	acsel_13-14.jpg,1767,688,al,1200,508,1236,594				
5	acsel_13-14.jpg,1767,688,al,264,377,292,447				
6	acsel_13-14.jpg,1767,688,al,1371,480,1406,533				
7	acsel_13-14.jpg,1767,688,al,1503,449,1532,577				
8	acsel_13-14.jpg,1767,688,al,1729,431,1758,681				
9	acsel_13-14.jpg,1767,688,al,1617,515,1645,581				
10	acsel_13-14.jpg,1767,688,sat,221,13,236,228				
11	acsel_13-14.jpg,1767,688,sat,686,115,701,317				
12	acsel_13-14.jpg,1767,688,sat,855,285,890,378				
13	acsel_13-14.jpg,1767,688,sat,990,316,1019,399				
14	acsel_13-14.jpg,1767,688,sat,102,231,138,327				
15	acsel_13-14.jpg,1767,688,sat,1110,362,1146,413				

Şekil 6.2. Ek açıklama .csv dosya içeriği.

Yolo versiyonları için bu aşamada, eğitim adımlarında direk olarak ek açıklama dosyası olan .txt dosya içeriğinden okuma yapıldığından harici bir işlem gerçekleştirilmemiştir.

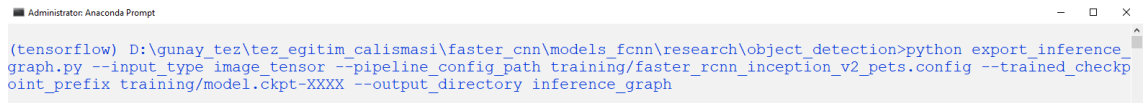
Son olarak yapılandırma, test ve eğitim dosyalarının da oluşturulmasının ardından nesne tespiti eğitimine başlamak için komut satırına Faster R-CNN örneği için Şekil 6.3, Yolo için Şekil 6.4’de verilen komut istemleri yazılmıştır. Bu komut istemlerinin çalıştırılmasının ardından her şey doğru ayarlanmış ise tensorflow eğitimleri başlar. Eğitim için sistemin donanımsal ön incelemelerinin ardından yaklaşık bir 30 saniyelik süreç için de eğitimler başlar. Bu eğitimlerin her biri farklı zamanlarda gerçekleştirilmiştir.

```
Administrator: Anaconda Prompt
(tensorflow) D:\gunay_tez\tez_egitim_calismasi\faster_cnn\models_fcnn\research\object_detection>python train.py --logtostderr --train_dir=training_10bin/ --pipeline_config_path=training_10bin/faster_rcnn_inception_v2_10bin.config
```

Şekil 6.3. Faster R-CNN ve SDD algoritmaları eğitim başlatma komut istemi.

veya kullanıcının belir bir adım sonrasında sistemin eğitildiğine karar vermesi sonucu klavyeden gönderdiği bir kesme ile sonlandırılmaktadır. Bu aşamada eğitimin kesilmeyip belirtilen adım sayısına eğitilmesinde yarar vardır. Böylece kullanıcının elinde daha fazla ağırlık değerleri mevcut olacaktır.

Eğitim bittiğinde en son ağırlık dosyaları kaydedilmiş olarak görünebilmektedir. Bu noktada Yolo algoritmaları için herhangi bir ek yapılandırma yapılmasına ihtiyaç duyulmamaktadır. Ağırlık değerleri direkt olarak test işlemlerinde kullanılabilir. Fakat Faster R-CNN ve SSD algoritmaları için en son kaydedilen eğitim ağırlık değerlerinin bir dışa aktarım işleminden geçmeleri gerekmektedir. Bu dışa aktarım için kullanılan komut istemi Şekil 6.7’te verilmiştir.

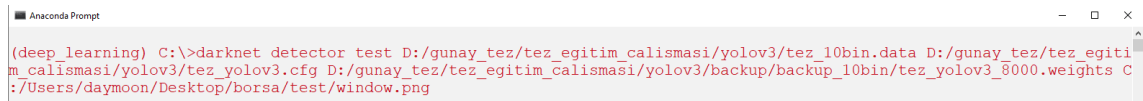


```
Administrator: Anaconda Prompt
(tensorflow) D:\gunay_tez\tez_egitim_calismasi\faster_cnn\models_fcnn\research\object_detection>python export_inference_graph.py --input_type image_tensor --pipeline_config_path training/faster_rcnn_inception_v2_pets.config --trained_checkpoint_prefix training/model.ckpt-XXXX --output_directory inference_graph
```

Şekil 6.7. Faster R-CNN ve SSD eğitim ağırlık değerleri dışa aktarım komut istemi.

Bu komut isteminin çalıştırılması ile *object_detection/inference_graph* klasöründe bir *frozen_inference_graph.pb* dosyası oluşturur. Bu “.pb” dosyası, nesne tespit ve sınıflandırıcısı için ağırlık değerleri içerir.

Bu eğitim süreçleri sonucunda Şekil 5.1’de verilen akış diyagramının ikinci aşamasını oluşturan eğitim işlemleri sonucunda modellerin ağırlık değerleri elde edilmiştir. Modeller elde edilen yeni ağırlık değerleri ile bu çalışma kapsamındaki özel nesnelere tespit etmek için hazır duruma getirilmiştir. Eğitim sonrasında hazırlanan araçların test edilmesi amacıyla her iki grup algoritma ayrı ayrı kendi ilgili komut istemleri ile çalıştırılmışlardır. Faster R-CNN ve SSD algoritmaları *object_detection_image.py* kod dosyası bir python “idle” yardımı ile çalıştırılmış olup, Yolo algoritmaları “*darknet.exe*” ile komut satırı üzerinden çalıştırılmışlardır. Yolo için komut satırı istemi Şekil 6.8’de gösterildiği gibidir.



```
Anaconda Prompt
(deep_learning) C:\>darknet detector test D:/gunay_tez/tez_egitim_calismasi/yolov3/tez_10bin.data D:/gunay_tez/tez_egitim_calismasi/yolov3/tez_yolov3.cfg D:/gunay_tez/tez_egitim_calismasi/yolov3/backup/backup_10bin/tez_yolov3_8000.weights C:/Users/daymoon/Desktop/borsa/test/window.png
```

Şekil 6.8. YOLO test işlemi komut istemi.

Bu araçlarının test edilebilmesi için test edilecek resim görüntülerinin saklanabileceği bir test klasörü oluşturulmuştur. Oluşturulan klasör içerisine algoritmaların daha önce test ve eğitim veri seti içerisinde olmayan yani araçların hiç görmediği resimler

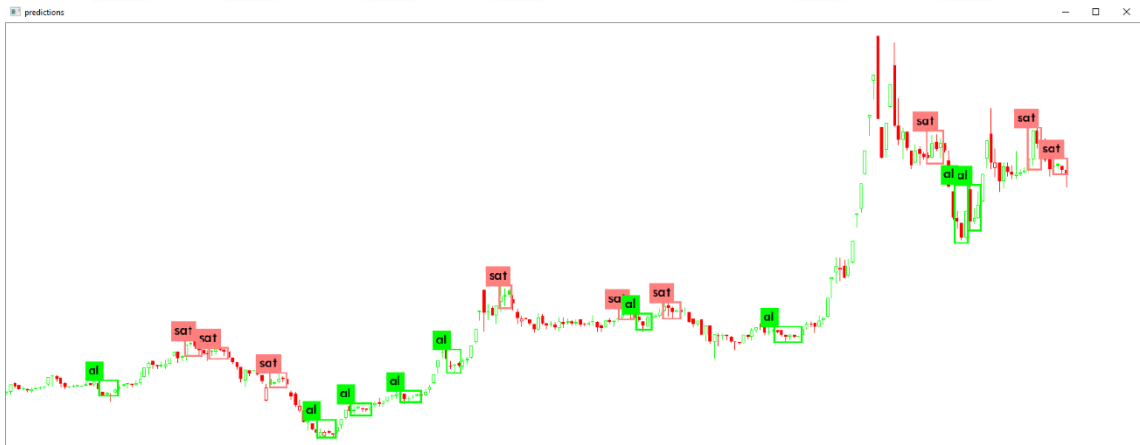
eklenmiştir. Yolo ile test işlemi, kodlama sayesinde toplu olarak yapılabildiği gibi, tek tek de gerçekleştirilebilmektedir. Tekli test işleminde test edilmek istenilen resim, bir python kodlama yardımı ile otomatik olarak istenilen boyutta ve istenilen klasöre kaydedilmiş ve ardından ilgili araç ile test edilmiştir. Bu resim kayıt işlemi gerçekleştiren program kod bloğu Şekil 6.9'da verilmiştir.

```
*screen.py - D:\gunay_tez\screen_uygulama\screen.py (3.7.1)*
File Edit Format Run Options Window Help
import pyautogui
import pkg_resources.py2_warn
import time
import os

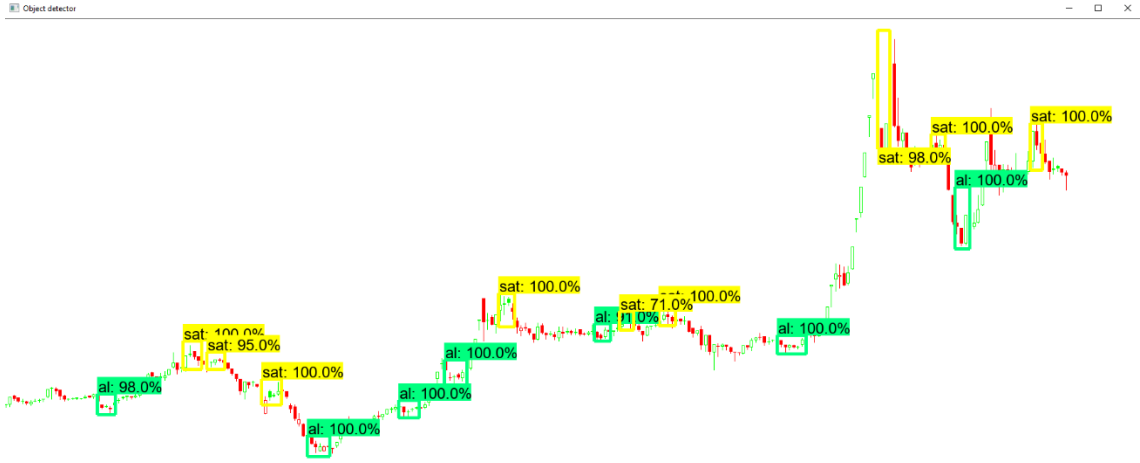
print("Lütfen bekleyiniz....Resim Kaydediliyor")
resim = pyautogui.screenshot('C:/Users/daymoon/Desktop/borsa/test/window.png', region=(93,73, 1766, 687))
```

Şekil 6.9. Ekran görüntüsü alma kod bloğu.

Şekilde gösterilen bu python kod bloğu, *pyautogui* kütüphanesi sayesinde bölge koordinatları ile boyutu belirlenen bir finansal grafik görüntüsünü istenilen adrese kaydedebilmektedir. Ardından sıra ile araçlar çalıştırılarak test işlemleri gerçekleştirilmektedir. Eğitim sonrasında gerçekleştirilen testlere ilişkin nesne tespitine ait örnek görüntüler aşağıda sırası ile Şekil 6.10, Şekil 6.11, Şekil 6.12, Şekil 6.13, Şekil 6.14, Şekil 6.15, Şekil 6.16'da verilmiştir.



Şekil 6.10. YoloV3 VG-2 test çıktısı örnek görünüm-1.



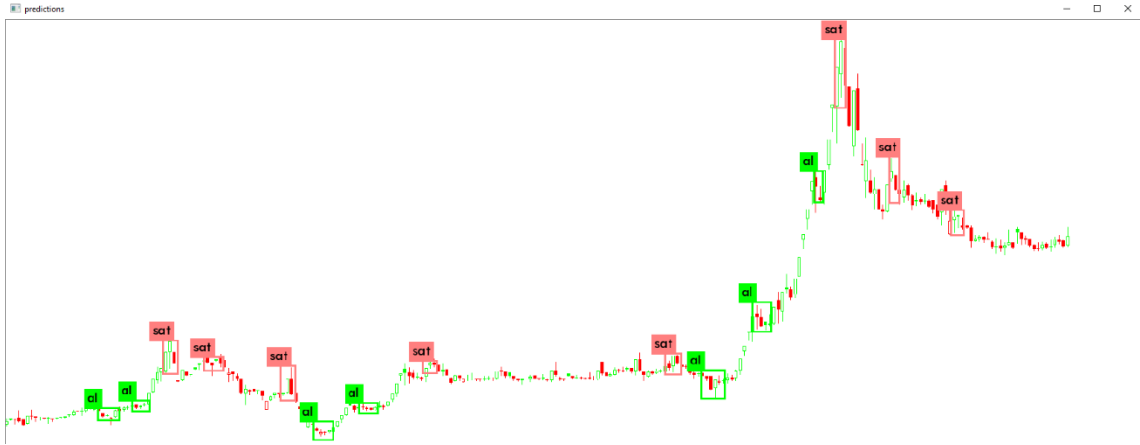
Şekil 6.11. Faster R-CNN VG-2 test çıktısı örnek görünüm-2.



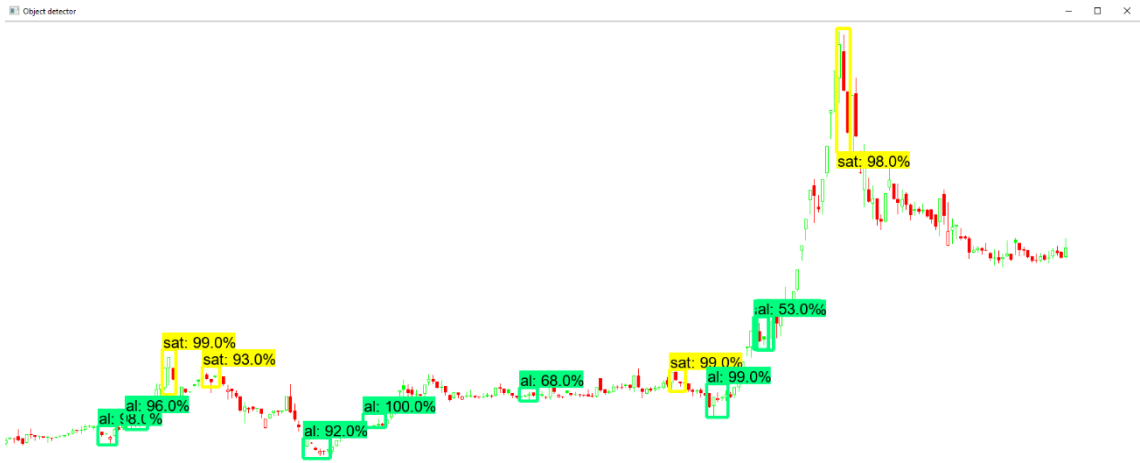
Şekil 6.12. YoloV3 VG-2 test çıktısı örnek görünüm-3.



Şekil 6.13. Faster R-CNN VG-2 test çıktısı örnek görünüm-4.



Şekil 6.14. YoloV3 VG-2 test çıktısı örnek görünüm-5.



Şekil 6.15. Faster R-CNN test çıktısı örnek görünüm-6.



Şekil 6.16. YoloV3 VG-2 test çıktısı örnek görünüm-7.

darknet detector train D:/gunay_tez/tez_egitim_calismasi/yolov3/tez_10bin.data D:/gunay_tez/tez_egitim_calismasi/yolov3/tez_yolov3.cfg D:/gunay_tez/tez_egitim_calismasi/yolov3/darknet53.conv.74



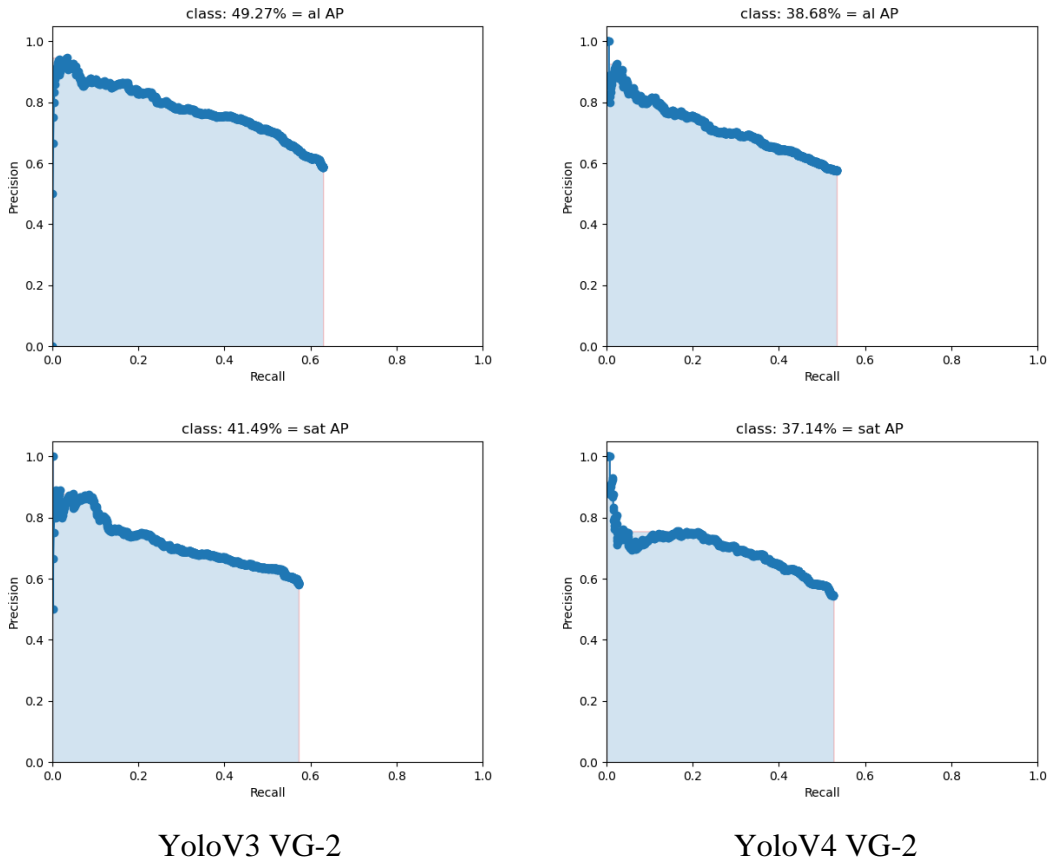
Şekil 6.17. YOLO modelleri dosya yapılanması.



