

**ALÜMİNYUM DÖKÜM HATALARININ DERİN  
ÖĞRENME YAKLAŞIMIYLA TESPİTİ VE  
SINIFLANDIRILMASI**

**İsmail Enes PARLAK**



T.C.  
BURSA ULUDAĞ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**ALÜMİNYUM DÖKÜM HATALARININ DERİN ÖĞRENME  
YAKLAŞIMIYLA TESPİTİ VE SINIFLANDIRILMASI**

İsmail Enes PARLAK  
511806001

Prof. Dr. Erdal EMEL  
(Danışman)

DOKTORA TEZİ  
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA-2022  
**Her hakkı saklıdır.**

**B.U.Ü. Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;**

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

**beyan ederim.**

../../....

**TEZ YAYINLANMA  
FİKRİ MÜLKİYET HAKLARI BEYANI**

Enstitü tarafından onaylanan lisansüstü tezin/raporun tamamını veya herhangi bir kısmını, basılı (kâğıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma izni Bursa Uludağ Üniversitesi'ne aittir. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet hakları ile tezin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları tarafımıza ait olacaktır. Tezde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alınarak kullandığını ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederiz.

Yükseköğretim Kurulu tarafından yayınlanan “**Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge**” kapsamında, yönerge tarafından belirtilen kısıtlamalar olmadığı takdirde tezin YÖK Ulusal Tez Merkezi / B.U.Ü. Kütüphanesi Açık Erişim Sistemi ve üye olunan diğer veri tabanlarının (Proquest veri tabanı gibi) erişimine açılması uygundur.

Danışman Adı-Soyadı  
Tarih

Öğrencinin Adı-Soyadı  
Tarih

İmza

İmza

Bu bölüme kişinin kendi el yazısı ile okudum  
anladım yazmalı ve imzalanmalıdır.

Bu bölüme kişinin kendi el yazısı ile okudum  
anladım yazmalı ve imzalanmalıdır

## ÖZET

Doktora Tezi

### ALÜMİNYUM DÖKÜM HATALARININ DERİN ÖĞRENME YAKLAŞIMIYLA TESPİTİ VE SINIFLANDIRILMASI

**İsmail Enes PARLAK**

Bursa Uludağ Üniversitesi  
Fen Bilimleri Enstitüsü  
Endüstri Mühendisliği Anabilim Dalı

**Danışman:** Prof. Dr. Erdal EMEL

Eşsiz özellikleri nedeniyle, yüksek basınçlı döküm tekniğiyle üretilen alüminyum döküm parçaları, özellikle otomotiv endüstrisinde yaygın olarak kullanılmaktadır. Bununla birlikte döküm parçalarının muayenesi, başka türlü görülemeyen iç kusurları incelemek için X-ışını gibi teknolojiler kullanılarak kritik bileşenlerin tahribatsız test edilmesini gerektiren bir süreçtir. Böylesine zaman alıcı bir görsel muayene, azami dikkatle iyi eğitilmiş uzmanlar gerektirir. Bu çalışmada, alüminyum döküm parçalarının iç kusurlarını uluslararası bir standarda uygun olarak tespit etmek, seviyelendirmek ve bu kusur bilgileriyle döküm süreci parametrelerini ilişkilendirmek amaçlanmıştır. Kullanılan veri seti (Al-Cast veri seti), X-ışını görüntülerinden oluşan özgün bir veri seti olup tüm görüntüler dikkatlice etiketlenmiştir ve tahribatsız muayene topluluğu için açık erişimli olarak paylaşılmıştır. Kusurların tespiti için derin öğrenme tabanlı bir nesne tespit yöntemi kullanılmış olup, tespitler gerçek-zamanlı ve yüksek doğrulukta (0.97'lik mAP değeri) gerçekleştirilmektedir. Tespitten elde edilen bilgilerle k-ortalamlar++ kümeleme algoritmasından faydalanılarak kusurlar, ASTM standartlarına göre kusur oranları belirlenerek seviyelendirilir. Ayrıca yedi farklı döküm süreci parametresi kullanılarak kusur oranı çıktısı üzerinden nispeten az sayıda yapılan deneylerle bu parametrelerin kusur oranına etkileri analiz edilmiştir. Sonuçlar özellikle ikinci faz piston hızı, üçüncü faz basıncı ve enjeksiyon dolu noktasının kusur oranı üzerinde anlamlı bir etkiye sahip olduğunu göstermektedir. Çalışma bütünsel olarak hem kalite kontrol hem de üretim aşamasında bir karar destek sistemi olarak kullanılmak üzere önemli bir potansiyele sahiptir.

**Anahtar Kelimeler:** Alüminyum, derin öğrenme, döküm süreci, kusur tespiti ve seviyelendirme, tahribatsız muayene, X-ışını, yüksek basınçlı döküm  
**2022, ix +118 sayfa.**

## ABSTRACT

PhD Thesis

### DETECTION AND CLASSIFICATION OF ALUMINUM CASTING DEFECTS BY DEEP LEARNING APPROACH

**İsmail Enes PARLAK**

Bursa Uludag University  
Graduate School of Natural and Applied Sciences  
Department of Industrial Engineering

**Supervisor:** Prof. Dr. Erdal EMEL

Due to their unique properties, aluminum casting parts produced by high pressure casting technique are widely used, especially in the automotive industry. However, inspection of castings is a process that requires non-destructive testing of critical components using technologies such as X-ray to examine internal defects that cannot be seen otherwise. Such a time-consuming visual inspection requires well-trained experts with the utmost care. In this study, it is aimed to detect and grade the internal defects of aluminum casting parts in accordance with an international standard and to correlate this defect information with the casting process parameters. The dataset used (Al-Cast dataset) is a unique dataset of X-ray images, all images are carefully labeled and shared open access for the nondestructive testing community. A deep learning-based object detection method is used for the detection of defects, and the detections are performed in real-time and with high accuracy (mAP value of 0.97). Using the k-means++ clustering algorithm with the information obtained from the detection, the defects are graded by determining the defect rates according to ASTM standards. In addition, the effects of these parameters on the defect rate were analyzed with a relatively small number of experiments on the defect rate output using seven different casting process parameters. The results show that especially the second phase piston speed, third phase pressure and injection fill level have a significant effect on the defect rate. The study has significant potential to be used as a whole decision support system both quality control and production phase.