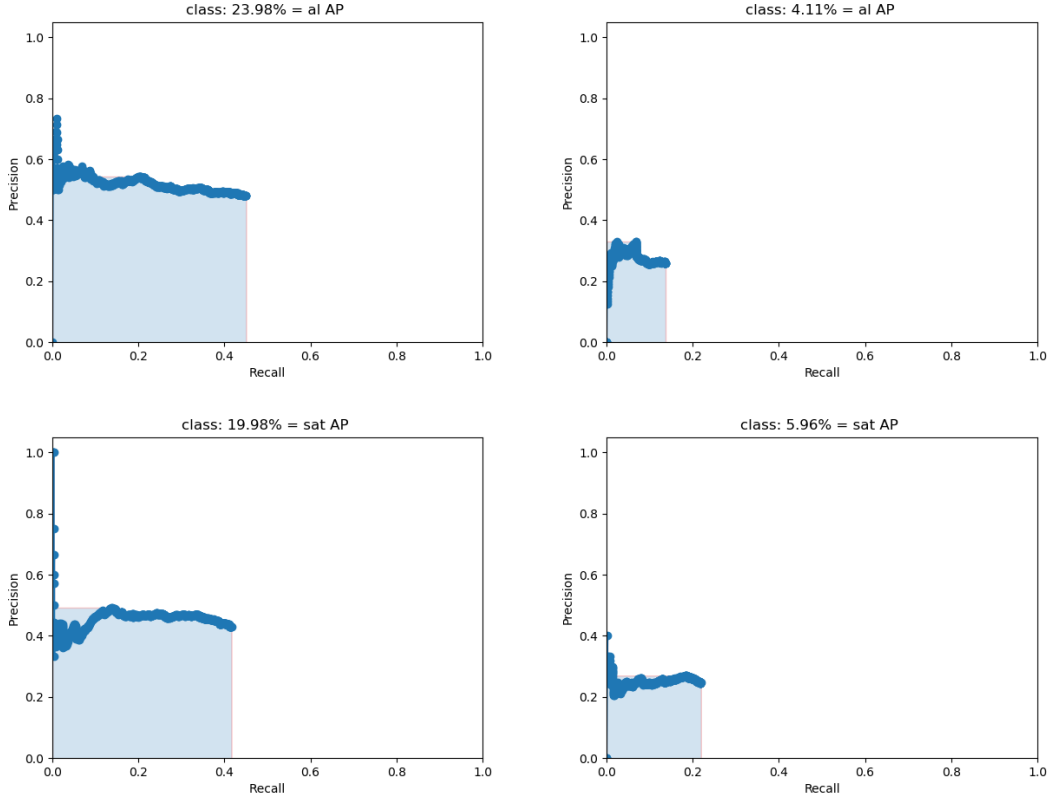
Şekil 6.18. Faster R-CNN ve SSD modelleri dosya yaplanması.

6.2.2. Eğitimlerin Metrik Değerlendirme Sonuçları

Modellerin eğitim aşamalarının ardından elde edilen ağırlıklarının nesne tanıma performansını değerlendirmek üzere bir dizi testlerden geçirilmiştir. İlk olarak eğitim çıktılarını AP, mAP, Precision ve Recall değerleri bakımından incelenmiştir. Bu incelemeler python programlama dili yazılmış bir kod bloğu ile gerçekleştirilmiştir [254]. Bu işlemler için öncelikle eğitim sonucu elde edilen yeni ağırlık değerlerine sahip algoritmalar ile test veri seti sisteme sokulmuş ve nesne algılaması gerçekleştirilen çıktı görüntüleri tek tek kaydedilmiştir. Ardından çıktı görüntülerinde elde edilen tespitler üzerine bir konumlandırma çalışması yine *LabelIMG* programı ile gerçekleştirilmiştir. Böylelikle görüntü üzerinde hem nesne etiketi hem tahmin etiketi oluşturulmuştur. Son olarak kullanılan yazılım ile metrik değerlerin hesaplanması gerçekleştirilmiştir. Bu metrik değerlerin hesaplanması için uygulanan aşamaların ardından elde edilen *Precision ve Recall* değerlerine dolayısıyla AP değerlerine ait çıktı grafikleri Şekil 6.19 ve Şekil 6.20’de verilmiştir.



Şekil 6.19. Modellerin AP değerlendirmesi-1.

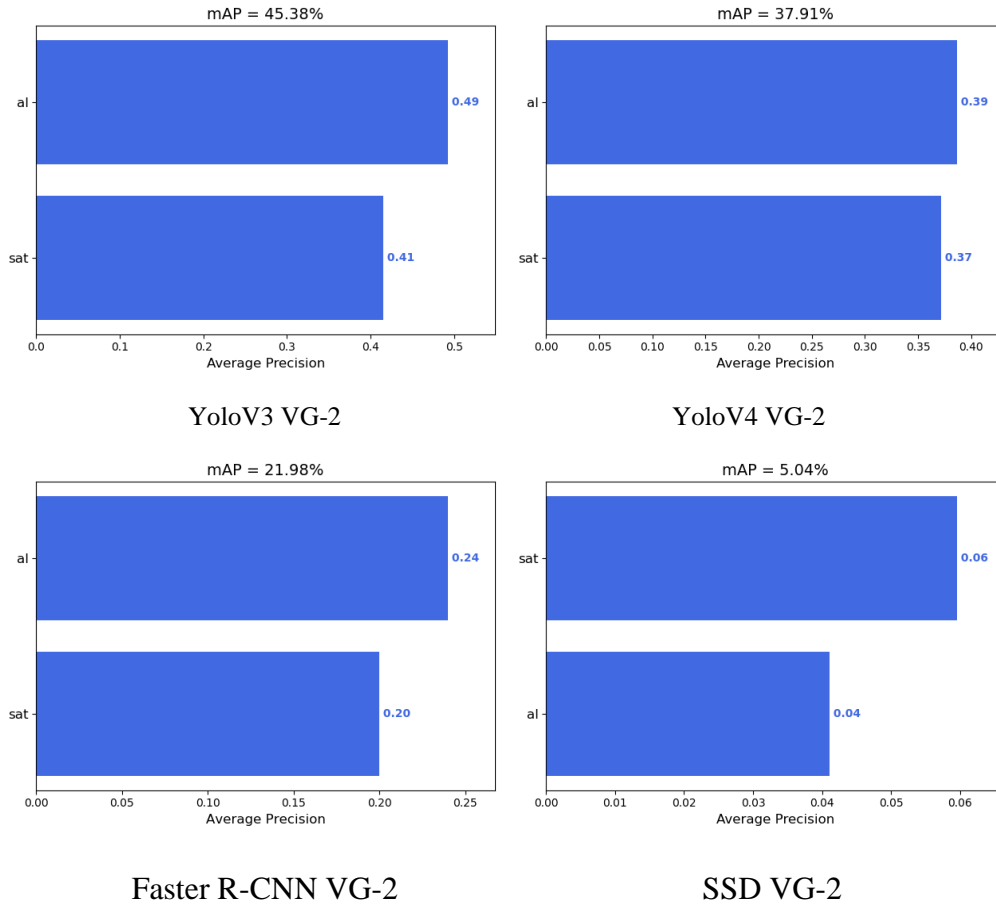


Faster R-CNN VG-2

SD VG-2

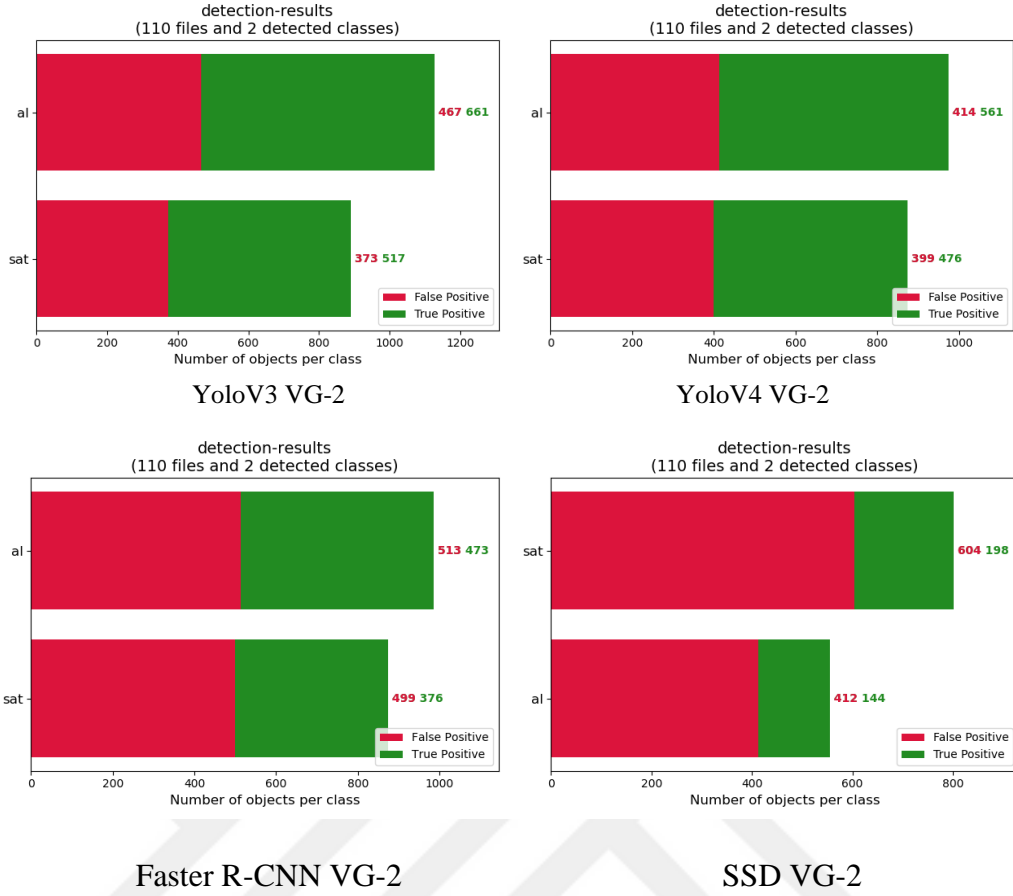
Şekil 6.20. Modellerin AP değerlendirmesi-2.

Temel olarak bu metrik değerlendirme precision-recall için IoU 0.5 metriğine bağlıdır. Denklem 4.2 ve 4.3'te verilen TP, FP, FN değerlerine göre işlemler gerçekleştirilmiş olup grafiksel gösterimler hazırlanmıştır. Bu hesaplamalar sonucunda elde edilen mAP değerlendirmesine ait grafik gösterim Şekil 6.21'de verilmiştir.



Şekil 6.21. Modellerin mAP değerlendirmesi.

Grafiksel gösterimde de görüleceği üzere başarılı olan model mAP değeri en yüksek olan model olarak ifade edilebilir. Burada YoloV3 VG-2 model %45.38 mAP değeri ile en başarılı model olarak ifade edilir. COCO veri seti için de değerlendirilecek olduğunda Yolov3 COCO %55.3'lük bir başarıyı 80 sınıf üzerinde gerçekleştirmiştir [255]. Modellerin diğer bir karşılaştırma metriği olan etiketlemeler ve modellerin test veri seti için etiketlenen konumların tahmin edilme değerleridir. Bu karşılaştırmaya ait grafik gösterim Şekil 6.22'te verilmiştir. Bu bağlamda yine etiketlenen konumları tahmin etme yüzdesi en başarılı model olarak Yolov3 VG-2 şeklinde ifade oldukça yerinde olacaktır.



Şekil 6.22. Modellerin tespit sonuçları (FP-TP).

Diğer taraftan tüm bu metrik değerlendirme işlemleri genel literatürün %99'un da kullanılan metrik karşılaştırma işlemleri olarak ifade edilebilir. Metrik değerlendirme sonuçları incelendiğinde genel literatür çalışmalarına kıyasla çok kötü bir sonuç elde edilmediği ortalama bir başarıya ulaşıldığı gözlemlenmektedir. Fakat çalışmanın giriş kısmında da belirtildiği üzere bu çalışma genel nesne tanıma algoritmaları ile çalışılmış olmakla birlikte bugüne kadar çalışılmamış farklı bir sınıf odaklı çalışmayı temsil etmektedir. Bu yüzden çalışma ile ilgili genelin dışında da bir değerlendirme yapılması gerekmektedir. Bu bağlamda sistem tarafında test edilen test veri seti *LabelIMG* programı üzerinde tek tek incelemeye alınmıştır. İncelenen test veri seti içeriğinden örnekler Şekil 6.23'te sunulmuştur.



Şekil 6.23.YoloV3 VG-2 tespit sonuçları değerlendirilmesi.

Genel literatürde üzerine basılarak ifade edilen TN değerleri hesaplanmaya alınmamaktadır. Şekilde de örneği görüleceği üzere modeller TP tahminlerin yanı sıra birçok TN tahminler üretmiştir. TN tahminler backgroundda da bahsedildiği üzere etiketlenmemiş bir durumun gerçekten doğru tahmin edilmesini ifade etmektedir. Yani kaçırılan pozitifler olarak da ifade edilirler. Yukarıda belirtilen mAP hesaplama yazılımları veya model eğitim yazılımları genel olarak bu görüntü üzerindeki değerlemeleri yeniden tahmin edemediğinden direkt olarak FP olarak değerlendirmektedirler. Bu durum 80-200 sınıftan oluşan büyük veri setlerinde (COCO – PASCAL VOC vb.) sonuçları büyük oranda etkilememektedir. Fakat bu çalışmadaki sınıf sayısının 2 olması durumu oldukça değiştirmektedir. Bu bağlamda çalışmaya konu olan sınıfların sıra dışı etiket sınıfları olması sebebi ile görüntüler üzerindeki test tahmin çıktıları tek tek analiz edilmiş ve sonuçlar Excel formülleri yardımıyla elde edilmiştir. Elde edilen sonuçlara ait gösterim Çizelge 6.5'te sunulmuştur.

Çizelge 6.5. Test veri seti model değerlendirmesi.

MODEL	Test Veri Seti Boyutu	TP	TN	FP	FN	Tahmin Sayısı	Precision	Recall	AP
YoloV3 VG-2	1948	1323	589	0	625	1912	1.00	0.69	1,00
YoloV4 VG-2		1100	305	0	848	1405	1.00	0.57	1,00
Faster R-CNN VG-2		1325	488	234	623	1813	0.84	0.61	0,96
SSD VG-2		792	1199	224	1156	1324	0.78	0.51	0,89

Çizelgede de görüleceği üzere Yolo versiyonlar için sonuçlar oldukça iyidir. Fakat bu çalışma kapsamında Faster R-CNN ve SSD modeller için tahmin doğruluk oranlarının yüksek olması genel modellerden daha iyi olması bile başarılı anlamına gelmemektedir. Çalışma bir yatırım kararları destekleyici nitelikte olduğunda %1'lik bir başarısızlık bile başarısız yatırımlara sebep olacağından modeller için başarılıdır denilemez. Daha kapsamlı test ve analizlere bölüm 5'te yer verilmiştir. Bu tez kapsamında finansal mum serileri 2D grafik görüntüleri üzerinde "Al-Sat" karar destek modeli tasarlanmıştır. Tasarlanan model, literatürde sayısız çalışma gerçekleştirilen, finansal veriler üzerinde tahminler yapılan modellerden farklı olarak sayısal işlemlerden ve algoritmalar ile hesaplanan sayısal sonuçlardan tamamen uzaktır. Tasarlanan model, bir yatırım enstrümanına ait 2D mum grafikleri üzerindeki alınacak ve satılacak eğilim başlangıç noktalarını birer nesne olarak tespit edilmesini sağlamaktadır. Genel olarak model, bir nesne tanıma ve sınıflandırma algoritması tabanlıdır. Nesne tespit ve sınıflandırma algoritmalarından yola çıkarak tasarlanan model için öncelikle ilgili etiketler oluşturulmuş ve bu etiketler ile literatürde popülaritesi yüksek dört farklı nesne tanıma algoritması eğitilmiştir. Eğitim sonucu elde edilen ağırlık değerlerinin entegre edildiği modeller bir dizi başarımlar testlerinden geçirilmiştir. Şekil 5.1'de verilen akış diyagramı üçüncü aşamasında bu yeni ağırlık değerlerine sahip modellerin, eğitim başarımları karşılaştırması gerçekleştirilmiştir. Yapılan karşılaştırmalarda elde edilen test-tespit sonuçlarına ait bulgular Çizelge 6.6'da verilmiştir.

Çizelge 6.6. Test veri seti tahmin değerleri.

MODEL	Test Veri Seti Etiket Sayısı	Tahmin Yüzdesi	Tahmin Doğruluk Oranı
YoloV3 VG-2	1948	98%	100%
YoloV4 VG-2		72%	100%
Faster R-CNN VG-2		93%	96%
SSD VG-2		68%	89%

Bu test veri seti değerlendirme işlemi test veri setinin yeni ağırlık değerlerine sahip modeller ile test edilerek tahmin doğruluğu üzerine gerçekleştirilen bir karşılaştırma işlemidir. Bu tespit sonuçlarında en çok ve başarılı tespiti YoloV3 VG-2 nesne tespit algoritması gerçekleştirmiştir. Çizelge 6.6’da verilen değerlendirme sonuçlarındaki tahmin yüzdesi, etiketlenen etiket sayısına karşılık, modelin ürettiği tahmin sayısını belirtmektedir. Test veri seti olarak toplam 110 adet grafik görünümünden oluşmaktadır. Bu grafik görünümünde 1948 etiket mevcuttur. Buna karşılık YoloV3 VG-2 algoritması grafikler üzerinde 1912 adet nesne tespiti gerçekleştirmiştir. Bu adımda %98 bir tespit başarısı elde edilmiştir. Diğer bir karşılaştırma olarak tespit edilen kararların doğruluk oranlarıdır. YoloV3 VG-2 bu noktada da ürettiği sonuçlar bakımından hatalı bir karar üretmemiş olup doğruluk oranı %100 olarak tespit edilmiştir. Her ne kadar tahmin ettiği karar sayısı düşük olsa da YOLOV4 VG-2 versiyonu da tahmin başarısında %100 bir başarı sağlamıştır. Diğer taraftan Faster R-CNN VG-2 algoritması 1813 nesne tespit sayısı ile %93’lük bir başarı gösterirken SSD VG-2 algoritması %63’lük bir tespit değeri sağlamıştır. Ancak bu noktada algoritmaların başarılarında aradığımız kriter, tespit sayısında gösterilen başarı olmayıp, tespit edilen nesnenin doğruluk oranıdır. Bunun da sebebi modelin üreteceği bir kararın hatalı olması, başarısız bir yatırıma yol açabileceğinden kaynaklanmaktadır. Bu sebeple Faster R-CNN VG-2 ve SSD VG-2 algoritmalarındaki doğruluk oranlarının %100’den düşük olması hatalı bir karar tespiti verdiğini göstermektedir. Algoritmaların test aşamasında ortak bir mum grafik üzerinde ürettikleri çıktılara ait görünüm Şekil 6.24, Şekil 6.25, Şekil 6.26 ve Şekil 6.27’de sunulmuş olup bu test değerlendirme işlemlerine ait genel grafik görünüm Şekil 6.28’de verilmiştir.