**Key words:** Aluminum, deep learning, defect detection and grading, die-casting process, high pressure die-casting, nondestructive testing, X-ray  
**2022, ix +118 pages.**

## ÖNSÖZ ve TEŞEKKÜR

Bu tezin hazırlanması sürecinde yol gösteren, değerli bilgi ve tecrübeleri ile desteğini esirgemeyen tez danışmanım Prof. Dr. Erdal EMEL'e en içten teşekkürlerimi sunarım.

Tez komitemde yer alan değerli hocalarım Doç. Dr. Aytaç YILDIZ ve Doç. Dr. Gıyasettin ÖZCAN'a bu süreçteki destekleri için teşekkürlerimi sunarım.

Tez çalışmasında yer alan verilerin elde edilmesindeki ve kullanılmasındaki destekleri için Alcast Metal Genel Müdürü Sayın Ali Osman KÜÇÜK ve Alcast Metal çalışanlarına teşekkürü borç bilirim.

Desteği ve anlayışı ile yoğun ve yorucu tez çalışmalarım süresince verdiği destek, gösterdiği sabır ve fedakârlık ile her zaman yanımda olan sevgili eşim Ayşenur EFE PARLAK'a ve bu çalışma süresince beni destekleyen anneme ve babama sonsuz teşekkürlerimi sunarım.

İsmail Enes PARLAK

27/11/2022

## İÇİNDEKİLER

	<b>Sayfa</b>
ÖZET.....	i
ABSTRACT.....	ii
ÖNSÖZ ve TEŞEKKÜR.....	iii
SİMGELER ve KISALTMALAR DİZİNİ .....	v
ŞEKİLLER DİZİNİ.....	vii
ÇİZELGELER DİZİNİ .....	ix
1. GİRİŞ .....	1
2. KAYNAK ÖZETLERİ .....	8
2.1. Derin Nesne Tespit Yöntemleri .....	8
2.2. Döküm Kusurlarının Tespiti .....	9
2.3. Basınçlı Döküm Süreci Parametrelerinin Optimizasyonu .....	16
3. MATERYAL ve YÖNTEM.....	19
3.1. Materyal .....	19
3.2.2. Modern nesne tespit yöntemleri ve YOLO .....	36
3.2.3. Kusur görüntülerinin bölütlenmesi ve eşiklenmesi.....	57
3.2.4. Kusur seviyelerinin belirlenmesi.....	67
3.2.5. Performans metrikleri.....	69
3.2.6. Döküm parametrelerinin optimizasyonu.....	71
4. BULGULAR ve TARTIŞMA.....	76
4.1. Bulgular.....	76
4.1.1. Ön çalışmalar (nesne tespiti için en uygun dedektörün belirlenmesi) .....	76
4.1.2. YOLOv5 dedektörleriyle kusur tespiti.....	78
4.1.3. Kusurların bölütlenmesi ve eşiklenmesi .....	82
4.1.4. Kusurların seviyelerinin belirlenmesi .....	91
4.1.5. Döküm süreci parametreleri .....	94
4.2. Tartışma.....	98
5. SONUÇ .....	101
KAYNAKLAR .....	103
ÖZGEÇMİŞ .....	118



## SİMGELER ve KISALTMALAR DİZİNİ

### Simgeler

$b_x, b_y, b_w, b_h$   
 $Pr(x)$   
 $\phi(x)$   
 $\lambda_{koordinat}, \lambda_{nesneyok}$   
 $G(x)$   
 $\sigma$   
 $\sigma^2$   
 $SC(i)$

### Açıklama

Sınırlayıcı kutu koordinat ve boyut bilgileri  
x'in gerçekleşme olasılığı  
Aktivasyon fonksiyonu  
Kayıp fonksiyonunda yer alan sabitler  
Gauss fonksiyonu  
Bulanıklık parametresi, standart sapma  
Varyans  
Silhouette katsayısı

### Kısaltmalar

YBD  
TM  
DR  
GPU  
CAGR  
CNN  
R-CNN  
R-FCN  
YOLO  
SSD  
SVM  
HOG  
BT  
LBP  
GAP  
ASTM  
YSA  
GA  
TB

### Açıklama

Yüksek Basıncılı Döküm  
Tahribatsız Muayene  
Dijital Radyografi  
Grafik İşleme Birimi  
(Graphics Processing Unit)  
Yıllık Bileşik Büyüme Oranı  
(Compound Annual Growth Rate)  
Konvolüsyonel Sinir Ağı  
(Convolutional Neural Network)  
Bölge Tabanlı CNN  
(Region Based CNN)  
Bölge Tabanlı Tam Evrişimli Ağ  
(Region-based Fully Convolutional Network)  
Yalnızca Bir Kere Bakarsın  
(You Only Look Once)  
Tek Atışlı Dedektör  
(Single Shot Detector)  
Destek Vektör Makineleri  
(Support Vector Machines)  
Yönlü Gradyanlar Histogramı  
(Histogram of Oriented Gradients)  
Bilgisayarlı Tomografi  
Yerel İkili Desenler  
(Local Binary Patterns)  
Gri Tonlamalı Sıralama Çiftleri  
(Grayscale Arranging Pairs)  
Amerikan Test ve Malzeme Derneği  
(American Society for Testing and Materials)  
Yapay Sinir Ağı  
Genetik Algoritma  
Tavlama Benzetimi

KDS	Karar Destek Sistemi
ReLU	Düzeltilmiş Doğrusal Birim (Rectified Linear Unit)
CPU	Merkezi İşlem Birimi (Central Processing Unit)
FPN	Öznitelik Piramit Ağı (Feature Pyramid Network)
PAN	Yol Toplama Ağı (Path Aggregation Network)
IoU	Birleşim Üzerinden Kesişim
RPN	Bölge Öneri Ağı
AHE	Uyarlamalı Histogram Eşitleme
CLAHE	Kontrast Sınırlı Uyarlanabilir Histogram Eşitleme
DBI	Davies-Bouldin İndeksi
BPO	Beyaz Piksel Oranı

## ŞEKİLLER DİZİNİ

	<b>Sayfa</b>
Şekil 1.1. 2021-2030 yılları arasında alüminyum pazar büyüklüğü tahmini (milyar ABD doları) (Precedence Research, 2021) .....	2
Şekil 1.2. Endüstriyel X-ışını kontrol sisteminin şematik gösterimi (Hanke ve diğerleri, 2008) .....	4
Şekil 2.1. Derin öğrenme tabanlı nesne dedektörleri (Parlak ve Emel, 2023) .....	10
Şekil 3.1. Veri setinde yer alan bazı görüntü örnekleri.....	20
Şekil 3.2. ASTM referanslarında yer alan bazı kusur görüntüleri (ASTM, 2001) .....	21
Şekil 3.3. Veri setinde yer alan görüntülerden kırılmış bazı kusur örnekleri.....	22
Şekil 3.4. Elipsoit kusur simülasyonu ile oluşturulan yapay kusur örnekleri.....	23
Şekil 3.5. Kusur tiplerinin veri setindeki toplam sayıları.....	24
Şekil 3.6. Veri artırım teknikleri ile oluşturulan görüntüler.....	24
Şekil 3.7. Etiketleme örnekleri.....	26
Şekil 3.8. Önerilen yöntemin şeması.....	27
Şekil 3.9. Farklı bilgisayarlı görme görevleri; (a) nesne tespiti, (b) örnek bölütleme ve (c) semantik bölütleme.....	30
Şekil 3.10. Basit bir CNN yapısının görünümü.....	31
Şekil 3.11. Giriş görüntüsü üzerinde filtrenin bir adım ilerlemesi.....	32
Şekil 3.12. Maksimum ortaklama örneği.....	33
Şekil 3.13. Genel bir nesne tespit algoritmasının gösterimi.....	35
Şekil 3.14. Nesne dedektörlerinin mimarisi (Bochkovskiy ve diğerleri, 2020b) .....	36
Şekil 3.15. Birleşim üzerinden kesişim ( <i>IoU</i> ).....	37
Şekil 3.16. R-CNN mimarisi (Girshick ve diğerleri, 2014) .....	38
Şekil 3.17. Daha Hızlı R-CNN mimarisi (Ren ve diğerleri, 2017) .....	39
Şekil 3.18. SSD mimarisi (Liu ve diğerleri, 2016b) .....	40
Şekil 3.19. VGG16 ağının katmanları (Simonyan ve Zisserman, 2015) .....	40
Şekil 3.20. EfficientDet mimarisi (Tan ve diğerleri, 2020b) .....	41
Şekil 3.21. YOLO girdisinin ızgara hücrelerine bölünmesi.....	43
Şekil 3.22. YOLO ağ mimarisinin basit gösterimi (Redmon ve diğerleri, 2016) .....	44
Şekil 3.23. YOLOv2’de kullanılan bağlantı kutularının temsili gösterimi.....	48
Şekil 3.24. ResNet atlama bağlantıları mimarisi (He ve diğerleri, 2016b) .....	50
Şekil 3.25. Darknet-53 mimarisi (Redmon ve Farhadi, 2018b) .....	51
Şekil 3.26. YOLOv3 mimarisindeki üç farklı ölçekte yapılan tahmin (Kathuria, 2018) .....	52
Şekil 3.27. FPN mimarisi.....	54
Şekil 3.28. YOLOv5 mimarisi (Xu ve diğerleri, 2021) .....	57
Şekil 3.29. Bir kusur görüntüsüne ait histogram.....	60
Şekil 3.30. ASTM kusur seviyelerine ait seviye aralıkları (ASTM International, 2014) .....	67
Şekil 3.31. Kusur seviyelendirme süreci.....	68
Şekil 3.32. <i>JI</i> değerinin hesaplanmasını gösteren bir örnek.....	72
Şekil 3.33. Enjeksiyonlu döküm bileşenleri.....	73

Şekil 3.34.	Döküm süreci parametrelerinin Ishikawa diyagramı (Syracos, 2003)	73
Şekil 4.1.	YOLOv5 modellerinin 100 epoch boyunca değişen mAP değerleri.....	80
Şekil 4.2.	Denklem (3.27)'ye göre karışıklık matrisleri.....	81
Şekil 4.3.	YOLOv5n ile elde edilen çeşitli test çıktıları.....	83
Şekil 4.4.	YOLOv5l ile elde edilen çeşitli test çıktıları.....	84
Şekil 4.5.	Kusurun orijinal görüntüden kırılması.....	85
Şekil 4.6.	Gürültü giderme işlemi uygulanmış görüntü örnekleri.....	86
Şekil 4.7.	Histogramları eşitlenmiş kusur görüntüsü örnekleri.....	86
Şekil 4.8.	Bazı iyi bilinen otomatik eşikleme yöntemlerinin çıktıları.....	88
Şekil 4.9.	k-ortalamalar++ ile farklı k değerlerinde bölütleme örnekleri.....	88
Şekil 4.10.	Önerilen yöntem grubuyla elde edilen bazı siyah-beyaz görüntü örnekleri.....	92
Şekil 4.11.	Kusur oranının hesaplanması.....	93
Şekil 4.12.	Parametre analizi için deneylerde kullanılan parçanın X-ışını görüntüsü.....	94
Şekil 4.13.	Kusur oranı ve faktörlerin değişimini gösteren grafikler.....	97
Şekil 4.14.	İstenirlik ve faktörler grafiği.....	98