Şekil 6.24. YoloV3 VG-2 test çıktısı



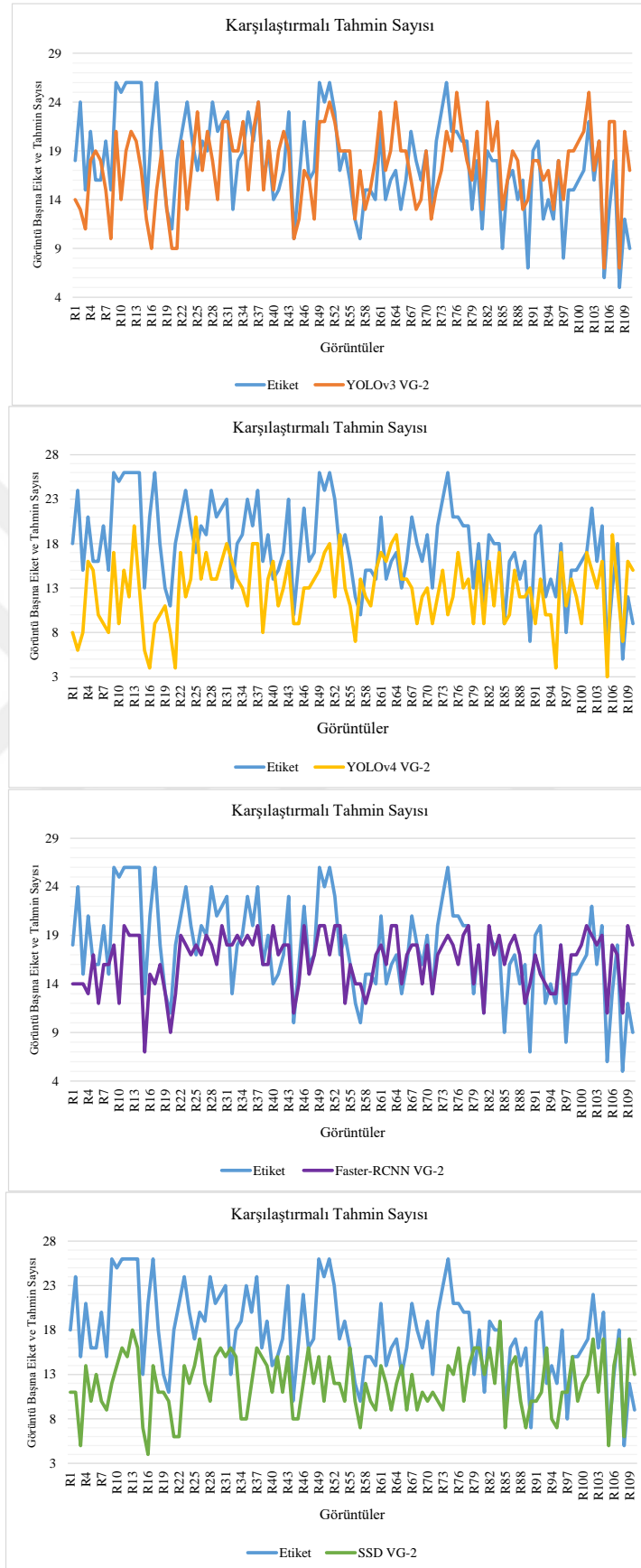
Şekil 6.25. YoloV4 VG-2 test çıktısı.



Şekil 6.26. Faster R-CNN VG-2 test çıktısı.

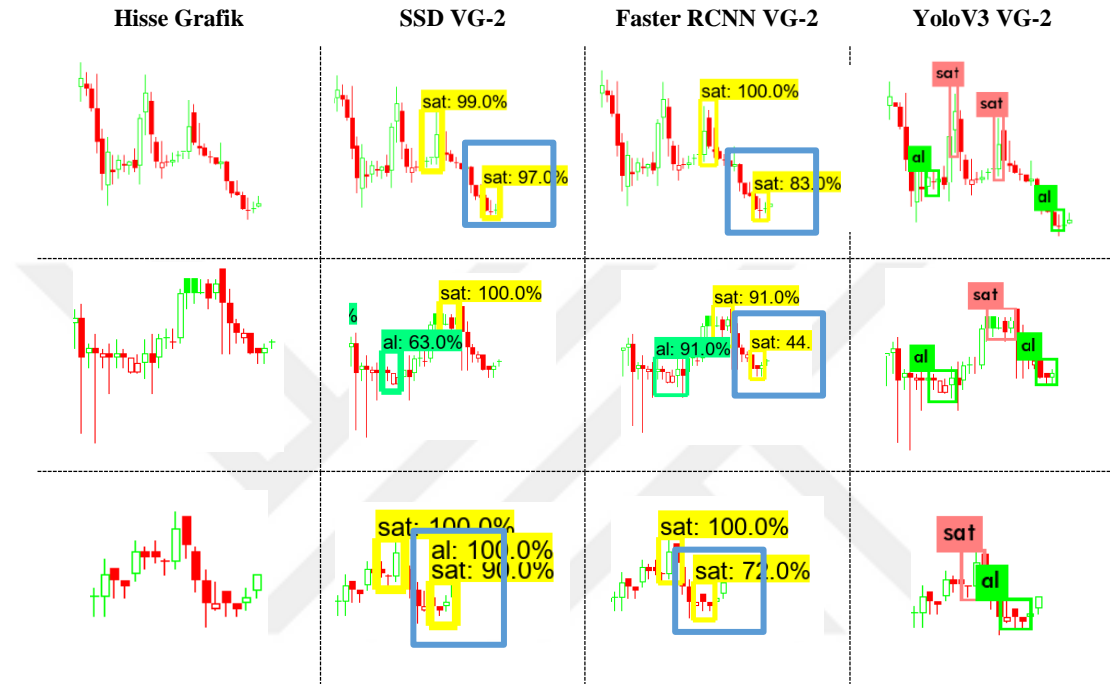


Şekil 6.27. SSD VG-2 test çıktısı.



Şekil 6.28. Algoritmaların test veri seti üzerindeki tahmin değerleri.

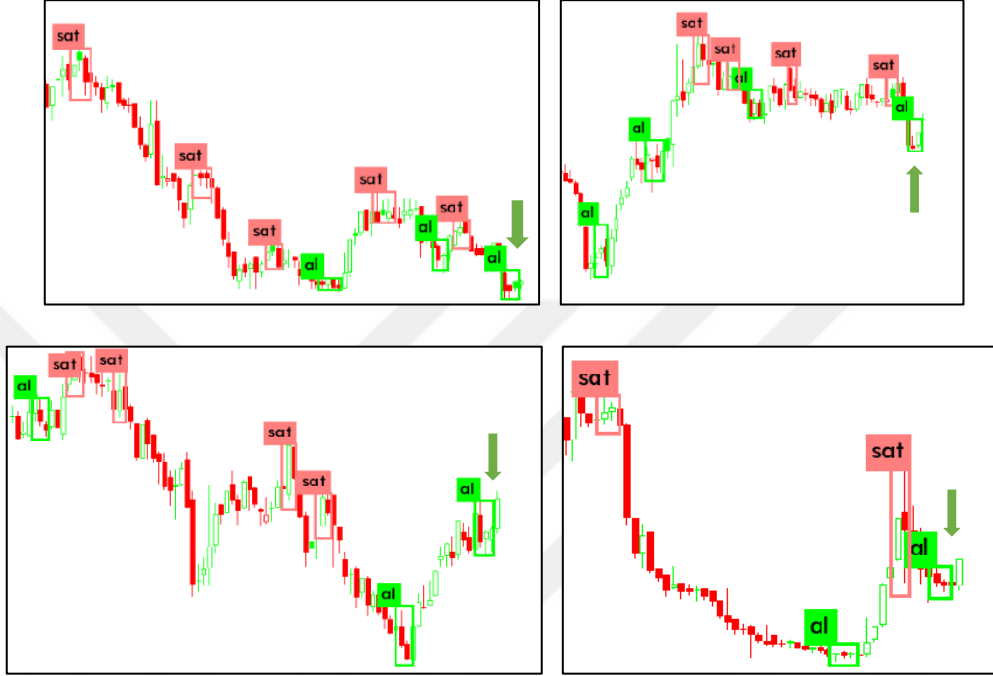
Belirtilen iki algoritma modelinin hatalı karar veya kararsız tespit ürettikleri bazı çıktılara ait grafik görünüm Şekil 6.29’da verilmiştir. Şekilde de görüldüğü üzere Faster R-CNN ve SSD algoritmalarına ait örnek görüntülerde YOLO algoritmalarından farklı kararlar üretmişlerdir. Bu durum algoritmaların doğruluk oranını düşürmüştür.



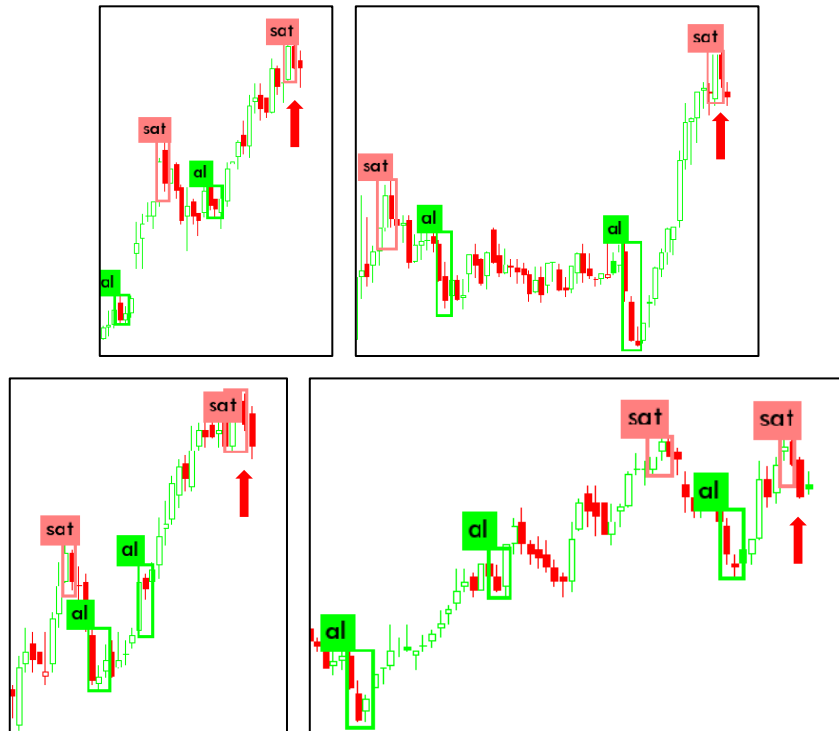
Şekil 6.29. Faster R-CNN ve SSD algoritmalarının hatalı karar görüntüleri

Şekil 5.2’de verilen akış diyagramının dördüncü basamağındaki değerlendirme işlemleri için BIST-100’e dahil 5 farklı hisse senedi seçilmiştir. Bu hisse seçim işlemi bir rastgele fonksiyon olan; $\gg\gg random.randrange(1, 100)$ ile gerçekleştirilmiştir. Seçilen hisselerin altı aylık dönemdeki günlük mum görüntüsüne ait grafik görüntüleri kaydedilmiştir. Bu aşamada test işlemleri, eğitilen dört farklı model (YoloV3 VG-2, YoloV4 VG-2, SSD VG-2, Faster R-CNN VG-2) ve YoloV3 VG-1 ile birlikte 2019 Ocak-Temmuz ayları arasında seçilen hisse grafik görüntüleri için gerçekleştirilmiştir. Test işlemi için gün sonuna ait grafik görüntüler üzerinde tüm modeller ile tespit karar çıktısı oluşturulmuş, oluşan çıktı sonuçlarına göre "Al-Sat" işlemleri gerçekleştirilmiştir. Modellerin ürettiği "Al-Sat" karar etiketleri Şekil 6.30 ve Şekil 6.31 örneklerindeki gibi görüntü üzerinde dağılım göstermektedirler. Görüntü üzerindeki eğilimler içinde birçok noktada bu karar etiketlerini görmek mümkündür. Fakat gerçek zamanlı yatırım işlemleri için görüntü üzerinde herhangi bir noktada

üretilen kararların değil, aktif olarak bir sonraki güne ait karar verilmesini sağlayan son gün karar tespitleri önemlidir. Bu sebeple yatırım kararlarının verilebilmesi için son gün/günler, mum/mumların üzerinde oluşan "Al-Sat" kararları dikkate alınmaktadır. Bu bağlamda "Al-Sat" kararı verilen grafik görüntülerine ait örnekler Şekil 6.30 ve Şekil 6.31'te verilmiştir. Kararları etkileyen tespit noktaları ok işaretleri ile belirtilmiştir.



Şekil 6.30. Yatırım için "Al" kararını belirleyen örnek grafik görüntüleri.



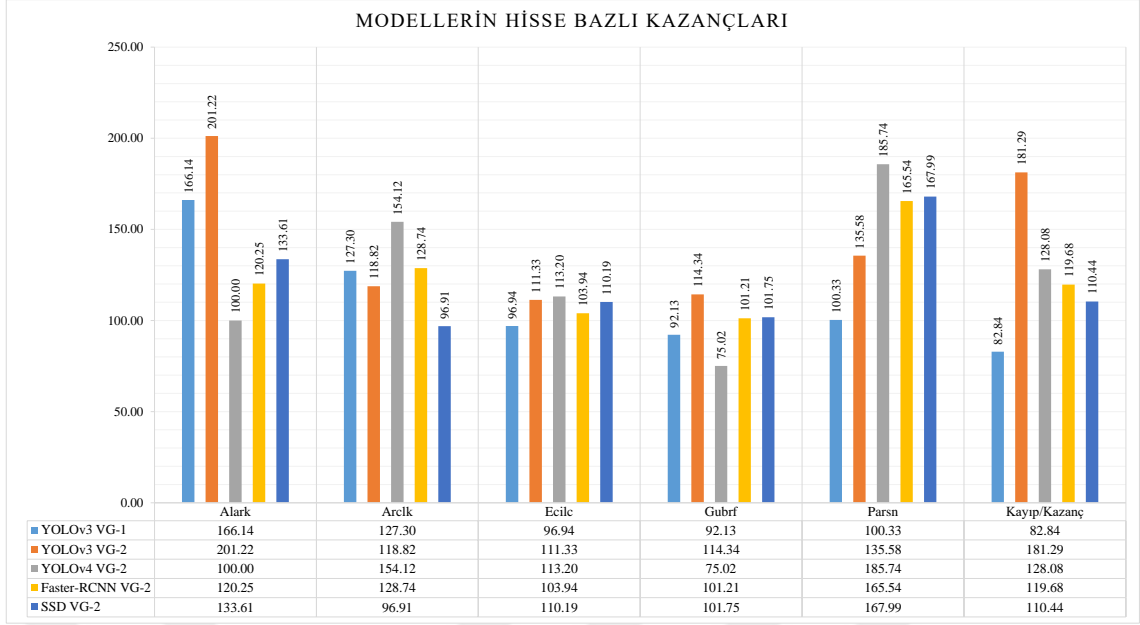
Şekil 6.31. Yatırım için “Sat” kararını belirleyen örnek grafik görüntüleri.

Başlangıç olarak her hisse senedi için bir nakit miktarı belirlenmiş ve bu nakit miktarı üzerinden kar-zarar analizi gerçekleştirilmiştir. Altı aylık test işlemleri sonunda elde edilen kar-zarar verilerine ait detay bilgiler bir tablo şeklinde kaydedilmiştir. Tabloya ait gösterim Çizelge 6.7’te verilmiştir.

Çizelge 6.7. Modellerin BIST-100 seçili hisse değerlendirilmesi.

Hisse	Anapara TL	MODEL				
		10000 veri seti 10000 etiket (VG-1)	20000 veri seti 10000 etiket (VG-2)			
		YoloV3 VG-1	YoloV3 VG-2	YoloV4 VG-2	Faster-RCNN VG-2	SSD VG-2
Alark	100	166.14	201.22	100.00	120.25	133.61
Arcık	100	127.30	118.82	154.12	128.74	96.91
Ecilc	100	96.94	111.33	113.20	103.94	110.19
Gubrf	100	92.13	114.34	75.02	101.21	101.75
Parsn	100	100.33	135.58	185.74	165.54	167.99
Toplam	500	582.84	681.29	628.08	619.68	610.44
Kayıp/Kazanç TL		82.84	181.29	128.08	119.68	110.44

Çizelgede de görüleceği üzere her bir hisse için 100TL değerinde bir anapara ayrılmış olup, modellerin 6 ay sonunda getiri sağladıkları kayıp-kazanç miktarları belirtilmiştir. Bu test aşamasında da en yüksek getiri ile başarı sağlayan modelin YoloV3 VG-2 olduğunu belirtmek gerekir. Yine çizelgede ilk versiyon diye adlandırılan YoloV3 VG-1 model en düşük getiriye sahip olan modeldir. Buradan da anlaşılacağı üzere büyük bir etiket havuzu oluşturup farklı etiketler ile eğitim gerçekleştirmek ilk çalışmaya kıyasla model başarısını arttırdığı görülmektedir. Ayrıca belirtilmesi gereken önemli bir hususta tüm modeller için bir kazanç söz konusudur. Sonuç olarak bu aşamada test edilen modellerin tümünün kazançlı sonuçlar sağlayabildiği net olarak söylenebilir. Modellerin hisse bazlı kazançlarına ait grafiksel dağılım Şekil 6.32’de verilmiştir.



Şekil 6.32. Modellerin hisse bazlı kazanç dağılım grafiği.

Bu aşamanın ardından, Şekil 5.2’de verilen akış diyagramının son aşaması olan ve genel literatürde sepet sistemi olarak bilinen yatırım yöntemi ile model test işlemleri gerçekleştirilmiştir. Bu test işlemleri makale çalışmasında gerçekleştirilen YoloV3 VG-1 ile bu çalışma kapsamında eğitilen ve bu ana kadar ki testlerde en başarılı sonuçlar alınan YoloV3 VG-2 arasındadır. Sepet kavramı, yatırım işlemlerinde çoklu senetlere sahip olarak zararı minimize eden kavram olarak bilinir. Bir başka deyişle portföyde birden fazla senede sahip olmak olarak adlandırılır. YoloV3 VG-1 için 31 Ağustos 2018 – 28 Şubat 2019 aralığındaki günlük ”Al-Sat” karar sinyalleri ile alım ve satım işlemleri gerçekleştirilmiştir. Bu alım satım işlemleri sonucunda modelin başarısı, belirtilen anaparayı hangi seviyeye getirdiği ile değerlendirilmiştir. Bu çalışmada da önceki değerlendirmeler baz alınarak en başarılı model olan YoloV3 VG-2 ile aynı tarihler üzerinde test işlemleri gerçekleştirilmiştir. Sepet mantığında gerçekleştirilen test işlemlerine ait ilk değerlendirme ve bu çalışma kapsamındaki değerlendirme sonuçlarına ait karşılaştırmalı özet tablo Çizelge 6.8’de verilmiştir.

Çizelge 6.8. Sepet işlemleri özet tablosu.

Test	YoloV3 VG-1	YoloV3 VG-2
Başlangıç Tarihi	31 Ağustos 2018	
Bitiş Tarihi	28 Şubat 2019	
Toplam Gün/Ay	181 / 6.03	
Sepet Sayısı	8	
Senet Sayısı (Al-Sat)	73	78
Toplam Kar	%59,25	%89,84
Başlangıç Anapara	10.000TL	
Sonuç Anapara	15.925,21TL	18.984,42TL

Çizelge genel olarak bir özetlemedir. Toplam elde edilen kar-zarar durumu ile birlikte bu oluşumun süreçleri hakkında bilgi vermektedir. Çizelgede belirtilen sepet sayısı altı aylık test sürecinde oluşturulan senet gruplarının sayısını belirtmektedir. İlk olarak anaparanın 10.000TL olduğu varsayılarak işlemler gerçekleştirilmiştir. 10.000TL’lik bir anapara başlangıcı ile gerçekleştirilen bu işlemlerin ardından anaparanın YoloV3 VG-1 testleri için anapara 15.925,21TL’ye ulaşırken YoloV3 VG-2 testlerinde anaparanın 18.984,42TL’ye ulaştığı görülmektedir¹. Başarı açısından bakıldığında ilgili çalışmalar bölümünde bahsedilen finansal tahminleme konusunda çalışılan birçok yöntemle göre oldukça başarılı sonuçlar üreten bir model tasarımı gerçekleştirildiği söylenebilir. Ayrıca genel değerlendirme olarak modeller 8 sepet grubu içerisinde toplamda 6 dönemi kazançlı sonuçlandırırken 2 dönemi zararlı sonuçlandırmıştır. Bu zararlar ise oldukça düşük yüzdedir.