## ÇİZELGELER DİZİNİ

		<b>Sayfa</b>
Çizelge 2.1.	Döküm kusurlarının derin öğrenme yöntemleriyle tespiti ve bölütlenmesi.....	14
Çizelge 3.1.	ANOVA tablosu.....	74
Çizelge 4.1.	Nesne dedektörlerinin performansları.....	78
Çizelge 4.2.	YOLOv5 yöntemlerinin kusur tespitindeki eğitim performansları.....	79
Çizelge 4.3.	YOLOv5n ve YOLOv5l dedektörlerinin test sonuçları.....	82
Çizelge 4.4.	k-ortalamalar++ kümeleme algoritmasının performansı (k sayısı) .....	89
Çizelge 4.5.	Üç farklı yöntemin küme sayısı k'nin seçimindeki performansı.....	91
Çizelge 4.6.	Parça tiplerine ait piksel sayıları.....	93
Çizelge 4.7.	Döküm süreci parametreleri ve değer aralıkları.....	94
Çizelge 4.8.	Deneylere ait veriler.....	95
Çizelge 4.9.	Ana etkilere ait ANOVA tablosu.....	97
Çizelge 4.10.	Regresyon modeli katsayıları.....	97

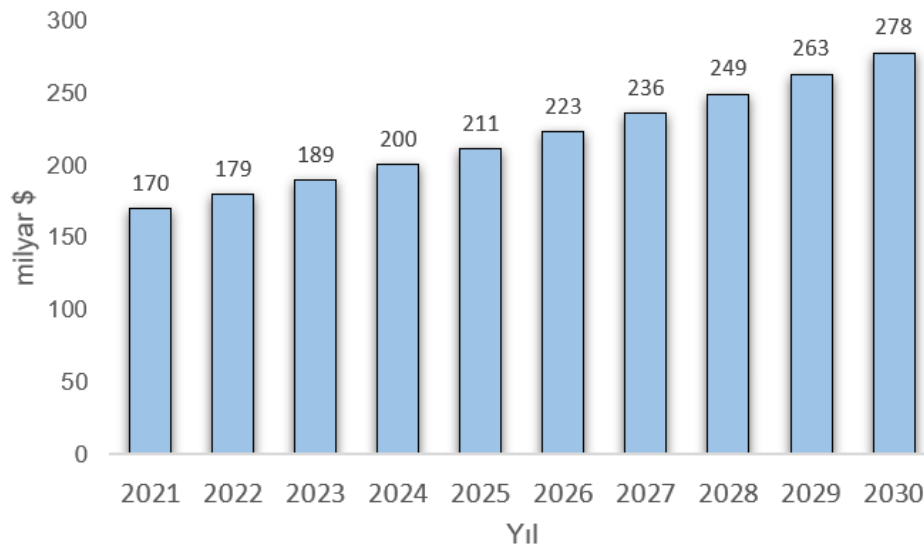
## 1. GİRİŞ

Motorlu araç bileşenlerinin üretiminde kullanılan çeşitli teknikler arasında en yaygın olanı, kalıcı kalıba döküm tekniklerinden biri olan basınçlı dökümdür. Basınçlı döküm, eriyik metalin, yine metalden bir kalıbın kalıp boşluğuna basınç altında enjekte edildiği bir döküm tekniğidir. Basınçlı döküm tekniklerinden biri olarak yüksek basınçlı döküm (YBD) ise, araç bileşenlerinin seri üretimi için en hızlı ve en uygun maliyetli olanıdır. Yüksek üretim hızına karşılık, boyutsal kararlılık ve yüksek parça doğruluğu, YBD tekniğinin döküm süreci için seçilmesinin ana nedenleri arasındadır. YBD tekniğinin diğer yöntemlere kıyasla daha yüksek üretim hızı sağlamasının nedeni ise eriyik metalin çok kısa bir süre içerisinde metal kalıp içinde katılaşıp, döküm parça olarak çıkarılabilmesidir (Bonollo ve diğerleri, 2015).

Alüminyum (Al) ve alaşımları, düşük yoğunluk, iyi elektriksel ve termal iletkenlik, yüksek süneklik ve dövülebilirlik gibi benzersiz özelliklerinden dolayı birçok alanda yaygın olarak kullanılmaktadır. Otomotiv endüstrisinde daha önce çelik ve dökme demir yaygın olarak kullanılmış olsa da son zamanlarda alüminyum gibi hafif alaşımlı dökümler tercih edilmektedir. Alüminyum alaşımlarının taşıt araçları uygulamalarında uzun bir geçmişi olsa da bu malzemelerin yaygın kullanımı sadece 1970'lerin ortalarında başlamıştır (Luo, 2021). Enerji maliyetlerindeki artış ve dünya çapında emisyon azaltma ihtiyacı, alüminyum taşıt araçlarındaki kullanımı için daha çekici hale getirmiştir. Son 30 yılda, beygir gücü ve performansı arttıkça binek otomobillerin ağırlığı iki katına çıkmıştır. Araç ağırlığını kontrol altında tutmak için alüminyum, üreticiler tarafından daha çok tercih edilmektedir (Das ve Yin, 2007). Bir binek otomobilde ortalama alüminyum kullanımı, araca bağlı olarak %12 ile %60 arasında değişmektedir (Salonitis ve diğerleri, 2019). Küresel alüminyum pazar büyüklüğünün 2021'de 169,8 milyar ABD doları olduğu tahmin edilirken, 2022'den 2030'a kadar %5,61'lik bir bileşik yıllık büyüme oranında (CAGR) büyüyerek 2030 yılına kadar yaklaşık olarak 277,5 milyar ABD dolarına ulaşması beklenmektedir (Şekil 1.1) (Precedence Research, 2021).

Otomotiv endüstrisi için üretilen alüminyum dökümlerin önemli bir bölümü, araç güvenliği için kritik öneme sahip olup, çoğu kez parça kalitesi %100 güvence altına alınmak durumundadır. YBD parçalarının görsel denetimi, kalite kontrol sürecinde

önemli bir adımdır. Herhangi bir üretim sürecinde kusurlar, o süreç içinde var olan toleranslara ve üretim parametrelerine bağlıdır. Kusur oluşumuna bağlı olarak özellikle mekanik mukavemet kriterlerinin karşılanamaması, son derece maliyetli bir süreç olan kalıp takımlarının yeniden tasarlanmasına yol açabilir (Zhang ve diğerleri, 2005). Döküm ürünleri, gözeneklilik, gaz boşluğu, büzülme, çekinti, çatlaklar ve yüzey kusurları gibi çeşitli kusurlar içerebilir. Kusurlar parçanın sertlik gibi birçok karakteristiğini olumsuz yönde etkileyebilir. Bu nedenle üretim sürecinde kusurların tespit edilmesi önem arz etmektedir. Yüzey kusurları, parçaya zarar vermeden veya pahalı ekipman kullanılmadan doğrudan insan gözüyle tespit edilebilirken, iç kusurları tespit etmek için görsel test, penetran testi, manyetik partikül incelemesi, radyografik muayene, ultrasonik muayene, girdap akımı testi ve akustik emisyon gibi çeşitli tahribatsız muayene tekniklerine veya teknolojilerine ihtiyaç vardır.



**Şekil 1.1.** 2021-2030 yılları arasında alüminyum pazar büyüklüğü tahmini (milyar ABD Doları) (Precedence Research, 2021)

Tahribatsız Muayene (TM) yöntemleri, malzemeyi kesmeden veya başka bir şekilde değiştirmeden malzemelerin yüzeyindeki ve içindeki kusurların veya hasarların tanımlanmasını ve karakterizasyonunu amaçlar (Lockard, 2015). Başka bir deyişle TM, malzemenin orijinal özelliklerini değiştirmeden veya test edilen nesneye zarar vermeden bazı standartlara kıyasla karakterizasyon veya kusurları bulmak için malzeme veya

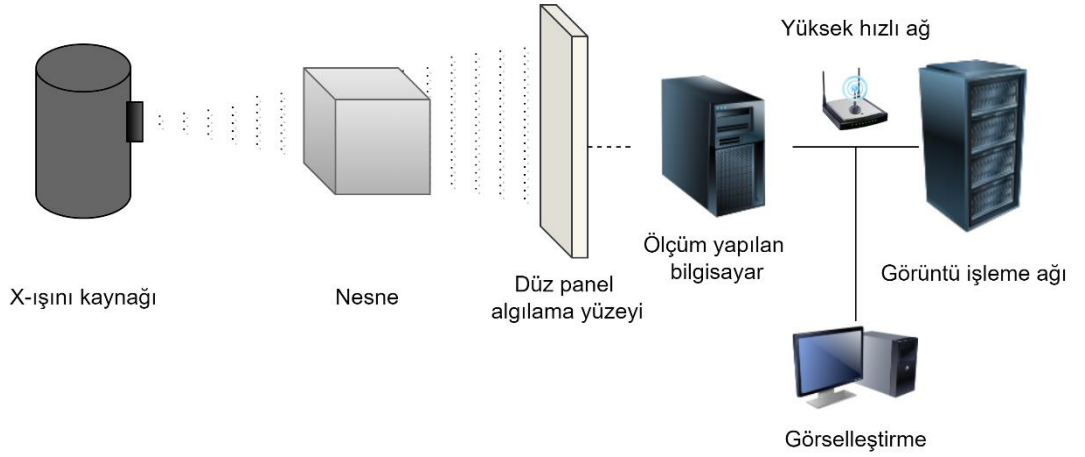
nesnelerin deęerlendirilmesi ve muayene srecini ifade eder (Gholizadeh, 2016). TM yntemleri, bir retim kalite kontrol sisteminde malzeme denetimi iin veya bireysel arařtırma iin numune bazında kullanılabilir. TM yntemlerinden biri olan radyografi ile test, radyografisi kaydedilen nesnenin i kalitesi iin kalıcı bir referans saęladığından dolayı dięer bazı yntemlere gre bir avantaja sahiptir.

Radyografi ile test, radyasyonun zelliklerine dayanmaktadır. Radyasyon, bir kaynaktan yayılarak yolu zerindeki ortamlardan geebilen elektromanyetik dalga veya paracık ıřınımı ile tanımlanır. Radyografi, radyasyonun zelliklerini kullanarak grnt yakalama uygulamasıdır. Radyasyonun temel zelliklerinden biri, malzemeler tarafından soęurulabilmesi ve soęurulan miktarın malzemenin kalınlığına gre deęiřmesidir. Radyasyon zellikle katı bir ortamdan geirilip, ortamdaki malzeme tarafından zayıflatıldıęında, malzemedeki sreksizlik blgelerinde, bu ıřınlar daha az zayıflama yařar ve dzgn blgelerdeki ıřınlara kıyasla ıřınım yoęunluęu daha fazla olur. Dolayısı ile radyasyona duyarlı film ya da grnt yakalama sensr, sreksizlięin bulunduęu alanda daha yoęun bir ıřınım alır ve bylelikle sreksizlięi grnr kılan bir grnt elde edilir. Radyografide kullanılan bařlıca iki tr radyasyon vardır: X-radyasyonu ve Gama radyasyonu. Gama ıřınlarını radyoaktif bir kaynak ve X-ıřınlarını ise vakumlu bir tp retirken, her ikisi iin de radyasyonun temel zellikleri geerlidir (Gupta ve dięerleri, 2021). X-ıřını reten tpe uygulanan voltaj, bir kaynaktan retilen X-ıřınının metallere nfuz etme yeteneęini belirler. X-ıřını hem demir ieren hem de demir iermeyen metallerdeki i kusurları tespit etmek iin kullanılabilir. Bu TM teknięi ile  boyutlu bir nesneden geen ıřınlar bir film veya sensr yzeyinde iki boyutlu grnt oluřturur. Őekil 1.2’de endstriyel X-ıřını ile TM kontrol sistemi iin Őematik bir kurulum basite gsterilmektedir.