Bu test işlemleri sırasında kayıp-kazanç bilgilerinin ölçülmesi adına “Al” sinyali üretilen senetlerin fiyat ve tarih bilgileri ile “Sat” sinyali üretilen senetlerin fiyat tarih bilgileri bir tabloya kaydedilmiştir. İşlemler sonucunda oluşturulmuş ve örneği verilen bir sepet tablosu görünümü Şekil 6.33’de verilmiştir.

¹ Bu kazanç hesaplamasına “Al-Sat” durumlarında oluşacak komisyon oranları (%0.02) dâhil edilmemiştir.

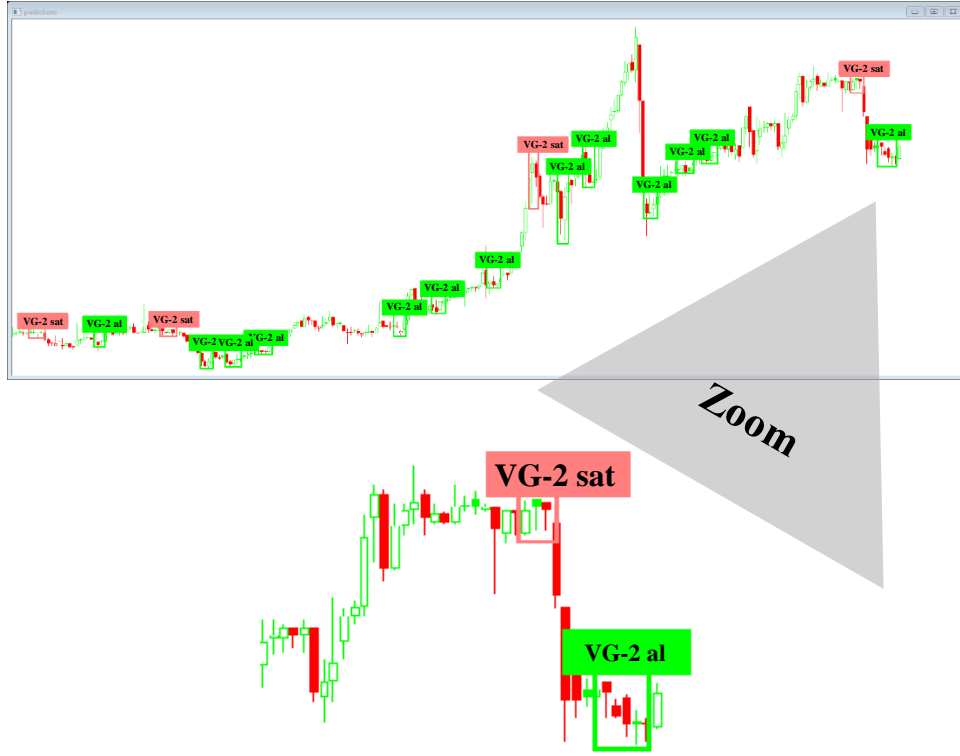
YoloV3 VG-1						YoloV3 VG-2					
Hisse	Al		Sat		Sonuç	Hisse	Al		Sat		Sonuç
	Öneri Tarihi	Öneri Fiyatı	Öneri Tarihi	Öneri Fiyatı	% Kar/Zarar		Öneri Tarihi	Öneri Fiyatı	Öneri Tarihi	Öneri Fiyatı	% Kar/Zarar
AKSGY	31.8.18	1.69	7.9.18	1.75	3.55	AKSGY	31.8.18	1.69	11.09.2018	1.71	1.18
AKSUE	31.8.18	6.50	25.9.18	8.92	37.23	AKSUE	31.8.18	6.50	26.09.2018	8.63	32.77
DOGUB	31.8.18	2.43	13.9.18	3.09	27.16	ARSAN	31.8.18	1.90	6.09.2018	1.86	-2.11
INDES	31.8.18	5.48	11.9.18	5.79	5.66	INDES	31.8.18	5.48	12.09.2018	5.61	2.37
KERVT	31.8.18	1.92	10.9.18	1.93	0.52	KERVT	31.8.18	1.92	11.09.2018	1.86	-3.12
KLMNA	31.8.18	3.26	6.9.18	3.21	-1.53	LİNK	31.8.18	8.81	7.09.2018	9.25	4.99
SEYKM	31.8.18	3.90	28.9.18	3.99	2.31	SEYKM	31.8.18	3.90	28.09.2018	3.99	2.31
						TURGG	31.8.18	30.80	4.09.2018	30.26	-1.75
Sepet Ortalama Kar/Zarar					10.70	Sepet Ortalama Kar/Zarar					4.58

Şekil 6.33. Örnek Sepet-1 görünümü.

Şekilde görünen bir sepet grubu için “Al” sinyallerine ait örnek gösterimler Şekil 6.34, Şekil 6.35 ve Şekil 6.36’da, “Sat” sinyallerine ait örnek gösterimler Şekil 6.37, Şekil 6.38 ve Şekil 6.39’da verilmiştir. Şekillerde nesne tespitlerine ait görüntüler, modellerin ürettiği bütünlük halinde verilmiş olup görünebilmesi ve anlaşılabilmesi adına ilgili bölüm *zoom* şeklinde ifade edilerek yakınlaştırma görüntüsü oluşturulmuştur. Yakınlaştırılmış görüntülere dikkat edilecek olunursa, genel “Al-Sat” stratejimizi oluşturan son gün/günlere ait mum/mumlar üzerinde oluşan tespitler geçerlidir. Tahmin için bu aşamada modellerin kendilerine ait model ismi olan VG-1 ve VG-2 isimleri ile ilgili etiketler kullanılmış ve bu test aşamasındaki tüm “Al-Sat” işlem aşamaları bu stratejiye göre gerçekleştirilmiştir.



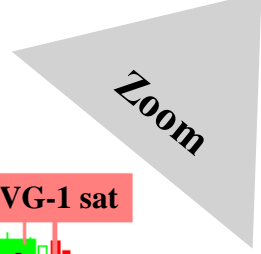
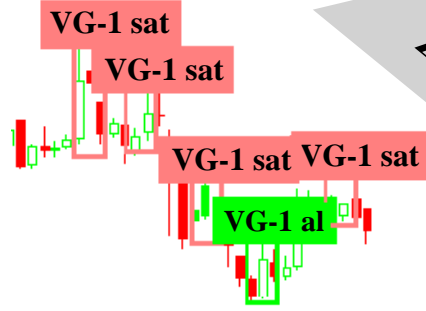
Şekil 6.34. Al sinyali üreten senet görünüm örnek-1.



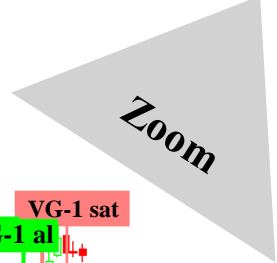
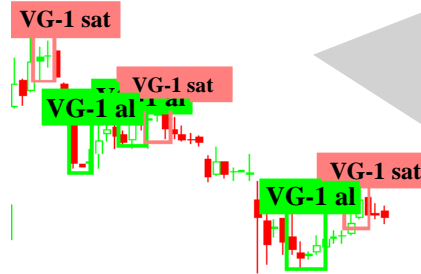
Şekil 6.35. Al sinyali üreten senet görünüm örnek-2.



Şekil 6.36. Al sinyali üreten senet görünüm örnek-3.



Şekil 6.37. Sat sinyali üreten senet görünüm örnek-1.



Şekil 6.38. Sat sinyali üreten senet görünüm örnek-2.



Şekil 6.39. Sat sinyali üreten senet görünüm örneği-3.

Şekil 6.33'teki tablonun oluşumu, toplamda YoloV3 VG-1 için 7, YoloV3 VG-2 için 8 farklı senet ile gerçekleştirilmiştir. Bu sayılar sadece 1 adet sepet grubunu temsil etmektedir. Tüm senetler için hatta tüm sepet grupları için "Al" sinyalleri belirli şekilde aynı günlerde oluşturulmuş iken "Sat" sinyallerinin oluşumu, farklı zamanlara denk gelmektedir. Bunun sebebi; strateji olarak belirli bir günde arama sonucu oluşan "Al" sinyallerinin ardından en son senet için "Sat" sinyali oluşumuna kadar bekleme gerçekleştirilmiştir. Portföyümüzde var olan senetlerden herhangi birinin test sonucunun "Sat" vermesi durumunda sonuç değerlendirilmesi yapılarak takibi sonlandırılmıştır. Bu şekilde bir grubun maksimum süreç sayısı belirlenmiştir. Her senet için oluşan "Sat" sinyali ile sinyalin olduğu gün sonu fiyatı tabloya eklenmiştir. "Al" sinyali ile "Sat" sinyalinin belirlendiği fiyat farklılığı kar-zarar sonucunu vermektedir. Yine bu kar-zarar değerinin yüzdesel hesaplaması gerçekleştirilip tabloya eklenmiştir. Tabloda görülen %kar sütünü satır bazında her senet için hisse senedinin getirisini ifade etmektedir. Sonuç olarak Sepet 1'de, senetlerin yüzdelik karların ortalaması hesaplandığında YoloV3 VG-1 için %10,71 iken YoloV3 VG-2 için bu oran %4,58 olarak belirlenmiştir. Burada Sepet 1 için YoloV3 VG-1'in daha başarılı olduğu gözlemlenmiştir. Fakat bu durum tablonun başında oluşmuş ve 28 günlük süreci kapsayan Sepet 1 için geçerlidir. Bu bağlamda Ek'ler bölümünde tablonun devamı ve Şekil 6.40'daki kazanç grafiği incelendiğinde YoloV3 VG-2'nin kazanç başarısının uzayan süreçte YoloV3 VG-1'den

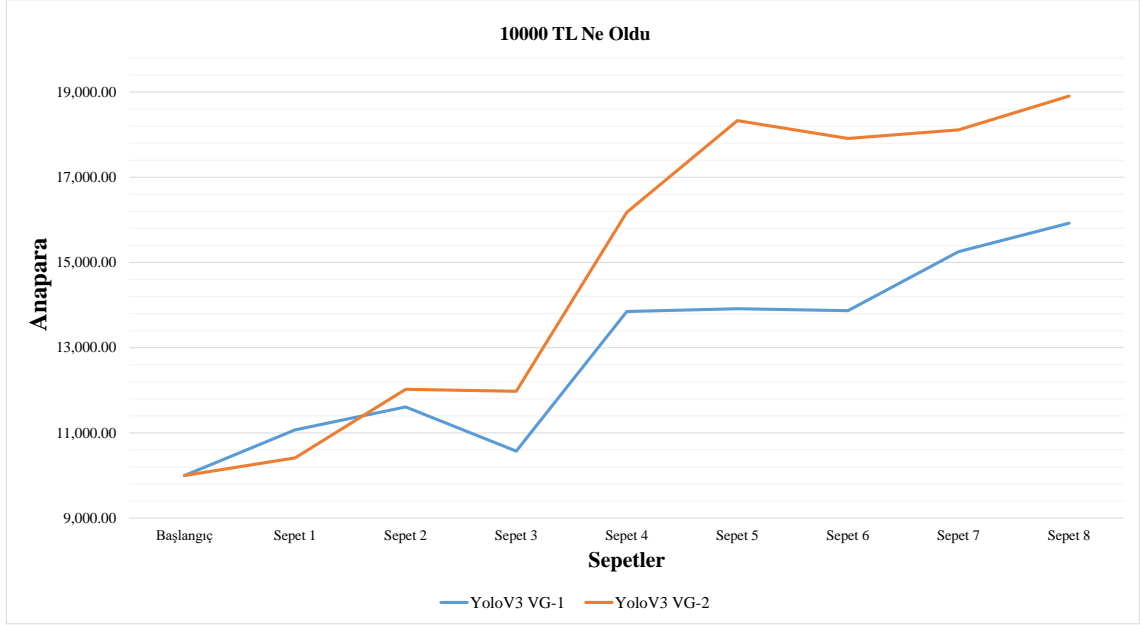
pozitif yukarı yönlü ayrıştığı net bir şekilde görülmektedir. Bu şekilde altı aylık süreçte 7 farklı sepet grup testi daha gerçekleştirilmiştir. Bu işlemlere ait veri uzun bir liste olduğundan tam tablo Ek'ler bölümde sunulmuştur. Genel tabloya ait elde edilen bilgiler yine bir özet halinde Çizelge 6.9'da verilmiştir. Bu çizelge, altı aylık süreçte modellerin gerçek kazanç durumlarını ifade etmek amacıyla kullanılan bir özet tablodur. Çizelgedeki sepet numaraları, en son satılan senet işleminden sonra oluşturulan yeni sepet ile 1,2,3,...,8 şeklinde numaralanmış toplam 8 sepet olduğunu ifade etmektedir. İşlem adedi ise her bir sepet içerisindeki "Al" sinyali ile sepete dahil edilen hisse senedi sayısını belirtir. Toplamda da görüldüğü üzere altı aylık süreç için hazırlanan bu test tablosunda YOLOv3 VG-1'de 73 adet hisse için, YoloV3 VG-2 için 78 adet hisse için "Al" işlemi ve aynı sayıda "Sat" işlemi gerçekleştirilmiştir.

Çizelge 6.9. Modellerin toplam sepet portföy değerlendirilmesi.

Sepet No	İşlem Adedi		Dönemsel % Kar		Anapara 10000TL		
	YOLOv3 VG-1	YoloV3 VG-2	YOLOv3 VG-1	YoloV3 VG-2	YOLOv3 VG-1	YoloV3 VG-2	
Sepet 1	7	8	10.70	4.58	11069.90	10458.04	
Sepet 2	7	8	4.87	15.45	11609.10	12073.41	
Sepet 3	4	5	-8.92	-0.38	10573.46	12027.69	
Sepet 4	7	8	30.98	35.06	13848.96	16244.78	
Sepet 5	9	8	0.49	13.29	13917.10	18403.94	
Sepet 6	12	9	-0.35	-2.27	13867.74	17986.37	
Sepet 7	10	12	10.00	1.13	15254.60	18189.48	
Sepet 8	17	20	4.40	4.37	15925.21	18984.42	Anapara
SONUÇ	TOPLAM	73	76	Toplam Kar %	%59,25	%89,86	

Çizelgede yer alan *Anapara* sütunu ise, her bir sepet oluşumu sonrasında anaparanın dönemsel karı/zararı ile hesaplanarak elde edilen toplam miktarı temsil etmektedir. Her bir döneme ait kar/zarar yüzdesi belirlenmiş ve bu kar/zarar yüzdesi bir önceki anaparanın getirisi olarak hesaplanmıştır. Hesaplama anapara sepet oluşumdaki senet sayısına eşit olarak paylaştırılmış ve değerlendirme gerçekleştirilmiştir. Ardından hesaplanan kazanç veya kayıp ile elde kalan anapara tekrar sonraki sepette al sinyali veren senet sayısına bölünerek her bir senet için eşit olarak dağıtılarak alım işlemi gerçekleştirilmiştir. Bu alım-satım stratejisi, bu çalışmaya özgü olmayıp genel sepet mantığı olarak ifade edilir. Tablo sonuç satırında da ifade edildiği gibi modellerin başarı oranı oldukça yüksek değerlerdedir. 180 günlük bir süreçte gerçekleşen "Al-Sat"

neticelerinde YoloV3 VG-1 %59'luk bir kazanç başarısı elde ederken YoloV3 VG-2 modelinin %89'luk bir kazanç sağlayabildiği gözlemlenmiştir. Sonuç olarak bu çalışma kapsamında yapılandırılan YoloV3 VG-2 modeli tüm test sonuçlarında başarı sağlayarak final model olarak belirlenmiştir. Gerçekleştirilen altı aylık test işlemleri sonucunda elde edilen kazançlara ait model grafik gösterimleri Şekil 6.40'da verilmiştir.



Şekil 6.40. Anapara kazanç grafiği.

7. SONUÇLAR VE ÖNERİLER

Yatırım yapmakta olan ve grafik analizlerine vakit ayırmadan işlem yapmaya çalışan pek çok yatırımcı mevcuttur. Fakat analiz yapmak, piyasa eğilimlerini takip etmede ve saklı fırsatları yakalamada önemli bir araçtır. Eğer bir yatırımcının neler düşündüklerini merak ediyor ve piyasalarda neler olabileceğine dair ön görüşlerde bulunulmak isteniyorsa kesinlikle grafikler ile analizler yapılmalı ve grafik okuma mutlaka öğrenilmelidir. Temelde piyasalar 2 durumdan mutlaka birini göstermektedirler. Bunlar yükseliş eğilimleri ve düşüş eğilimleridir. Hiçbir piyasa enstrümanı sürekli olarak yatay bir eğilimde bulunmaz. Piyasalar bu eğilimler sırasında grafikler üzerinde bazı dönüş anlarında belirli gruplaşma davranışında bulunurlar. Özellikle mum grafikleri üzerinde bu davranışlar oldukça belirgin görülebilir. Bu davranışlar formasyon olarak adlandırılırlar. Bununla birlikte piyasa hareketlerinin matematiksel olarak hesaplanarak incelenmesi sonucunda bazı formüller de piyasaya ait göstergeleri oluştururlar. Bu gösterge ve formasyonlar grafikler üzerinde yatırımcılara işlem yapmadan önce piyasa hareketliliği ve piyasanın sonraki durumu hakkında bazı fikirler verebilir. Tabi bu öngörü tecrübesini kazanmak piyasayı sürekli takip etmek, sürekli yatırım işlemleri yapmak, grafik okumak ve analiz etmekten geçmektedir. Bu bağlamda yatırımcılar yıllar boyunca fiyat grafikleri üzerinde oluşan formasyon ve indikatörleri inceleyerek tekrar tekrar ortaya çıkan kalıpların oluşturduğu formlar sayesinde tecrübeler kazanmışlardır.

Yatırımcılar insanın doğası gereği bu formasyon ve indikatörleri takip etmeye meyillidirler ve genelde piyasalara ya iştahla ya da korkuyla yaklaşırlar. Bu da piyasaların eğilim fark etmeksizin benzer şekilde yanıt vermesine yol açmaktadır. Grafikler üzerinde belirli kalıp yapıların oluşmasındaki temel neden de budur. Bu bağlamda bir yatırım aracının geçmişte görünen eğilim yönlerinden elde edilen tecrübe ile gerçek zamanda tahminler gerçekleştirerek kazançlar elde etmek, grafiklere farklı bir bakış açısı ile bakabilmeyi gerektirir.

Bu çalışma kapsamında neredeyse tüm analizlerin temelini oluşturan sayısal verilerden tamamen uzak durup, sadece finansal zaman serilerinin oluşturduğu 2D mum grafikler üzerinden karar verici bir derin öğrenme tabanlı nesne tanıma modeli tasarlanmıştır.

Çalışma; bir yatırım elemanın gelecek eğilim yönüne karar verecek “*Al-Sat*” karar modelidir. Bu karar işleminin, 2D görsel mum grafik dışında ek herhangi bir zaman serisine gerek kalmaksızın gerçekleştirilmesi hedeflenmiştir. Günümüzde yapay zeka temelli özellikle finansal alanda yapılan çalışmalara bakıldığında uygulama aşamasında biraz geri planda kaldıkları söylenebilir. Özellikle fiyat tahmin uygulamalarında bir sonraki hamleleri tahmin temek için sürekli belirli bir geçmişi eğitime gereksinimi bu çalışmaların dezavantajları olarak söylenebilir. Bu bakımdan kullanılabilirlikleri oldukça düşüktür. Bu çalışma sayesinde geliştirilen model en uygun parametreler ile bir kez eğitilerek mum grafiklerine sahip tüm yatırım enstrümanları için uygulanabilir olmasından dolayı oldukça kullanışlı ve başarılı bir modeldir.

Bu tez kapsamında yenilikçi bir finansal karar destek sistemi olarak tasarlanmış bu model, dört farklı derin öğrenme tabanlı nesne tespit algoritmaları ile eğitilmişlerdir. Bu algoritmalar günümüz derin öğrenme teknolojinde ait en popüler algoritmalarıdır. Oluşturulan modeller üç farklı senaryo yöntemi ile test edilmiştir. Bu test sonuçlarına göre modeller arasından en yüksek başarıya sahip olan model final model olarak belirlenmiştir. Belirlenen final model gerçek zaman verilerinin oluşturulduğu testlerden geçirilmiştir. Bu test sonuçları; final modelin günümüz hisse senedi piyasalarına ait mum grafikleri üzerinde kullanımına uygun olduğunu göstermiştir. Ayrıca sistem bir kez eğitildikten sonra tahmin edilecek yatırım enstrümanı grafiğinden başka farklı hiçbir ek parametre ve bilgiye ihtiyaç duymamaktadır.

Birinci test aşamasında modeller, oluşturulan veri seti içerisindeki test seti başarımlarını değerlendirilmesinden geçirilmişlerdir. Bu değerlendirme işlemi etiketleme sayısı ile modellerin ürettiği karar tespit sayılarının karşılaştırılması ve ayrıca karar tespitlerinin doğruluğunun belirlenmesi yönünde gerçekleştirilmiştir. Bu test değerlendirmelerini YoloV3 VG-2 ve YoloV4 VG-2 algoritmaları %100 başarı sağlamışlardır.

İkinci değerlendirme senaryosu için BIST-100’e dahil olan senetler içerisinde beş farklı senet basit bir rastgele formülü ile seçilmiştir. Altı aylık bir zaman sürecinde seçilen senetlerin mum grafikleri üzerinde modellerin “*Al-Sat*” nesne tespiti yönünde testler yapılmıştır. Nesne tespitleri aşamasında modellerin ürettiği “*Al-Sat*” kararlarına yönelik veriler bir tablo halinde düzenlenmiştir. Ardından modellerin ürettiği “*Al-Sat*” kararları sonucunda her bir senet için ayrılan 100TL tutarındaki toplamda 500TL’lik bir anaparaya ne kadarlık getiri sağlayabildiği değerlendirilmiştir. Yine bu aşamadaki değerlendirme sonucunda tez kapsamında oluşturulan YoloV3 VG-2 modeli en başarılı

sonuca ulařarak en ok getiriyi saęlamıřtır.

Son ařamada nceki testlerden bařarı ile gemiř YoloV3 VG-2 modeli ile bu alıřmaların fikri olarak grlen YoloV3 VG-1 arasında bir karřılařtırma iřlemi gerekleřtirilmiřtir. İki algoritmanın da aynı olduęunu belirtmek gerekir. İki arasındaki fark veri setlerinin oluřturduęu farklılıktır. Kısaca zetlemek gerekirse; VG-1 tam 10000 veri setinden oluřturulmuř ve bu veri seti eęitim-test olarak paylařtırılmıřtır. VG-2’de ise ilave olarak 10000 etiketleme daha gerekleřtirilmiř ve toplamda 20000 etikete ulařılmıřtır. Ardından VG-2 iin 20000 etiket arasından 10000 adet yani ilk etiketlerden kesinlikle farklı olacak karma bir veri seti seilmiřtir. Bu iřlem de yine rastgele gerekleřtirilmiřtir. Bu ařamada ki test iřlemine gelindięinde makale iin testi gerekleřtirilmiř ve tablo haline getirilen tm veriler YoloV3 VG-2 algoritması ile tekrardan test edilmiřtir. Bu son karřılařtırma iřlemi aslında veri iyileřtirmenin sonularının deęerlendirmesi olarak ifade edilebilir. Altı aylık dnem iin elde edilen test sonularına bakıldıęında YoloV3 VG-2 ile YoloV3 VG-1’den kesinlikle daha kazanlı bir sonuca ulařılabildięi yani veri iyileřtirmenin kazanca etki saęladıęı sylenebilir.

Sonu olarak, tez kapsamında belirlenen problem alanına ait zm iin drt farklı algoritma ile modelleme gerekleřtirilmiřtir. Bu modelleme genel literatre bakıldıęında olduka farklı bir bakıř aısına sahip olup olduka yeniliki bir yaklařımdır. Tm modeller ile ayrı ayrı testler gerekleřtirilmiř ve sonuta bir final model bařarısı elde edilmiřtir. Bu final model; gn sonu kapanıř verilerine sahip bir yatırım aracının 2D mum grafikleri zerinde “*Al-Sat*” karar tespiti iin uygulanabilir bir modeldir.

7.1. GELECEK ALIřMALAR

Bu tez alıřması sonrasındaki hedef, hatalı tespitler yapan modeller iin parametre deęiřiklikleri ile eęitim kalitelerini arttırmaktır. Dięer taraftan bu alıřmada etiketli veri seti arttırılmıř olduęundan daha byk veri seti ile eęitim iřlemi gerekleřtirilerek veri seti bymesinin sonuca katkıları deęerlendirilebilir. Bilindięi zere nesne tespit algoritmalarının temel amaları her ne kadar nesnelere doęru tespit etmek olsa da nihai amaları doęru nesnelere en kısa srede (FPS) tespit etmek istemeleridir. Birok algoritmaların karřılařtırmaları da bu yndedir. Fakat bu alıřmada, hızlı tespitin arka

plana atılarak nesnelere daha doğru tespit edilecek, yani algoritmaları hızlı tespitinden ziyade doğru tespitine yönlendirecek parametrik denemeler uygulanacaktır. Son olarak gelişen farklı algoritmalar ile yeni eğitimler gerçekleştirilebilir ve aynı zamanda sadece finansal tabanlı bir algoritma geliştirme çalışmaları gerçekleştirilebilir.



8. KAYNAKLAR

- [1] S. Birogul, G. Temur, ve U. Kose, "YOLO Object Recognition Algorithm and 'Buy-Sell Decision' Model over 2D Candlestick Charts," *IEEE Access*, c. 8, ss. 91894-91915, 2020.
- [2] Ç. Orçun, "Finansal Piyasalarda Alım Satım Kararlarında Teknik Analiz ve İMKB Uygulaması," Yüksek lisans tezi, Finansman Programı, Sosyal Bilimler Enstitüsü, Dokuz Eylül Üniversitesi, İzmir, Türkiye, 2010.
- [3] Y. Taigman, M. Yang, M. Ranzato ve L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, ABD, 2014, ss. 1701-1708.
- [4] F. Schroff, D. Kalenichenko ve J. Philbin, "FaceNet: A Unified Embedding For Face Recognition And Clustering," *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, ABD, 2015, ss. 815-823.
- [5] Y. Sun, X. Wang ve X. Tang, "Deep Learning Face Representation from Predicting 10,000 Classes," *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, ABD, 2014, ss. 1891-1898.
- [6] J. Salamon ve E. Gomez, "Melody Extraction From Polyphonic Music Signals Using Pitch Contour Characteristics," *IEEE Transactions on Audio, Speech, and Language Processing*, c. 20, sayı 6, ss. 1759-1770, 2012.
- [7] C. Hsu, D. Wang, J. R. Jang ve K. Hu, "A Tandem Algorithm for Singing Pitch Extraction and Voice Separation From Music Accompaniment," *IEEE Transactions on Audio, Speech, and Language Processing*, c. 20, sayı 5, ss. 1482-1491, 2012.
- [8] A. Krizhevsky ve G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Proceedings of the 25th International Conference on Neural Information Processing Systems*, Birleşik Krallık, 2012, ss. 1097-1105.
- [9] K. He, X. Zhang, S. Ren ve J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, ABD, 2016, ss. 770-778.
- [10] C. Case, J. Casper, B. Catanzaro, G. Diamos, ve E. Elsen. (2020, 9 Kasım). *Deep Speech : Scaling up end-to-end speech recognition*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1412.5567>.
- [11] A. Graves, N. Jaitly, "Towards End-to-End Speech Recognition with Recurrent Neural Networks," *Proceedings of the 31st International Conference on International Conference on Machine Learning*, China, 2014, ss. 1764-1772.
- [12] J. Weston, N. E. C. L. America, ve I. Way, "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning," *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, Berlin, Almanya, 2008, ss.160-167.

- [13] O. Melamud ve J. Goldberger, “Learning Generic Context Embedding with Bidirectional LSTM,” *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, Berlin, Almanya, 2016, ss. 51-61.
- [14] E. Ben-reuven, J. Goldberger, “Training Deep Neural-Networks Using A Noise Adaptation Layer,” *International Conference on Learning Representations*, Toulon, France, 2015.
- [15] S. Min, B. Lee, ve S. Yoon, “Deep learning in bioinformatics,” *Brief Bioinformatics*, c. 18, sayı 5, ss. 851–869, 2017.
- [16] S. Rönqvist ve P. Sarlin, “Bank distress in the news : Describing events through deep learning,” *Neurocomputing*, c. 264, ss. 57-70, 2017.
- [17] S. Feuerriegel, R. Fehrer. (2020, 09 Kasım). *Improving Decision Analytics With Deep Learning : The Case Of Financial Disclosures*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1508.01993>.
- [18] X. Ding, Y. Zhang, T. Liu, ve J. Duan, “Deep Learning for Event-Driven Stock Prediction,” *In Proceedings of the 24th International Conference on Artificial Intelligence*, Buenos Aires, Argentina, 2015, ss. 2327–2333.
- [19] J. B. Heaton, N. G. Polson, ve J. H. Witte. (2020, 09 Kasım). *Deep Portfolio Theory*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1605.07230>.
- [20] R. Xiong, E. P. Nichols, ve Y. Shen. (2020, 09 Kasım). *Deep Learning Stock Volatility with Google Domestic Trends*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1512.04916>.
- [21] M. Dixon, D. Klabjan, ve J. H. Bang, “Classification-based Financial Markets Prediction using Deep Neural Networks,” *Algorithmic Finance*, ss.1-20, 2016.
- [22] O. B. Sezer, M. U. Gudelek, ve A. M. Ozbayoglu, “Financial time series forecasting with deep learning: A systematic literature review: 2005–2019,” *Applied Soft Computing Journal*, c. 90, sayı 106181, ss. 1-32, 2020.
- [23] S.-C. T. Chou, C.-C. Yang, C.-H. Chan ve F. Lai, "A Rule-Based Neural Stock Trading Decision Support System," *IEEE/IAFE Conference on Computational Intelligence for Financial Engineering*, New York, ABD, 1996, ss. 148-154.
- [24] M. Özçalıcı, “Aşırı Öğrenme Makineleri İle Hisse Senedi Fiyat Tahmini,” *Hacettepe Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, c. 35, sayı 1, ss. 67–67, 2017.
- [25] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, ve R. R. Salakhutdinov. (2020, 09 Kasım). *Improving neural networks by preventing co-adaptation of feature detector*. arXiv.org [Online]. Erişim: <http://arxiv.org/abs/1207.0580>.
- [26] R. Singh ve S. Srivastava, “Stock prediction using deep learning,” *Multimedia Tools and Applications*, c. 76, sayı 18, ss. 18569–18584, 2017.
- [27] W. Bao, J. Yue, ve Y. Rao, “A deep learning framework for financial time series using stacked autoencoders and long- short term memory,” *PLoS ONE*, c. 12, sayı 7, ss. 1-24, 2017.
- [28] H. D. Huynh, L. M. Dang, ve D. Duong, “A New Model For Stock Price Movements Prediction Using Deep Neural Network,” *In Proceedings of the Eighth International Symposium on Information and Communication Technology*, New York, ABD, 2017, ss. 57–627.

- [29] H. Chung ve K. Shin, "Genetic Algorithm-Optimized Long Short-Term Memory Network for Stock Market Prediction," *Sustainability*, c. 10, sayı 3765, ss. 1-18, 2018.
- [30] H. Li, Y. Shen, ve Y. Zhu, "Stock Price Prediction Using Attention-based Multi-Input LSTM," *Proceedings of The 10th Asian Conference on Machine Learning*, Beijing, China, 2018, ss. 454-469.
- [31] S. Roy, A. Unmesh, ve V. P. Namboodiri, "Deep Active Learning For Object Detection," *29th British Machine Vision Conference*, Newcastle, Birleşik Krallık, 2018, ss. 1-12.
- [32] O. B. Sezer ve A. M. Ozbayoglu, "Financial Trading Model With Stock Bar Chart Image Time Series With Deep Convolutional Neural Networks," *Intelligent Automation & Soft Computing Journal*, c. 26, sayı 2, ss. 323-334, 2020.
- [33] A. Soy Temür, M. Akgün, ve G. Temür, "Predicting Housing Sales İn Turkey Using Arima, Lstm And Hybrid Models," *Journal Bussines Economics Managment*, c. 20, sayı 5, ss. 920-938, 2019.
- [34] H. Gündüz, "Derin Öğrenme Yöntemleri İle Zaman Serisi Tahmini," Doktora tezi, Bilgisayar Mühendisliği, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi, İstanbul, Türkiye, 2019.
- [35] A. Hasan, "Derin Öğrenme Ve Makine Öğrenmesi Yöntemleriyle Borsa Alım Satım Davranışlarının Modellenmesi," Doktora tezi, Bilgisayar Mühendisliği, Fen Bilimleri, Yıldız Teknik Üniversitesi, İstanbul, Türkiye, 2020.
- [36] M. S. Rzepczynski, "Analysis of Financial Time Series (a review)," *Financial Analysts Journal*, c. 62, sayı 6, ss. 101-102, 2019.
- [37] E. Zivot, J. Wang, "Nonlinear Time Series Models," *Modeling Financial Time Series with S-PLUS & Springerlink*, c. 978, ss. 653-712, 2006.
- [38] T. W. Liao, "Clustering Of Time Series Data A Survey," *Pattern Recognition*, c. 38, ss. 1857-1874, 2005.
- [39] B. Y. J. D. Hamilton, "A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle," *The Econometric Society Stable*, c. 57, sayı 2, ss. 357-384, 1989.
- [40] O. B. Sezer, "Analysis and Optimization of the Time Series Data with Deep Artificial Neural Networks: Financial Estimation Algorithms," Doktora tezi, Bilgisayar Mühendisliği, Fen Bilimleri Enstitüsü, TOBB Ekonomi ve Teknoloji Üniversitesi, Ankara, 2018.
- [41] B. S. Campus ve H. Road, "A Study On Fundamental And Technical Analysis," *International Journal of Marketing, Financial Services & Management Research*, c. 2, sayı 5, ss. 44-59, 2013.
- [42] University of Cambridge. (2020, 09 Kasım). *Finance Technical Analysis*. University of Cambridge [Online]. Erişim: http://www.mrao.cam.ac.uk/~mph/Technical_Analysis.pdf.
- [43] A. Hedayati, M. Hedayati, ve M. Esfandyari, "Stock Market Index Prediction Using Artificial Neural Network," *Journal of Economics, Finance and Administrative Science*, c. 21, sayı 41, ss. 89-93, 2016.
- [44] Anonim. (2020, 09 Kasım). *Dow Theory*. StockCharts.com [Online]. Erişim:

- https://school.stockcharts.com/doku.php?id=market_analysis:dow_theory.
- [45] B. M. C. Thomsett, *Support and Resistance Simplified*, Birleşik Krallık: Marketplace Books, 2003, ss. 50-58.
- [46] Anonim. (2020, 09 Kasım). *Grafiklerin Dili; Trend, Eğitim Çizgileri, Fiyat Kanalları*. GCM Yatırım Menkul Değerler A.Ş. [Online]. Erişim: <https://www.gcmforex.com/egitim/teknik-analiz/grafiklerin-dili/>.
- [47] P. G. Zhang, "Time Series Forecasting Using A Hybrid ARIMA And Neural Network Model," *Neurocomputing*, c. 50, ss. 159–175, 2003.
- [48] S. Makridakis ve M. Hibon, "ARMA Models And The Box-Jenkins Methodology," *Journal of Forecasting*, c. 16, sayı 3, ss. 147–163, 1997.
- [49] B. Groda ve J. Vrbka, "Prediction Of Stock Price Developments Using The Box-Jenkins Method," *Innovative Economic Symposium*, France, 2017, ss. 1-8.
- [50] T. Nyoni, "Munich Personal RePEc Archive Box-Jenkins ARIMA approach to predicting net FDI inflows in Zimbabwe," Munich Personal RePEc Archive, 2018, sayı 87737, [Online]. Erişim: <https://mpr.ub.uni-muenchen.de/id/eprint/87737>.
- [51] B. G. Andreas, "Experimental Psychology," Birleşik Krallık: John Wiley & Sons, 1960, ss. 333.
- [52] Anonim. (2020, 09 Kasım). *AKSA Grafik*. Yahoo.Finans [Online]. Erişim: <https://finance.yahoo.com/quote/AKSA.IS/chart?p=AKSA.IS> -.
- [53] S. Nison, "Japanese Candlestick Charting Techniques: A Contemporary Guide to the Ancient Investment Techniques of the Far East," ABD: Penguin Putnam Inc., 2001, ss. 287.
- [54] S. Nison, "The Candlestick Course," ABD: John Wiley & Sons, 2003, ss. 121.
- [55] C. Mitchell. (2020, 09 Kasım). *Kagi Chart*. Investopedia [Online]. Erişim: <https://www.investopedia.com/terms/k/kagichart.asp>.
- [56] C. Murphy. (2020, 09 Kasım). *A Look at Kagi Charts*. Investopedia [Online]. Erişim: https://www.investopedia.com/articles/trading/07/kagi_chart.asp.
- [57] Anonim. (2020, 09 Kasım). *AKSUE Grafik*. Yahoo.Finans [Online]. Erişim: <https://finance.yahoo.com/quote/AKSUE.IS/chart?p=AKSUE.IS> -.
- [58] J. G. De Gooijer ve K. Kumar, "Some Recent Developments In Non-Linear Time Series Modelling, Testing, And Forecasting," *International Journal of Forecasting*, c. 8, sayı 2, ss. 135–156, 1992.
- [59] A. Bagnall ve G. Janacek, "Clustering Time Series with Clipped Data," *Machine Learning*, c.58, ss. 151–178, 2005.
- [60] L. Pines. (2020, 09 Kasım). *Head & Shoulder Chart Patterns - Learn Why Traders Use Them So Much*. Commodity.com [Online]. Erişim: <https://commodity.com/technical-analysis/head-shoulders/>.
- [61] A. Heyeş. (2020, 09 Kasım). *Head And Shoulders Pattern Definition*. Investopedia [Online]. Erişim: <https://www.investopedia.com/terms/h/head-shoulders.asp>.
- [62] Anonim. (2020, 09 Kasım). *Cup with Handle*. StockCharts.com [Online]. Erişim: https://school.stockcharts.com/doku.php?id=chart_analysis:chart_patterns:cup_wi

th_handle_continuation.