Radyografi yntemiyle elde edilen gri tonlamalı iki boyutlu grntler, olası kusurları tespit etmek iin oęu fabrikada kalite kontrol uzmanları veya denetiler tarafından incelenir. Bununla birlikte, gz yorgunluęu, konsantrasyon kaybı ve kiřisel nyargılar dahil olmak zere, insan grsel incelemesiyle ilgili eřitli sorunlar vardır. Uzmanlar bilgisayar ekranındaki dijital radyografi (DR) grntsne bakarak, parada bir kusur olup olmadığına, varsa kusurun tipine ve derecesine ve bu bilgileri kullanarak paranın seri retime uygun olup olmadığına karar verirler. Ancak bu iřlemin ileri seviyede uzmanlařma gereksinimi, yksek iřilik maliyeti, dřk verimlilik ve doęruluk gibi



dezavantajları vardır. Manuel muayene aynı zamanda tutarsız sonuçlar verme eğilimindedir ve bu da kalite kontrol sürecinin yetersiz kalmasına neden olur. Bu durum zaman kaybı ve ek maliyetler gibi birçok olumsuzluğa neden olmaktadır. Literatürde insan ve makine performansını karşılaştıran çalışmalar yer almaktadır ve bu çalışmalarda denetim ve muayene süreci için insan denetçilerin zayıf ve güçlü yönleri belirtilmektedir (Drury, 2001; Drury ve Sinclair, 1983; Kujawińska ve Vogt, 2015; Sylla, 2002; Woldstad ve Pesante-Santana, 2006). Denetim görevinde manuel denetimden kaynaklanan dezavantajları en aza indirmek ve denetim sürecine destek olmak için doğru, tutarlı ve gerçek zamanlı bir otomatik sisteme ihtiyaç vardır.



**Şekil 1.2.** Endüstriyel X-ışını kontrol sisteminin şematik gösterimi (Hanke ve diğerleri, 2008)

Derin öğrenme alanındaki son gelişmeler, söz konusu denetim sürecinin otomatikleştirilmesine önemli bir katkı sağlamıştır ve sağlamaya devam etmektedir (Kim ve diğerleri, 2021; Zheng ve diğerleri, 2021). Derin öğrenme yöntemleri aynı anda görüntü üzerinden nesnelerin özniteliklerini öğrenip, nesnelere tespit edebilir. Derin öğrenme tabanlı tespit yöntemleri, ilgili görüntü verileri kullanılarak eğitildiği sürece, çeşitli nesnelere veya daha özel olarak kusur türlerini tespit etmek için kullanılabilir. Bu yöntemler, görüntü ön işleme ile manuel öznitelik çıkarma gerektirmediğinden genellikle geleneksel kusur tespit yöntemlerinden daha otomatik, daha esnek ve daha güvenilirdir. Son yıllarda, derin öğrenme yöntemlerinin geliştirilmesi ve hızlı ve büyük ölçekli hesaplamalara olanak sağlayan grafik işleme birimlerinin (GPU) yaygınlaşması,

bilgisayarla görmeyi de ileri seviyelere taşımıştır (X. Feng ve diğerleri, 2019; HajiRassouliha ve diğerleri, 2018). Bu gibi teknolojik gelişmeler sayesinde akıllı üretim (smart manufacturing) olanakları da artmaktadır. Akıllı üretim, üretim makinelerinin ağ ile tamamen bağlı olduğu, sensörler tarafından izlendiği ve ürün kalitesini, sistem üretkenliğini ve sürdürülebilirliğini iyileştirmek için gelişmiş hesaplama zekası tarafından kontrol edildiği yeni bir üretim paradigmasını ifade eder (Wang ve diğerleri, 2018). Akıllı üretimi desteklemek için kalite kontrol ve muayene süreçlerinin de otomatize edilmesi gerekmektedir.

Derin öğrenme, otomatik öznitelik çıkarma için çok sayıda doğrusal olmayan işlem katmanı kullanır ve her katman, bir önceki katmanın çıktısını girdi olarak kabul eder (Deng ve Yu, 2014). Böyle bir yapı ile, öznitelik çıkarma ve sınıflandırmayı tek bir çerçeveye entegre etmek ve bunları birlikte optimize etmek mümkündür. Burada son katman, beklenen etiketi çıkarmak için kullanılır ve diğer katmanlar öznitelik çıkarma görevini yerine getirirler. Devasa model ölçeğinden, ayrıntılı ağ yapısından ve etkili optimizasyon algoritmalarından yararlanan derin sinir ağları, ham görüntülerden denetimli (supervised) veya denetimsiz (unsupervised) bir şekilde doğrudan veya hiyerarşik olarak genel özniteliklerin nasıl oluşturacağını öğrenebilir (Wang ve diğerleri, 2019). Son yıllarda, özellikle nesne tanıma alanında, derin öğrenme tekniklerinin bilgisayarla görüye uygulanmasında muazzam ilerleme kaydedilmiştir (Jiao ve diğerleri, 2019; Zou ve diğerleri, 2019).

Derin öğrenme yöntemlerinden biri olan ve bilgisayarlı görü alanlarında uygulanabilen derin nesne tespit yöntemleri ile bir görüntü üzerinden nesne tespiti yapmak mümkündür. Basitçe nesne tespit yöntemleri, iyi bilinen konvolüsyonel sinir ağlarının (CNN) yaptığı görüntü sınıflandırmadan farklı olarak bir görüntü üzerinden birden fazla nesneyi konumlarıyla birlikte tespit edip sınıflandırma görevini gerçekleştirir. YBD tekniğiyle üretilen alüminyum döküm parçalarının X-ışını görüntüleri üzerinden kusur tespiti yapmak, nesne tespit yöntemleri sayesinde oldukça hızlı ve doğru bir şekilde gerçekleştirilebilir. Bu da sürecin otomatize edilmesi için atılabilecek adımlardan biridir.