- [63] J. Chen. (2020, 09 Kasım). *Double Top and Bottom Definition*. Investopedia [Online]. Erişim: <https://www.investopedia.com/terms/d/double-top-and-bottom.asp>.
- [64] C. Mitchell. (2020, 09 Kasım). *Triple Top Definition*. Investopedia [Online]. Erişim: <https://www.investopedia.com/terms/t/tripletop.asp>.
- [65] John J. Murphy, "*Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*," ABD: Penguin Groups, 1999, ss. 32.
- [66] G. Scott. (2020, 09 Kasım). *Wedge Definition*. Investopedia [Online]. Erişim: <https://www.investopedia.com/terms/w/wedge.asp>.
- [67] C. Boyte. (2020, 09 Kasım). *What are the main differences between a Symmetrical Triangle pattern and a pennant?*. Investopedia [Online]. Erişim: <https://www.investopedia.com/ask/answers/013015/what-are-main-differences-between-symmetrical-triangle-pattern-and-pennant.asp>.
- [68] Anonim. (2020, 09 Kasım). *Exhaustion Gap Definition*. Investopedia [Online]. Erişim: <https://www.investopedia.com/terms/e/exhaustiongap.asp>.
- [69] Anonim. (2020, 09 Kasım). *Common Gap Definition*. Investopedia [Online]. Erişim: <https://www.investopedia.com/terms/c/commongap.asp>.
- [70] Anonim. (2020, 09 Kasım). *Breakaway Gap Definition*. Investopedia [Online]. Erişim: <https://www.investopedia.com/terms/b/breakawaygap.asp>.
- [71] T. N. Bulkowski. (2020, 09 Kasım). *Bearish Three-Line Strike*. Thomas N. Bulkowski [Online]. Erişim: <http://thepatternsite.com/ThreeLineStrikeBear.html>.
- [72] T. N. Bulkowski. (2020, 09 Kasım). *Two Black Gapping*. Thomas N. Bulkowski [Online]. Erişim: <http://thepatternsite.com/TwoBlackGapping.html>.
- [73] T. N. Bulkowski. (2020, 09 Kasım). *Three Black Crows*. Thomas N. Bulkowski [Online]. Erişim: <http://thepatternsite.com/ThreeBlackCrows.html>.
- [74] T. N. Bulkowski. (2020, 09 Kasım). *Evening Star*. Thomas N. Bulkowski [Online]. Erişim: <http://thepatternsite.com/EveningStar.html>.
- [75] T. N. Bulkowski. (2020, 09 Kasım). *Bullish Abandoned Baby*. Thomas N. Bulkowski [Online]. Erişim: <http://thepatternsite.com/AbandonBabyBull.html>.
- [76] G. Appel, *Technical Analysis: Power Tools for Active Investors*, Birleşik Krallık: Team Moore, 2005, ss. 106.
- [77] Anonim. (2020, 09 Kasım). *Stop & Reverse: Now Use The Parabolic SAR To Determine Price Direction*. Commodity.com [Online]. Erişim: <https://commodity.com/technical-analysis/parabolic-sar/>.
- [78] Anonim. (2020, 09 Kasım). *Göreceli Güç Endeksi (RSI) — Teknik Göstergeler — Göstergeler ve Sinyaller*. TradingView [Online]. Erişim: <https://tr.tradingview.com/scripts/relativestrengthindex/>.
- [79] Anonim. (2020, 09 Kasım). *Emtia Kanal Endeksi (CCI) — Teknik Göstergeler — Göstergeler ve Sinyaller*. TradingView [Online]. Erişim: <https://tr.tradingview.com/scripts/commoditychannelindex/>.
- [80] Anonim. (2020, 09 Kasım). *Stokastik Osilatör — Teknik Göstergeler — Göstergeler ve Sinyaller*. TradingView [Online]. Erişim:

<https://tr.tradingview.com/scripts/stochastic/>.

- [81] J. Bernstein, *The Compleat Day Trader: Trading Systems, Strategies, Timing Indicators and Analytical Methods*, Birleşik Krallık: McGraw-Hill Companies, 1995, ss. 176.
- [82] A. Hayes. (2020, 09 Kasım). *Using stochastics, cycles & R.S.I*. Investopedia [Online]. Erişim: <https://www.investopedia.com/terms/s/stochasticoscillator.asp>.
- [83] Anonim. (2020, 09 Kasım). *Bollinger Bantları (BB) — Teknik Göstergeler*. TradingView [Online]. Erişim: <https://tr.tradingview.com/ideas/bollingerbands/>.
- [84] R. R. Prechter, “Elliott Waves, Fibonacci and Statistics,” *Focus (Madison)*, ss. 254–258, 2005.
- [85] R. N. Elliott. (2020, 09 Kasım). *Elliott Dalga Prensibi*. Wikipedia [Online]. Erişim: https://tr.wikipedia.org/wiki/Elliott_dalga_prensibi.
- [86] S. W. Poser. (2020, 09 Kasım). *Elliott Wave Principle*. Wikipedia [Online]. Erişim: https://en.wikipedia.org/wiki/Elliott_wave_principle.
- [87] J. L. Person, *Candlestick and Pivot Point Trading Triggers: Setups for Stock, Forex, and Futures Markets*, New Jersey: Wiley, 2007, ss. 183.
- [88] J. Schmidhuber, “Deep Learning in neural networks: An overview,” *Neural Networks*, c. 61. ss. 85–117, 2015.
- [89] Y. Lecun, Y. Bengio, ve G. Hinton, “Deep learning,” *Nature*, c. 521, ss. 436-444, 2015.
- [90] Y. Bengio, A. Courville, ve P. Vincent, “Representation Learning : A Review and New Perspectives,” *IEEE Transactions on Software Engineering*, c. 35, sayı 8, ss. 1798-1828, 2013.
- [91] Y. Bengio, “Learning Deep Architectures for AI,” *Foundations and Trends in Machine Learning*, c. 2, sayı 1, ss. 1-55, 2009.
- [92] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen ve A. A. S. Awwal, “A State-Of-The-Art Survey On Deep Learning Theory And Architectures,” *Electron*, c. 8, sayı 3, ss. 1–67, 2019.
- [93] B. Of ve W. Pitts, “How We Know Universals,” *Bulletin of Mathematical Biophysics*, c. 9, ss. 127–147, 1947.
- [94] D. E. Rumelhart ve J. L. McClelland, "Learning Internal Representations by Error Propagation," *MIT Press*, ss. 318-362, 1987.
- [95] G. E. Hinton, R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” *Science*, c. 313, sayı 5786, ss. 504-507, 2006.
- [96] S. Furui, L. Deng, M. Gales, H. Ney, ve K. Tokuda, “Fundamental technologies in modern speech recognition,” *IEEE Signal Processing Magazine*, c. 29, sayı 6, ss. 16–17, 2012.
- [97] J. Deng, W. Dong, R. Socher, L. Li, K. Li, ve L. Fei-fei, “ImageNet : A Large-Scale Hierarchical Image Database,” *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, ABD, 2009, ss. 248-255.
- [98] M. Seltzer, D. Yu, ve A. Mohamed, “Binary Coding Of Speech Spectrograms Using A Deep Auto-Encoder Binary Coding Of Speech Spectrograms Using A Deep Auto - Encoder,” *11th Annual Conference of the International Speech Communication Association*, Japan, 2010, ss. 1692-1695.

- [99] G. E. Dahl, M. A. Ranzato, A. Mohamed, ve G. Hinton, "Phone Recognition with the Mean-Covariance Restricted Boltzmann Machine," *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, Birleşik Krallık, 2010, ss. 469-477.
- [100] C. Szegedy ve S. G. Com, "Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift," *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, Fransa, 2015, ss. 448-456.
- [101] P. Sermanet ve D. Eigen. (2020, 09 Kasım). *OverFeat : Integrated Recognition , Localization and Detection using Convolutional Networks*. arXiv.org [Online]. Erişim: <https://arxiv.org/pdf/1312.6229.pdf>.
- [102] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, ve V. K. Asari, "Going Deeper with Convolutions," *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, ABD, 2015, ss. 1-9.
- [103] K. Simonyan ve A. Zisserman, "Very Deep Convolutional Networks For Large-Scale Image Recognition," *3rd International Conference on Learning Representations*, San Diego, ABD, 2015, ss. 1–14.
- [104] K. He, X. Zhang, S. Ren, ve J. Sun, "Deep Residual Learning For Image Recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, ABD, 2016, ss. 770–778.
- [105] J. Heaton, "Ian Goodfellow , Yoshua Bengio , and Aaron Courville: Deep learning : The MIT," *Genetic Programming and Evolvable Machines*, c. 19, sayı 1, ss. 305–307, 2018.
- [106] G. E. Hinton ve S. Osindero, "A Fast Learning Algorithm For Deep Belief Nets * 500 Units 500 Units," *Neural Computation*, c. 18, sayı 7, ss. 1524-1554, 2006.
- [107] G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ABD, 2010, ss. 807–814.
- [108] Y. Bengio ve P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, c. 86, sayı 11, ss. 2278-2324, 1998.
- [109] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg ve L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, c. 115, sayı 3, ss. 211–252, 2015.
- [110] G. Işık ve H. Artuner, "Radyo Sinyallerinin Derin Öğrenme Sinir Ağları ile Tanınması," *24th Signal Processing and Communication Application Conference, SIU 2016 - Proceedings*, ABD, 2016, ss. 837–840.
- [111] L. Claesson ve B. Hansson, "Deep Learning Methods and Applications Classification of Traffic Signs and Detection of Alzheimer's Disease from Images," Yüksek lisans tezi, Communication Engineering and Biomedical Engineering, Department of Signals and Systems, Chalmers University Of Technology, Gothenburg, Sweden, 2017.
- [112] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, ve R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting,"

Journal Machine Learning Research, c. 15, ss. 1929–1958, 2014.

- [113] M. Lin, Q. Chen, ve S. Yan. (2020, 09 Kasım). *Network In Network*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1312.4400>.
- [114] J. T. Springenberg, A. Dosovitskiy, T. Brox, ve M. Riedmiller. (2020, 09 Kasım). *Striving for Simplicity: The All Convolutional Net*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1412.6806>.
- [115] G. Huang, Z. Liu, L. Van Der Maaten, ve K. Q. Weinberger, “Densely Connected Convolutional Networks,” *IEEE Conference on Computer Vision and Pattern Recognition*, ABD, 2017, ss. 2261-2269.
- [116] G. Larsson, M. Maire, ve G. Shakhnarovich. (2020, 09 Kasım). *FractalNet: Ultra-Deep Neural Networks without Residuals*. arXiv.org [Online]. Erişim: <http://arxiv.org/abs/1605.07648>.
- [117] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, ve Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Las Vegas, ABD, 2016, ss. 2818–2826.
- [118] C. Szegedy, S. Ioffe, V. Vanhoucke, ve A. A. Alemi, “Inception-V4, Inception-Resnet And The Impact Of Residual Connections On Learning,” *31st AAAI Conference on Artificial Intelligence*, California, ABD, 2017, ss. 4278–4284,.
- [119] M. D. Zeiler ve R. Fergus, “Visualizing and Understanding Convolutional Networks,” *European Conference on Computer Vision*, ABD, 2014, ss. 818-833.
- [120] S. Ren, K. He, R. Girshick, ve J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, c. 39, sayı 6, ss. 1137-1149, 2017.
- [121] C. Google, “Xception: Deep Learning with Depthwise Separable Convolutions,” *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, ABD, 2017, ss. 1800-1807.
- [122] M. Liang ve X. Hu, “Recurrent Convolutional Neural Network For Object Recognition,” *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, ABD, 2015, ss. 3367-3375.
- [123] Y. Huang, X. Li, C. Yan, L. Liu, ve H. Dai, “MIRD-Net for Medical Image Segmentation,” *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Singapore, 2020, ss. 207–219.
- [124] Y. Li, W. Ouyang, X. Wang, ve X. Tang, “ViP-CNN: Visual Phrase Guided Convolutional Neural Network,” *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, ABD, 2017, ss. 7244-7253.
- [125] M. Rastegari, “LCNN: Lookup-based Convolutional Neural Network,” *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, ABD, 2017, ss. 860-869.
- [126] J. Long, E. Shelhamer, ve T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, c. 39, sayı 4, ss. 640-651, 2017.
- [127] X. Chen ve B. Russell. (2020, 09 Kasım). *PixelNet: Representation of The Pixels, by The Pixels, and for The Pixels*. arXiv.org [Online]. Erişim:

<https://arxiv.org/abs/1702.06506>.

- [128] G. Huang, Y. Sun, Z. Liu, D. Sedra, ve K. Q. Weinberger, “Deep Networks with Stochastic Depth,” *European Conference on Computer Vision*, Amsterdam, Hollanda, 2016, ss. 646-661.
- [129] C. Lee, S. Xie, ve P. W. Gallagher. (2020, 09 Kasım). *Deeply-Supervised Nets*. arXiv [Online]. Erişim: <https://arxiv.org/abs/1409.5185>.
- [130] M. Pezeshki, L. Fan, A. Courville, ve Y. Bengio, "Deconstructing the Ladder Network Architecture," *Proceedings of Machine Learning Research*, c. 48, ss. 2368-2376, 2016.
- [131] Waseem Rawat ve Z. Wang, “Deep Convolutional Neural Networks for ImageClassification: A Comprehensive Review,” *Neural Computation*, c. 29, sayı 9, ss. 2352–2449, 2017.
- [131] E. Tzeng, J. Hoffman, T. Darrell, U. C. Berkeley, K. Saenko, ve U. Lowell, “Simultaneous Deep Transfer Across Domains and Tasks,” *Domain Adaptation in Computer Vision Applications*, Meylan, France: Springer, 2017, ss. 173-180.
- [133] M. L. Mar, A. Mohamed, M. Philipose, M. Richardson, ve R. Caruana. (2020, 09 Kasım). *Do Deep Convolutional Nets Really Need to be Deep and Convolutional ?*. arXiv [Online]. Erişim: <https://arxiv.org/abs/1603.0569>.
- [134] L. J. Ba ve R. Caruana, “Do Deep Nets Really Need to be Deep ?,” *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Thailand, 2014, ss. 2654-2662.
- [135] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, ve Y. Bengio, “FitNets: Hints for thin deep nets,” *3rd International Conference on Learning Representations*, San Diego, ABD, 2015, ss. 1–13.
- [136] D. Mishkin ve J. Matas. (2020, 09 Kasım). All you need is a good init. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1511.06422>.
- [137] G. Pandey ve A. Dukkipati. (2020, 09 Kasım). *To go deep or wide in learning?*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1402.5634>.
- [138] A. Ratner, C. De Sa, S. Wu, D. Selsam, ve C. Ré, “Data Programming : Creating Large Training Sets , Quickly,” *Proceedings of the 30th International Conference on Neural Information Processing Systems*, New York, ABD, 2016, ss. 3574–3582.
- [139] C. R. Aberger, S. Tu, K. Olukotun, ve C. Ré, “EmptyHeaded: A Relational Engine For Graph Processing,” *Proceedings of the ACM SIGMOD International Conference on Management of Data*, New York, ABD, 2016, ss. 431–446.
- [140] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, ve K. Keutzer. (2020, 09 Kasım). *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size*. arXiv.org [Online]. Erişim: <http://arxiv.org/abs/1602.07360>.
- [141] S. Han, H. Mao ve W. J. Dally, “Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding,” *4th International Conference on Learning Representations*, San Juan, Porto Rico. 2016, ss.1-14.
- [142] M. Niepert, K. Kutzkov, K. Kutzkov, ve N. Eu. (2020, 09 Kasım). *Learning*

- Convolutional Neural Networks for Graphs*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1605.05273>.
- [143] X. Jia, X. Xu, B. Cai, ve K. Guo, “Single Image Super-Resolution Using Multi-Scale Convolutional Neural Network,” *Advances in Multimedia Information Processing*, Çin: Springer, 2018, ss. 149-157.
- [144] B. Ahn, N. I. Cho, ve S. Member. (2020, 09 Kasım). *Block-Matching Convolutional Neural Network for Image Denoising*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1704.00524>.
- [145] S. Ma, J. Liu, ve C. W. Chen, “A-Lamp: Adaptive Layout-Aware Multi-patch Deep Convolutional Neural Network for Photo Aesthetic Assessment,” *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, ABD, 2017, ss. 722-731.
- [146] X. Cao, F. Zhou, L. Xu, D. Meng, Z. Xu, ve J. Paisley, “Hyperspectral Image Classification With Markov Random Fields and a Convolutional Neural Network,” *IEEE Transactions on Image Processing*, c. 27, sayı 5, ss. 2354–2367, 2018.
- [147] B. D. De Vos, F. F. Berendsen, M. A. Viergever, ve M. Staring. (2020, 09 Kasım). *End-to-End Unsupervised Deformable Image Network*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1704.06065>.
- [148] X. Wang, G. Oxholm, D. Zhang ve Y. Wang, "Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer," *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, ABD, 2017, ss. 7178-7186.
- [149] M. Babae, D. Dinh, ve G. Rigoll, “A Deep Convolutional Neural Network for Background Subtraction,” *Pattern Recognition*, c.76, ss. 65-649, 2017.
- [150] Z. Alom, P. Sidike, M. Hasan, T. M. Taha, ve V. K. Asari. (2020, 09 Kasım). *Handwritten Bangla Character Recognition Using The State-of-Art Deep Convolutional Neural Networks*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1712.09872>.
- [151] Z. Alom, A. A. S. Awwal, R. Lowe-webb, ve T. M. Taha, “Optical beam classification using deep learning : a comparison with rule- and feature-based classification,” *Optics and Photonics for Information Processing XI*, California, ABD, 2017, ss.1039500.
- [152] P. Sidike, V. Sagan, M. Maimaitijiang, M. Maimaitiyiming, N. Shako, J. Burken, T. Mockler, F. B. Fritschi, “dPEN: deep Progressively Expanded Network for mapping heterogeneous agricultural landscape using WorldView-3 satellite imagery,” *Remote Sensing of Environment*, c. 221, ss. 756–772, 2019.
- [153] Z. Alom, M. Alam, T. M. Taha, ve K. M. Iftekharuddin, “Object Recognition using Cellular Simultaneous Recurrent Networks and Convolutional Neural Network,” *International Joint Conference on Neural Networks*, Anchorage, ABD, 2017, ss. 2873-2880.
- [154] C. A. Ronao ve S. Cho, “Human Activity Recognition With Smartphone Sensors Using Deep Learning Neural Networks,” *Expert Systems with Applications*, c. 59, ss. 235–244, 2016.
- [155] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, ve S. Krishnaswamy, “Deep

- Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition,” *Proceedings of the 24th International Conference on Artificial Intelligence*, Buenos Aires, Arjantin, 2015, ss. 3995-4001.
- [156] N. Y. Hammerla, S. Halloran, ve T. Plötz, “Deep, Convolutional, And Recurrent Models For Human Activity Recognition Using Wearables,” *Proceedings of the 25th International Conference on Artificial Intelligence*, New York, ABD, 2016, ss.1533-1540.
- [157] F. J. Ordóñez ve D. Roggen, “Deep Convolutional and LSTM Recurrent Activity Recognition,” *Sensors*, c. 16, sayı 115, ss. 1-25, 2016.
- [158] N. M. Rad, S. M. Kia, C. Zarbo, T. Laarhoven, G. Jurman, P. Venuti, E. Marchiori ve C. Furlanello, “Deep Learning For Automatic Stereotypical Motor Movement Detection Using Wearable Sensors In Autism Spectrum Disorders,” *Signal Processing*, c. 144, ss. 180–191, 2018.
- [159] D. Rav, C. Wong, B. Lo, ve G. Yang, "Deep Learning For Human Activity Recognition: A Resource Efficient Implementation On Low-Power Devices," *IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks*, San Francisco, ABD, 2016, ss. 71-76.
- [160] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy ve A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, c. 40, sayı 4, ss. 834-848, 2018.
- [161] V. Badrinarayanan, A. Kendall, R. Cipolla, ve S. Member, “SegNet: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, c. 39, sayı 12, ss. 2481-2495, 2017.
- [162] G. Lin, A. Milan, C. Shen, ve I. Reid, “RefineNet: Multi-Path Refinement Networks For High-Resolution Semantic Segmentation,” *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, ABD, 2017, ss. 5168–5177.
- [163] H. Zhao, J. Shi, X. Qi, X. Wang ve J. Jia, "Pyramid Scene Parsing Network," *IEEE Conference on Computer Vision and Pattern Recognition* , Honolulu, ABD, 2017, ss. 6230-6239.
- [164] O. Ronneberger, P. Fischer, ve T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Munich, Almanya, 2015, ss. 234-241.
- [165] M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, ve K. A. Vijayan. (2020, 09 Kasım). *Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1802.06955>.
- [166] R. Girshick, J. Donahue, T. Darrell, U. C. Berkeley, ve J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, ABD, 2014, ss. 580-587.
- [167] T. Lin, P. Goyal, R. Girshick, K. He ve P. Dollár, "Focal Loss for Dense Object

- Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, c. 42, sayı 2, ss. 318-327, 2020.
- [168] X. Wang, "A-Fast-RCNN: Hard Positive Generation via Adversary for Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, ABD, 2017, ss. 3039-3048.
- [169] K. He, G. Gkioxari, P. Dollár ve R. Girshick. (2020, 09 Kasım). *Mask R-CNN*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1703.06870>, 2017.
- [170] J. Redmon, S. Divvala, R. Girshick, ve A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, ABD, 2016, ss. 779-788.
- [171] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu ve A. C. Berg, "SSD: Single Shot MultiBox Detector," *European Conference on Computer Vision*, Amsterdam, Hollanda, 2016, ss. 21-37.
- [172] Z. Alom, C. Yakopcic, T. M. Taha, ve V. K. Asari. (2020, 09 Kasım). *Microscopic Nuclei Classification, Segmentation and Detection with improved Deep Convolutional Neural Network (DCNN) Approaches*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1811.03447>.
- [173] J. Hou, S. Wang, Y. Lai, Y. Tsao, H. Chang ve H. Wang, "Audio-Visual Speech Enhancement Using Multimodal Deep Convolutional Neural Networks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, c. 2, sayı 2, ss. 117-128, 2018.
- [174] Y. Xu, Q. Kong, Q. Huang, W. Wang ve M. D. Plumbley, "Convolutional gated recurrent neural network incorporating spatial features for audio tagging," *International Joint Conference on Neural Networks*, Anchorage, ABD, 2017, ss. 3461-3466.
- [175] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A.W.M. van-der-Laak, B. van-Ginneken ve C. I. Sánchez, "A Survey On Deep Learning In Medical Image Analysis," *Medical Image Analysis*, c. 42, ss. 60–88, 2017.
- [176] Z. Zhang, Y. Xie, F. Xing, M. MCGough, ve L. Yang. (2020, 11 Kasım). *MDNet: A Semantically and Visually Interpretable Medical Image Diagnosis Network*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1604.00494>.
- [177] B. A. Hamilton. (2020, 11 Kasım). *A Fully Convolutional Neural Network for Cardiac Segmentation in Short-Axis MRI*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1909.00823>, 2019.
- [178] J. Hong, U. R. Acharya, S. V Bhandary, ve K. Chua, "Segmentation Of Optic Disc, Fovea And Retinal Vasculature Using A Single Convolutional Neural Network," *Journal of Computational Science*, c. 20, ss. 70–79, 2017.
- [179] P. Moeskops, M. A. Viergever, A. M. Mendrik, L. S. Vries, M. J. N. L. Benders ve I. Išgum, "Automatic Segmentation of MR Brain Images With a Convolutional Neural Network," *IEEE Transactions on Medical Imaging*, c. 35, sayı 5, ss. 1252-1261, 2016.
- [180] M. Z. Alom, C. Yakopcic, T. M. Taha ve V. K. Asari, "Microscopic Blood Cell Classification Using Inception Recurrent Residual Convolutional Neural Networks," *IEEE National Aerospace and Electronics Conference*, Dayton,

ABD, 2018, ss. 222-227.

- [181] M. Z. Alom, C. Yakopcic, T. M. Taha ve V. K. Asari, "Nuclei Segmentation with Recurrent Residual Convolutional Neural Networks based U-Net (R2U-Net)," *IEEE National Aerospace and Electronics Conference*, Dayton, ABD, 2018, ss. 228-233.
- [182] M. Z. Alom, C. Yakopcic, T. M. Taha, ve V. K. Asari, "Breast Cancer Classification from Histopathological Images with Inception Recurrent Residual Convolutional Neural Network," *Journal of Digital Imaging*, c. 32, ss. 605–617, 2019.
- [183] Anonim. (2020, 10 Kasım). *ImageNet*. Stanford University, Princeton University [Online]. Erişim: <http://www.image-net.org/>.
- [184] L. Y. Pratt, "Discriminability-Based Transfer between Neural Networks," *Advances in Neural Information Processing Systems*, ABD, 1992, ss. 204–211.
- [185] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, c. 60, ss. 91–110, 2004.
- [186] H. Bay, A. Ess, T. Tuytelaars, ve L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, c. 110, sayı 3, ss. 346-359, 2008.
- [187] M. Calonder, V. Lepetit, C. Strecha, ve P. Fua, "BRIEF: Binary Robust Independent Elementary Features," *European Conference of Computer Vision*, Berlin, Almanya: Springer, 2010, ss. 778-792.
- [188] C. Borngrund, "Machine Vision For Automation Of Earth-Moving Machines : Transfer Learning Experiments With YOLOv3," Yüksek lisans tezi, Computer Science and Engineering, Luleå University of Technology, İsveç, 2019.
- [189] J. Brownlee. (2020, 11 Kasım). *A Gentle Introduction to Computer Vision*. Jason Brownlee [Online]. Erişim: <https://machinelearningmastery.com/what-is-computer-vision/>.
- [190] Anonim. (2020, 11 Kasım). What is Computer Vision?. Techopedia. [Online]. Erişim: <https://www.techopedia.com/definition/32309/computer-vision>.
- [191] Simon J.D. Prince, "Computer Vision: Models, Learning And Inference," England: Cambridge University Press, 2012, ss. 176.
- [192] T. Celik ve H. Kusetogullari, "Solar-Powered Automated Road Surveillance System for Speed Violation Detection," *IEEE Transactions on Industrial Electronics*, c. 57, sayı 9, ss. 3216-3227, 2010.
- [193] H. Kusetogullari, T. Celik, ve B. Data, "Real Time Vehicle Detection And Driver Warning System Using Computer Vision," *Electronics World*, c. 116, sayı 1885, ss. 24–27, 2010.
- [194] M. F. Demir, A. Cankirli, B. Karabatak, A. Yavariabdi, E. Mendi ve H. Kusetogullari, "Real-Time Resistor Color Code Recognition Using Image Processing in Mobile Devices," *International Conference on Intelligent Systems*, Funchal - Madeira, Portekiz, 2018, ss. 26-30.
- [195] A. Cheddad, H. Kusetogullari ve H. Grahn, "Object Recognition Using Shape Growth Pattern," *Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis*, Ljubljana, Slovenya, 2017, ss. 47-52.

- [196] F. G. Yaşar ve H. Kusetogullari, "Underwater Human Body Detection Using Computer Vision Algorithms," *26th Signal Processing and Communications Applications Conference*, Izmir, Türkiye, 2018, ss. 1-4.
- [197] H. Kusetogullari, H. Demirel, T. Celik, ve N. S. Bayindir, "Real Time Detection And Tracking Of Vehicles For Speed Measurement And License Plate Detection," *The Seventh IASTED International Conference on Visualization, Imaging and Image Processing*, ABD, 2007, ss. 53-58.
- [198] N. Sebe, I. Cohen, A. Garg, ve T. S. Huang, "Office Activity Recognition," *Machine Learning in Computer Vision*, Netherlands: Springer, 2005, ss. 157-173.
- [199] M. Everingham, S. M. Eslami, L. Gool, C. K. Williams, J. Win ve A. Zisserman, "The PASCAL Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision*, ABD, 2015, ss. 98–136.
- [200] J. C. P. Cheng ve M. Wang, "Automated Detection Of Sewer Pipe Defects In Closed-Circuit Television Images Using Deep Learning Techniques," *Automation in Construction*, c. 95, ss. 155–171, 2018.
- [201] K. He, X. Zhang, S. Ren ve J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, c. 37, sayı 9, ss. 1904-1916, 2015.
- [202] R. Girshick, "Fast R-CNN," *IEEE International Conference on Computer Vision*, Santiago, Şili, 2015, ss. 1440-1448.
- [203] Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, ve V. K. Asari, "Improved Inception-Residual Convolutional Neural Network for Object Recognition," *Neural Computing & Application*, c. 32, ss. 279-293, 2020.
- [204] T. Lin, C. L. Zitnick, ve P. Doll, "Microsoft COCO: Common Objects in Context," *European Conference on Computer Vision*, Zurich, İsviçre, 2014, ss. 740-755.
- [205] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu ve M. Pietikäinen, "Deep Learning for Generic Object Detection: A Survey," *International Journal of Computer Vision*, c. 128, ss. 261–318, 2020.
- [206] J. Redmon ve A. Farhadi. (2020, 09 Kasım). *YOLOv3*. Joseph Chet Redmon [Online]. Erişim: <https://pjreddie.com/media/files/papers/YOLOv3.pdf>.
- [207] A. Bochkovskiy, C.-Y. Wang, ve H.-Y. M. Liao. (2020, 7 Kasım). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. arXiv.org [Online]. Erişim: <http://arxiv.org/abs/2004.10934>.
- [208] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, ve A. Torralba. (2020, 7 Kasım). *Object detectors emerge in deep scene CNNs*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1412.6856>.
- [209] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva ve A. Torralba, "Learning Deep Features for Discriminative Localization," *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, ABD, 2016, ss. 2921-2929.
- [210] R. G. Cinbis, J. Verbeek ve C. Schmid, "Weakly Supervised Object Localization with Multi-Fold Multiple Instance Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, c. 39, sayı 1, ss. 189-203, 2017.

- [211] M. Oquab, "Is object localization for free? – Weakly-supervised learning with convolutional neural networks," *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, ABD, 2015, ss. 685-694.
- [212] R. Girshick, J. Malik, B. Hariharan, ve P. Arbel, "Object Instance Segmentation and Fine-Grained Localization Using Hypercolumns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, c. 39, sayı 4, ss. 627-639, 2017.
- [213] A. Shrivastava, R. Sukthankar, J. Malik, ve A. Gupta. (2020, 6 Kasım). *Beyond Skip Connections: Top-Down Modulation for Object Detection*. arXiv.org [Online]. Erişim: <http://arxiv.org/abs/1612.06851>.
- [214] S. Ren, K. He, R. Girshick, ve J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, c. 39, sayı 6, ss. 1137-1149, 2017.
- [215] K. Lenc, A. Vedaldi, "R. R-cnn, "R-CNN minus R," *Proceedings of the British Machine Vision Conference*, Swansea, Birleşik Krallık, 2015, ss. 5.1-5.12.
- [216] J. Redmon, S. Divvala, R. Girshick ve A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, ABD, 2016, ss. 779-788.
- [217] Y. Luo, S. Li, K. Sun, R. Renteria ve K. Choi, "Implementation of deep learning neural network for real-time object recognition in OpenCL framework," *International SoC Design Conference*, Seoul, Güney Kore, 2017, ss. 298-299.
- [218] G. Alpaydin, "An Adaptive Deep Neural Network for Detection, Recognition of Objects with Long Range Auto Surveillance," *IEEE 12th International Conference on Semantic Computing*, Laguna Hills, ABD, 2018, ss. 316-317.
- [219] Daniel Maturana ve S. Scherer, "VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, Almanya, 2015, ss. 922-928.
- [220] S. Hong, B. Roh, K.-H. Kim, Y. Cheon, ve M. Park. (2020, 09 Kasım). *PVANet: Lightweight Deep Neural Networks for Real-time Object Detection*. arXiv.org [Online]. Erişim: <http://arxiv.org/abs/1611.08588>.
- [221] R. F. C. Networks ve J. Dai, "Object Detection via Region-based Fully Convolutional Networks," *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ABD, 2016, ss. 379-387.
- [222] T. Kong, A. Yao, Y. Chen, ve F. Sun, "HyperNet: Towards accurate region proposal generation and joint object detection," *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, ABD, 2016, ss. 845-853.
- [223] J. Redmon ve A. Farhadi, "YOLO9000: Better, Faster, Stronger," *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, ABD, 2017, ss. 6517-6525.
- [224] K. He, X. Zhang, S. Ren ve J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *IEEE International Conference on Computer Vision*, Santiago, Şili, 2015, ss. 1026-1034.
- [225] S. Xie, R. Girshick, P. Dollár, Z. Tu ve K. He, "Aggregated Residual Transformations for Deep Neural Networks," *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, ABD, 2017, ss. 5987-5995.

- [226] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan ve S. Belongie, "Feature Pyramid Networks for Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, ABD, 2017, ss. 936-944.
- [227] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, ve J. Sun. (2020, 6 Kasım). *Light-Head R-CNN: In Defense of Two-Stage Object Detector*. arXiv.org [Online]. Erişim: <https://arxiv.org/abs/1711.07264>.
- [228] J. R. R. Uijlings, T. Gevers, ve A. W. M. Smeulders, "Selective Search for Object Recognition," *International Journal of Computer Vision*, *International Journal of Computer Vision*, c. 104, sayı 4, ss. 154-171, 2013.
- [229] R. Girshick, J. Donahue, T. Darrell ve J. Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, c. 38, sayı 1, ss. 142-158, 2016.
- [230] S. Pokhrel. (2020, 09 Kasım). *Image Data Labelling and Annotation — Everything you need to know*. Towards Data Science [Online]. Erişim: <https://towardsdatascience.com/image-data-labelling-and-annotation-everything-you-need-to-know-86ede6c684b1>.
- [231] T.Y. Lin, G. Patterson, M.R. Ronchi, Y. Cui, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, L. Zitnick, P. Dollar. (2020, 10 Kasım). *COCO - Common Objects in Context*. COCO Consortium [Online]. Erişim: <https://cocodataset.org/#home>.
- [232] Y. Aytar, A. Eslami, A. Sorokin. (2020, 09 Kasım). *The PASCAL Visual Object Classes* [Online]. Erişim: <http://host.robots.ox.ac.uk/pascal/VOC/index.html>.
- [233] Anonim. (2020, 7 Kasım). *YOLO Darknet TXT Annotation Format*. Roboflow [Online]. Erişim: <https://roboflow.com/formats/yolo-darknet-txt>.
- [234] A. Rosebrock. (2020, 7 Kasım). *Intersection over Union (IoU) for object detection-PyImageSearch*. Adrian Rosebrock [Online]. Erişim: <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>.
- [235] Anonim. (2020, 8 Kasım). *Conformant Products - The Khronos Group Inc*. Khronosgroup [Online]. Erişim: <https://www.khronos.org/conformance/adopters/conformant-products#opencl>.
- [236] Anonim. (2020, 8 Kasım). *Deep Learning | NVIDIA Developer*. NVIDIA [Online]. Erişim: <https://developer.nvidia.com/deep-learning>.
- [237] Anonim. (2020, 7 Kasım). *How Intel® Xeon Phi™ Processors Benefit Machine Learning/Deep*. Intel [Online]. Erişim: <https://software.intel.com/content/www/us/en/develop/blogs/how-xeon-phi-processors-benefit-machine-and-deep-learning-apps-frameworks.html>.
- [238] Anonim. (2020, 7 Kasım). *Google's New Chip Makes Machine Learning Way Faster*. Computerworld [Online]. Erişim: <https://www.computerworld.com/article/3072652/googles-new-chip-makes-machine-learning-way-faster.html>.
- [239] Anonim. (2020, 9 Kasım). *Deep Learning and Machine Intelligence Solutions*. AMD [Online]. Erişim: <https://www.amd.com/en/graphics/servers-radeon-instinct-deep-learning>.
- [240] Anonim. (2020, 10 Kasım). *CUDA Zone | NVIDIA Developer*. NVIDIA

- [Online]. Erişim: <https://developer.nvidia.com/cuda-zone>.
- [241] Anonim. (2020, 11 Kasım). *CUDA*. Wikipedia [Online]. Erişim: https://en.wikipedia.org/wiki/CUDA#cite_note-CUDA_intro_-_TomsHardware.
- [242] S. A. Manavski ve G. Valle, “*CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment*,” *BMC Bioinformatics*, c. 9, sayı 10, ss. 2-10, 2008.
- [243] Anonim. (2020, 7 Kasım). *NVIDIA cuDNN | NVIDIA Developer*. NVIDIA [Online]. Erişim: <https://developer.nvidia.com/cudnn>.
- [244] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, ve X. Zheng. (2020, 7 Kasım). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. TensorFlow.org [Online]. Erişim: <https://www.tensorflow.org/overview>.
- [245] Anonim. (2020, 9 Kasım). *Intel acquires Itseez*. Intel [Online]. Erişim: <https://opencv.org/intel-acquires-itseez/>.
- [246] Anonim. (2020, 11 Kasım). *OpenCV*. Wikipedia [Online]. Erişim: <https://tr.wikipedia.org/wiki/OpenCV>.
- [247] K. Pulli, A. Baksheev, K. Korniyakov, ve V. Eruhimov, “Realtime computer vision with OpenCV,” *Communications of the ACM*, c. 55, sayı 6, ss. 61-69, 2012.
- [248] Anonim. (2020, 11 Kasım). *NumPy*. Wikipedia [Online]. Erişim: <https://en.wikipedia.org/wiki/NumPy>.
- [249] J. Hunter,. (2020, 11 Kasım). *Usage — Matplotlib 2.0.2 documentation* [Online]. Erişim: https://matplotlib.org/faq/usage_faq.html#coding-styles.
- [250] Anonim. (2020, 11 Kasım). *Matplotlib*. Wikipedia [Online]. Erişim: https://en.wikipedia.org/wiki/Matplotlib#cite_note-3.
- [251] T. Talin. (2020, 11 Kasım). *GitHub - tzutalin/labelImg: LabelImg is a graphical image annotation tool and label object bounding boxes in images*. GitHub [Online]. Erişim: <https://github.com/tzutalin/labelImg>.
- [252] D. Zuenko. (2020, 11 Kasım). *Preparing data before training YOLO v2 and v3. #DeepFashion dataset*. Denis Zuenko [Online]. Erişim: <https://blog.goodaudience.com/part-1-preparing-data-before-training-yolo-v2-and-v3-deepfashion-dataset-3122cd7dd884>.
- [253] Anonim. (2020, 11 Kasım). *An overview of deep-learning based object-detection algorithms*. Medium [Online]. Erişim: <https://medium.com/@fractaldle/brief-overview-on-object-detection-algorithms-ec516929be93>.
- [254] J. Cartucho, R. Ventura ve M. Veloso, "Robust Object Recognition Through Symbiotic Deep Learning In Mobile Robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Madrid, İspanya, 2018, ss. 2336-2341.

[255] M. N. A. Islam ve S. K. Khan. (2020, 11 Kasım). *Hishabnet: Detection, localization and calculation of handwritten bengali mathematical expressions*. arXiv.org [Online]. Eriřim: <https://arxiv.org/abs/1909.00823>, 2019.