Dökümhanelerin, müşterilerine ürettikleri parçalar hakkında doğru bilgi verebilmeleri için kusurun türünü, parça üzerindeki konumunu ve derecesini bilmek önemlidir. Ayrıca bu parçaların güvenlik açısından kusurları olup olmadığının tespit edilmesi

gerekmektedir. Bununla birlikte döküm parçalarında meydana gelen kusurlar döküm süreci parametreleriyle doğrudan ilişkilidir. Birinci ve ikinci faz hızları, kalıp boşluğu doldurma süresi, birinci faz uzunluğu, kalıp sıcaklığı, birinci ve ikinci soğutma periyotları, konumlandırma ve soğutma kanalı sayısı gibi döküm aşamalarındaki kusur dağılımını ve miktarını etkileyebilecek birçok birbiriyle ilişkili süreç parametresi vardır (Cica ve Kramar, 2018). Süreç parametrelerinin doğru bir şekilde seçimi ürün kalitesi açısından önem taşımaktadır. Tüm bu süreçleri bir araya getiren otomatik ve akıllı bir sistem tasarımı, döküm sürecini kararlı, yüksek doğruluğa sahip ve hızlı kılacaktır. Bu tezde amaç otomobillerde kullanılmak üzere üretilen alüminyum döküm parçalarının YBD tekniği muayene ve döküm sürecindeki denetimini ve kontrolünü otomatikleştirecek akıllı bir sistem tasarımı oluşturmaktır. Çalışmada kullanılan tüm görüntüler ve bilgiler bir dökümhaneden alınmıştır. Bu otomatikleştirme işlemi üç genel aşama içermektedir:

- I. X-ışını görüntüleri üzerinden modern nesne tespit algoritmaları kullanılarak kusurların tespitinin yapılması,
- II. Tespit edilen kusurun makine öğrenmesi algoritmaları ve modern bilgisayarlı görü yöntemleri ile bölütlenerek ikili değerlere dönüştürülmesi ve bu ikili (binary) görüntü üzerinden uluslararası bir standarda uygun olarak kusurun seviyelerinin belirlenmesi,
- III. Basınçlı enjeksiyon makinesi parametrelerinin farklı ayarları kullanılarak üretilen bir parça tipi üzerinden kusura sebep olan parametre ayarlarının belirlenmesi ve en doğru parametrelerin önerilmesi.

Bu üç aşama sonunda mevcut durumda birden fazla uzman tarafından yapılan bu işler bilgisayar gerçek-zamanlı, kararlı ve yüksek doğrulukta yapılarak sürecin otomatikleştirilmesi hedeflenmiştir. Çalışma birçok açıdan özgün değerler taşımaktadır. Tezin özgün değerleri ve topluluğa katkıları şu şekilde özetlenebilir:

- I. Bilindiği kadarıyla henüz YBD tekniği ile üretilen alüminyum parçalarının derin öğrenme yöntemleri kullanılarak kusur tespiti ve seviyelerinin belirlenmesine dair literatürdeki ilk çalışma olması,

- II. Amerikan Test ve Malzeme Derneđi (ASTM) standartlarına uygun olarak alüminyum döküm parçalarında üç tip iç kusurun otomatik bir şekilde belirlenmesi,
- III. TM arařtırmacıları için X-ışını görüntülerinden oluşan açık-eriřimli bir veri seti,
- IV. Bilgisayarlı görü ile tespit edilen iç kusurları döküm süreci parametreleri ile ilişkilendirerek en iyi süreç parametrelerini öneren bir akıllı sistem.

Çalışmanın ikinci bölümünde döküm parçalarındaki kusurların tespiti için yapılmıř çalışmalar ve nesne tespit yöntemlerine ait çalışmalar özetlenmiřtir. Üçüncü bölümde tezde kullanılan materyal ve yöntemler açıklanmıřtır. Dördüncü bölümde hazırlanan veri seti üzerinden yöntemlerin performanslarının deđerlendirilmesi ve tartıřmaya yer verilmiřtir. Son olarak beřinci bölümde sonuçlar yer almaktadır.

## 2. KAYNAK ÖZETLERİ

Bu bölümde literatürde yer alan çalışmalara yer verilmiştir. İlk alt bölümde modern nesne tespit yöntemlerinin ilk örnekleri yer almaktadır. İkinci alt bölümde ise derin öğrenme yöntemleri ile metal döküm parçalarının kusurlarının tespiti ve bölütlenmesi üzerine yapılan çalışmalar özetlenmiştir.