9. EKLER

9.1. EK 1: *donustur.py* KODLAMA DOSYASI

```
#Script to convert yolo annotations to voc format
#Yolo açılama formatından VOC açıklama formatına dönüştürme scripti
import os
import xml.etree.cElementTree as ET
from PIL import Image
ANNOTATIONS_DIR_PREFIX = "D:/gunay tez/data asil/data 20bin/train"
DESTINATION_DIR = "D:/gunay_tez/data_asil/data_20bin/train"
CLASS_MAPPING = {
    '0': 'name'
    # Add your remaining classes here.
}
def create_root(file_prefix, width, height):
    root = ET.Element("annotation")
    ET.SubElement(root, "folder").text = "converted_labels"
    ET.SubElement(root, "filename").text = "{}.jpg".format(file_prefix)
    ET.SubElement(root, "path").text = "{}.jpg".format(file_prefix)
    source = ET.SubElement(root, "source")
    ET.SubElement(source, "database").text = "Unknown"
    size = ET.SubElement(root, "size")
    ET.SubElement(size, "width").text = str(width)
    ET.SubElement(size, "height").text = str(height)
    ET.SubElement(size, "depth").text = "3"
    ET.SubElement(root, "segmented").text = "0"
    return root
def create_object_annotation(root, voc_labels):
    for voc_label in voc_labels:
        obj = ET.SubElement(root, "object")
        ET.SubElement(obj, "name").text=str(voc_label[0])
        ET.SubElement(obj, "pose").text = "Unspecified"
        ET.SubElement(obj, "truncated").text = str(0)
        ET.SubElement(obj, "difficult").text = str(0)
        bbox = ET.SubElement(obj, "bndbox")
        ET.SubElement(bbox, "xmin").text = str(voc_label[1])
        ET.SubElement(bbox, "ymin").text = str(voc_label[2])
        ET.SubElement(bbox, "xmax").text = str(voc_label[3])
        ET.SubElement(bbox, "ymax").text = str(voc_label[4])
    return root
def create_file(file_prefix, width, height, voc_labels):
    root = create_root(file_prefix, width, height)
    root = create_object_annotation(root, voc_labels)
    tree = ET.ElementTree(root)
    tree.write("{}\{}.xml".format(DESTINATION_DIR, file_prefix))
def read_file(file_path):
    file_prefix = file_path.split(".txt")[0]
    image_file_name = "{}.jpg".format(file_prefix)
    img = Image.open("{}\{}".format("D:/gunay_tez/data_asil/data_20bin/train",
    image_file_name))
    w, h = img.size
    with open(os.path.join(ANNOTATIONS_DIR_PREFIX,file_path), 'r') as file:
        lines = file.readlines()
        voc_labels = []
        for line in lines:
            voc = []
            line = line.strip()
            data = line.split()
            CLASS_MAPPING.get(data[0])
            a = int(data[0])
            if a == 0:
                voc.append("al")
            else:
                voc.append("sat")
            bbox_width = float(data[3]) * w
            bbox_height = float(data[4]) * h
```

```

        center_x = float(data[1]) * w
        center_y = float(data[2]) * h
        voc.append(round(center_x - (bbox_width / 2)))
        voc.append(round(center_y - (bbox_height / 2)))
        voc.append(round(center_x + (bbox_width / 2)))
        voc.append(round(center_y + (bbox_height / 2)))
        voc_labels.append(voc)
        create_file(file_prefix, w, h, voc_labels)
    print("Processing complete for file: {}".format(file_path))
def start():
    if not os.path.exists(DESTINATION_DIR):
        os.makedirs(DESTINATION_DIR)
    for filename in os.listdir(ANNOTATIONS_DIR_PREFIX):
        if filename.endswith('.txt'):
            read_file(filename)
        else:
            print("Skipping file: {}".format(filename))
if __name__ == "__main__":
    start()

```



9.2. EK 2: *paylas.py* KODLAMA DOSYASI

```
import glob, os

# Current directory
current_dir = os.path.dirname(os.path.abspath(__file__))
print(current_dir)

# Directory where the data will reside, relative to 'darknet.exe'
path_data = 'C:/Users/daymoon/PycharmProjects/darknet-master/build/darknet/x64/data/data_article/'

# Percentage of images to be used for the test set
percentage_test = 20;

# Create and/or truncate train.txt and test.txt
file_train = open('train.txt', 'w')
file_test = open('test.txt', 'w')

# Populate train.txt and test.txt
counter = 1
index_test = round(100 / percentage_test)
for pathAndFilename in glob.iglob(os.path.join(current_dir, "*.jpg")):
    title, ext = os.path.splitext(os.path.basename(pathAndFilename))
    #file = open(title + '.txt', 'w')
    #file.write('0 0.5 0.5 1 1')
    #file.close()

    if counter == index_test:
        counter = 1
        file_test.write(path_data + title + '.jpg' + "\n")
    else:
        file_train.write(path_data + title + '.jpg' + "\n")
        counter = counter + 1
```

9.3. EK 3: PERFORMANS VERİ SETİ TEST TABLOSU

YoloV3 VG-1							YoloV3 VG-2					
Sıra	Hisse	Al		Sat		Sonuç	Hisse	Al		Sat		Sonuç
		Sinyal İçin		Sonuç Sinyali İçin		% Kar		Sinyal İçin		Sonuç Sinyali İçin		% Kar
		Öneri Tarihi	Öneri Fiyatı	Öneri Tarihi	Öneri Fiyatı			Öneri Tarihi	Öneri Fiyatı	Öneri Tarihi	Öneri Fiyatı	
1	AKSGY	31.8.18	1.69	7.9.18	1.75	3.55	AKSGY	31.8.18	1.69	11.09.18	1.71	1.18
2	AKSUE	31.8.18	6.50	25.9.18	8.92	37.23	AKSUE	31.8.18	6.50	26.09.18	8.63	32.77
3	DOGUB	31.8.18	2.43	13.9.18	3.09	27.16	ARSAN	31.8.18	1.90	6.09.18	1.86	-2.11
4	INDES	31.8.18	5.48	11.9.18	5.79	5.66	INDES	31.8.18	5.48	12.09.18	5.61	2.37
5	KERVT	31.8.18	1.92	10.9.18	1.93	0.52	KERVT	31.8.18	1.92	11.09.18	1.86	-3.12
6	KLMNA	31.8.18	3.26	6.9.18	3.21	-1.53	LİNK	31.8.18	8.81	7.09.18	9.25	4.99
7	SEYKM	31.8.18	3.90	28.9.18	3.99	2.31	SEYKM	31.8.18	3.90	28.09.18	3.99	2.31
8							TURGG	31.8.18	30.80	4.09.18	30.26	-1.75
	SEPET 1	SEPET ORTALAMA KAR				10.70	SEPET 1	SEPET ORTALAMA KAR				4.58
11	ADESE	1.10.18	2.85	15.10.18	4.31	51.23	ADESE	1.10.18	2.85	23.10.18	6.10	114.04
12	BRYAT	1.10.18	37.82	12.10.18	36.30	-4.02	ANHYT	1.10.18	5.51	4.10.18	5.46	-0.91
13	DIRIT	1.10.18	0.79	4.10.18	0.78	-1.27	ARMDA	1.10.18	9.31	19.10.18	10.50	12.78
14	ECZYT	1.10.18	7.32	3.10.18	7.14	-2.46	DURDO	1.10.18	3.25	9.10.18	3.81	17.23
15	ERBOS	1.10.18	15.80	4.10.18	15.97	1.08	KAREL	1.10.18	5.46	3.10.18	5.19	-4.95
16	SAHOL	1.10.18	7.33	19.10.18	7.11	-3.00	SAHOL	1.10.18	7.33	2.10.18	6.94	-5.32
17	VKING	1.10.18	1.34	22.10.18	1.24	-7.46	VKING	1.10.18	1.34	4.10.18	1.27	-5.22
18							ULAS	1.10.18	0.49	3.10.18	0.47	-4.08
	SEPET 2	SEPET ORTALAMA KAR				4.87	SEPET 2	SEPET ORTALAMA KAR				15.45
24	AKFGY	24.10.18	1.93	9.11.18	1.35	-30.05						
25	AKGRT	24.10.18	3.01	30.10.18	2.95	-1.99						
26	AYGAZ	24.10.18	11.33	25.10.18	10.83	-4.41						
27	CMENT	24.10.18	7.75	26.10.18	7.81	0.77	CMENT	24.10.18	7.75	30.10.18	7.73	-0.26
28							AKYHO	24.10.18	2.73	31.10.18	2.50	-8.42
29							DGATE	24.10.18	4.79	31.10.18	4.53	-5.43
30							ICBCT	24.10.18	6.14	9.11.18	6.99	13.84
31							KORDS	24.10.18	9.84	9.11.18	9.68	-1.63
	SEPET 3	SEPET ORTALAMA KAR				-8.92	SEPET 3	SEPET ORTALAMA KAR				-0.38

Sıra	Hisse	Al		Sat		Sonuç
		Sinyal İçin		Sonuç Sinyali İçin		% Kar
		Öneri Tarihi	Öneri Fiyatı	Öneri Tarihi	Öneri Fiyatı	
54	AKENR	13.12.18	0.62	2.1.19	0.61	-1.61
55	AKFGY	13.12.18	1.48	21.12.18	1.57	6.08
56	ANHYT	13.12.18	5.63	20.12.18	5.49	-2.49
57	BLCYT	13.12.18	1.29	27.12.18	1.26	-2.33
58	ENJSA	13.12.18	4.78	19.12.18	4.79	0.21
59	ERBOS	13.12.18	15.06	25.12.18	14.97	-0.60
60	ERSU	13.12.18	0.94	25.12.18	0.90	-4.26
61	HEKTS	13.12.18	3.44	14.12.18	3.39	-1.45
62	ISGYO	13.12.18	1.04	10.1.19	0.99	-4.81
63	MAVI	13.12.18	34.10	18.12.18	31.92	-6.39
64	MPARK	13.12.18	11.80	28.12.18	10.57	-10.42
65	PKART	13.12.18	2.31	3.1.19	2.86	23.81
66						
67						
68						
	SEPET 6	SEPET ORTALAMA KAR				-0.35

Hisse	Al		Sat		Sonuç
	Sinyal İçin		Sonuç Sinyali İçin		% Kar
	Öneri Tarihi	Öneri Fiyatı	Öneri Tarihi	Öneri Fiyatı	
AKENR	13.12.18	0.62	31.12.18	0.63	1.61
ERBOS	13.12.18	15.41	26.12.18	14.97	-2.86
ERSU	13.12.18	0.94	17.12.18	0.91	-3.19
HEKTS	13.12.18	3.44	17.12.18	3.35	-2.62
ISGYO	13.12.18	1.04	2.1.19	1.04	0.00
MAVI	13.12.18	34.10	25.12.18	31.88	-6.51
AEFES	13.12.18	18.75	28.12.18	18.39	-1.92
ISFIN	13.12.18	4.99	25.12.18	4.45	-10.82
SAMAT	13.12.18	0.68	24.12.18	0.72	5.88
SEPET 6	SEPET ORTALAMA KAR				-2.27

Sıra	Hisse	Al		Sat		Sonuç
		Sinyal İçin		Sonuç Sinyali İçin		% Kar
		Öneri Tarihi	Öneri Fiyatı	Öneri Tarihi	Öneri Fiyatı	
69	AKSGY	4.1.19	1.76	14.01.19	1.79	1.70
70	CMBTN	4.1.19	28.50	14.01.19	28.12	-1.33
71	HEKTS	4.1.19	8.89	8.02.19	10.94	23.06
72	ISYAT	4.1.19	0.74	11.01.19	0.77	4.05
73	MARKA	4.1.19	0.66	8.01.19	0.63	-4.55
74	MEGAP	4.1.19	2.04	9.01.19	2.03	-0.49
75	SAFKR	4.1.19	1.53	24.01.19	1.77	15.69
76	SAHOL	4.1.19	6.95	8.02.19	8.81	26.76
77	SELEC	4.1.19	2.70	15.01.19	3.16	17.04
78	TKFEN	4.1.19	18.87	4.02.19	22.28	18.07
79						
80						
81						
82						
83						
84						
85						
	SEPET 7	SEPET ORTALAMA KAR				10.00

Hisse	Al		Sat		Sonuç	
	Sinyal İçin		Sonuç Sinyali İçin		% Kar	
	Öneri Tarihi	Öneri Fiyatı	Öneri Tarihi	Öneri Fiyatı		
AKSGY	4.1.19	1.76	11.1.19	1.83	3.98	
HEKTS	4.1.19	3.24	8.1.19	3.24	0.00	
MARKA	4.1.19	0.66	8.1.19	0.63	-4.55	
MEGAP	4.1.19	1.17	10.1.19	1.15	-1.71	
TKFEN	4.1.19	17.67	11.1.19	18.41	4.19	
AVOD	4.1.19	2.05	16.1.19	2.03	-0.98	
BIMAS	4.1.19	41.30	15.1.19	42.13	2.01	
GARAN	4.1.19	8.00	10.1.19	7.87	-1.63	
HLBNK	4.1.19	6.81	10.1.19	6.78	-0.44	
HURGZ	4.1.19	0.90	1.2.19	0.95	5.56	
ISCTR	4.1.19	4.22	8.1.19	4.21	-0.24	
ULUUN	4.1.19	2.04	18.1.19	2.19	7.35	
	SEPET 7	SEPET ORTALAMA KAR				1.13

Sıra	Hisse	Al		Sat		Sonuç
		Sinyal İçin		Sonuç Sinyali İçin		% Kar
		Öneri Tarihi	Öneri Fiyatı	Öneri Tarihi	Öneri Fiyatı	
86	AFYON	11.2.19	4.81	13.2.19	4.60	-4.37
87	AKBNK	11.2.19	6.94	28.2.19	6.87	-1.01
88	AKCNS	11.2.19	7.03	13.2.19	6.94	-1.28
89	AKENR	11.2.19	0.65	15.2.19	0.64	-1.54
90	ASELS	11.2.19	24.32	25.1.19	24.96	2.63
91	ATLAS	11.2.19	1.02	20.2.19	1.08	5.88
92	BFREN	11.2.19	158.95	14.2.19	158.39	-0.35
93	BRSAN	11.2.19	7.44	13.2.19	7.32	-1.61
94	CLEBI	11.2.19	55.16	13.2.19	53.95	-2.19
95	CRDFA	11.2.19	1.01	12.2.19	0.99	-1.98
96	DOAS	11.2.19	4.82	27.2.19	5.03	4.36
97	KLMSN	11.2.19	5.02	15.3.19	6.61	31.67
98	KRDMB	11.2.19	1.95	1.3.19	2.1	7.69
99	NETAS	11.2.19	7.41	13.2.19	7.16	-3.37
100	PINSU	11.2.19	1.32	14.2.19	1.3	-1.52
101	TRGYO	11.2.19	1.76	6.3.19	1.92	9.09
102	ULUSE	11.2.19	16.06	13.5.19	21.3	32.63
103						
104						
106						
107						
108						
109						
110						
111						
112						
	SEPET 8	SEPET ORTALAMA KAR				4.40

Hisse	Al		Sat		Sonuç
	Sinyal İçin		Sonuç Sinyali İçin		% Kar
	Öneri Tarihi	Öneri Fiyatı	Öneri Tarihi	Öneri Fiyatı	
AFYON	11.2.19	1.74	14.2.19	1.71	-1.72
ASELS	11.2.19	12.09	25.2.19	12.41	2.65
ATLAS	11.2.19	1.02	21.2.19	1.07	4.90
BRFREN	11.2.19	158.95	13.2.19	158.39	-0.35
BRSAN	11.2.19	7.44	13.2.19	7.32	-1.61
CLEBI	11.2.19	55.16	14.2.19	52.16	-5.44
KLMSN	11.2.19	4.82	28.2.19	5.79	20.12
KRDMB	11.2.19	1.85	27.2.19	2.21	19.46
NETAS	11.2.19	7.41	26.2.19	8.34	12.55
TRGYO	11.2.19	1.76	1.3.19	1.93	9.66
ULUSE	11.2.19	16.06	13.2.19	15.75	-1.93
ALARK	11.2.19	2.40	25.2.19	2.60	8.33
ATEKS	11.2.19	8.89	13.3.19	8.91	0.22
DYOBY	11.2.19	2.93	14.2.19	2.84	-3.07
ECZYT	11.2.19	6.88	27.2.19	6.90	0.29
EGGUB	11.2.19	23.95	18.2.19	23.53	-1.75
KATMR	11.2.19	3.07	28.2.19	3.16	2.93
KRDMA	11.2.19	1.85	13.2.20	1.8	-2.70
SELEC	11.2.19	3.19	1.3.19	3.6	12.85
ULUUN	11.2.19	2.33	12.3.19	2.61	12.02
SEPET 8	SEPET ORTALAMA KAR				4.37

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Günay TEMÜR
Doğum Tarihi ve Yeri : 1983 – Cide
Yabancı Dili : İngilizce

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Doktora	Elektrik, Elektronik ve Bilgisayar Müh.	Düzce Üniversitesi	2021
Lisans	Bilgisayar Mühendisliği	Düzce Üniversitesi	2020
Y. Lisans	Elektrik Eğitimi	Düzce Üniversitesi	2013
Lisans	Bilgisayar Sistemleri Öğretmenliği	Süleyman Demirel Üniversitesi	2006
Lise	Bilgisayar Yazılımı	Sakarya Teknik Lisesi	2001

SSCI, SCI-Exp, AHCI Kapsamındaki Dergilerde Yayımlanan Tam Makale:

- [1] S. Birogul, G. Temür ve U. Köse, “YOLO Object Recognition Algorithm and ‘Buy-Sell Decision’ Model over 2D Candlestick Charts,” *IEEE Access*, c. 8, ss. 91894–91915, 2020.
- [2] A. Soy Temür, M. Akgün ve G. Temür, “Predicting Housing Sales In Turkey Using Arima, Lstm And Hybrid Models,” *Journal Bussiness Economics and Managment*, c. 20, sayı 5, ss. 920–938, 2019.

Diğer Hakemli Ulusal veya Uluslararası Dergilerde Yayımlanan Tam Makale:

- [3] A. Soy Temür, İ.İ. İşler ve G. Temür, “Finansal Performansın Topsis Yöntemi ile Değerlendirilmesi: Bist Perakende Ticaret İşletmeleri Üzerine Bir Uygulama,” *Kesit Akademi*, c. 3, sayı 11, ss. 712-729, 2017.
- [4] İ. Şahin ve G. Temür, “Yapay Sinir Ağlarının Otomatik Olarak FPGA Çipine Uygulanması İçin Otomatik Denetleyici Tasarım Aracı,” *SDU International Journal of Technological Science*, c. 8, sayı 1, ss. 34-52, 2016.

- [5] İ. Şahin ve G. Temür, “Yüzde Tabanlı String Eşleme Problemi İçin Yeni Bir Donanım Modülü Tasarımı,” *Sakarya University Journal of Science*, c. 20, sayı 3, ss. 441–450, 2016.

Ulusal, Uluslararası Kongre ve Sempozyumlarda Sözlü Olarak Sunulan ve Tam Metin Olarak Yayımlanan Bildiriler:

- [6] İ. Düzdar ve G. Temür, "Comparing of results and implementation of clustering methods of data mining software with a data set," *Electric Electronics, Computer Science, Biomedical Engineerings' Meeting*, Istanbul, 2017, ss. 1-5.
- [7] G. Temür ve A. Soy Temür, “Ulaştırma Şirketlerinin Hisse Senedi Fiyatlarının Ulaştırma Endeksine Etkisi: Latek-Reysaş Lojistik Şirketi Analizi,” , *II. Ulusal Lojistik ve Tedarik Zinciri Kongresi*, Aksaray, 2013.

Sözlü Olarak Sunulan ve Yayımlanmayan Bildiriler:

- [8] İ. Şahin ve G. Temür, “A Hardware Module Design For Levenshtein Distance Algorithm,” 5TH International Vocational Schools Symposium’ unda sunuldu, Prizen, 2016