### 2.1. Derin Nesne Tespit Yöntemleri

Nesne tespiti, dijital görüntülerde belirli bir sınıftaki görsel nesnelerin örneklerini tespit etmeyle ilgilenen önemli bir bilgisayarla görme görevidir. Nesne tespitinin amacı, bilgisayarlı görü uygulamalarının ihtiyaç duyduğu en temel bilgi parçalarından biri olan “Hangi nesne nerede?” sorusuna yanıt sağlayan hesaplamalı modeller ve teknikler geliştirmektir (Zou ve diğerleri, 2019). Nesne tespiti, geniş uygulama yelpazesi ve son teknolojik gelişmeler nedeniyle son yıllarda oldukça ilgi görmektedir. Bu görev, anomali, hata ve kusur tespiti (Mery, 2021; Pang ve diğerleri, 2021), güvenlik (Akçay ve Breckon, 2018; Kmiec ve Glowacz, 2015), otonom sürüş (Feng ve diğerleri, 2022; Wu ve diğerleri, 2017), ulaşım gözetimi (Mhalla ve diğerleri, 2019), dron sahne analizi (Araujo ve diğerleri, 2020; Zhu ve ark. 2018) ve robotik görme (O’Keeffe ve Villing, 2018; Wan ve Goudos, 2020) gibi hem akademi hem de endüstrilerde kapsamlı bir araştırma altındadır. Son yirmi yılda, nesne tespitinin ilerlemesinin genellikle iki tarihsel dönemden geçtiği yaygın olarak kabul edilmektedir: “Geleneksel nesne tespit dönemi (2014 öncesi)” ve “derin öğrenmeye dayalı tespit dönemi (2014 sonrası)” (Zou ve diğerleri, 2019). Nesne tespit tekniklerinin hızla gelişmesine yol açan birçok çaba ve faktör arasında, derin ağlar ve GPU çok önemli yer tutmaktadırlar. Mevcut durumda derin öğrenme modelleri, tüm bilgisayarlı görme alanında yaygın olarak benimsenmiştir. Son teknoloji nesne dedektörlerinin çoğu, görüntülerden veya videolardan özellikleri çıkarmak için omurga ve tespit ağı olarak derin öğrenme ağlarını kullanır (Jiao ve diğerleri, 2019). Derin öğrenmeye dayalı modern nesne dedektörleri tek aşamalı dedektörler ve iki aşamalı dedektörler olarak iki gruba ayrılabilir. R-CNN (Girshick ve diğerleri, 2014), Hızlı R-CNN (Girshick, 2015), Daha hızlı R-CNN (Ren ve diğerleri, 2017), R-FCN (Dai ve diğerleri, 2016) ve Mask R-CNN (He ve diğerleri, 2020) gibi iki aşamalı dedektörler ilk

aşamada bölge önerileri oluşturur ve daha sonra bir sonraki aşamada sınıflandırma ve sınırlayıcı kutu regresyonu için bölge önerilerini kullanır. Bu tip dedektörler yüksek doğruluk oranlarına sahip olmalarına rağmen tek aşamalı dedektörlere göre tespit işlemini oldukça yavaş gerçekleştirirler. Ancak YOLO (Redmon ve diğerleri, 2016), YOLOv2 (Redmon ve Farhadi, 2017), YOLOv3 (Redmon ve Farhadi, 2018), YOLOv4 (Bochkovskiy ve diğerleri, 2020), YOLOv5 (Jocher, 2020), SSD (Liu ve diğerleri, 2016), EfficientDet (Tan ve diğerleri, 2020) ve RetinaNet (Lin ve diğerleri, 2017) gibi dedektörler nesnelere çok hızlı bir şekilde tespit ederler. Bunun nedeni, bu dedektörlerin nesne tespitini basit bir regresyon problemi olarak görmeleri ve bölge öneri modülünü kullanmamalarıdır. Diğer yandan bu dedektörlerin çoğu, iki aşamalı dedektörlerden daha düşük doğruluk oranlarında tespit işlemini gerçekleştirir. Ancak bu durum tüm tek aşamalı nesne tespit yöntemleri için geçerli değildir. Nesne tespit yöntemlerinin seçimi doğrudan probleminin amacıyla ilgilidir. Gerçek zamanlı tespit hedeflendiği çalışmalarda tek aşamalı dedektörlerin seçilmesi daha uygun olabilir. Ancak zaman kısıtının olmadığı ve doğru tespit oranının en önemli kriter olduğu durumlarda iki aşamalı yöntemler daha tercih edilebilir görünmektedir (Zaidi ve diğerleri, 2022). Şekil 2.1’de modern nesne tespit dedektörlerinin ortaya çıkış tarihleri kronolojik olarak verilmiştir. Ayrıca ImageNet<sup>1</sup>, PASCAL VOC (Everingham ve diğerleri, 2015), MS COCO (Lin ve diğerleri, 2014) ve Open Images V5 (Kuznetsova ve diğerleri, 2020) gibi birçok kıyaslama (benchmark) şimdiye kadar nesne tespit alanında önemli bir rol oynamıştır.

## 2.2. Döküm Kusurlarının Tespiti

Son 35 yılda döküm kusurlarının otomatik tespiti için uygulanan çeşitli bilgisayarlı görme yöntemleri tanımlamıştır. İlk çalışmalara 1980’lerin sonlarında rastlanmaktadır (Boerner ve Strecker, 1988; Filbert ve diğerleri, 1987). Bugün, endüstriyel veya akademik anlamda kullanılan dört tip yöntem tanımlanabilir (Mery, 2021):

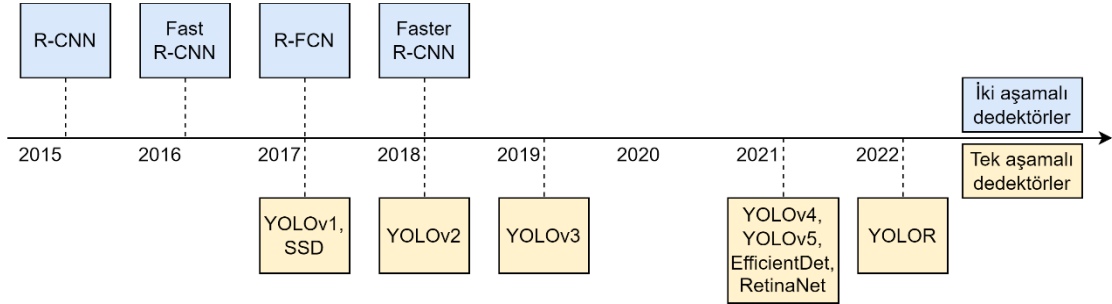
1. Klasik yöntemler,
2. Çoklu görüşlere dayalı yöntemler,

---

<sup>1</sup> <https://www.image-net.org/>

3. Bilgisayarlı tomografiye (BT) dayalı yöntemler,
4. Derin öğrenme tabanlı yöntemler.

Klasik yöntemler, geleneksel görüntü işleme ve örüntü tanıma tekniklerine dayalı yaklaşımlara karşılık gelmektedir. Bu yaklaşımlarda, döküm parçalarının otomatik denetimi için manuel olarak belirlenen öznelilikler kullanılmaktadır. Bu yöntem ailesi referans yöntemler ve yapı hakkında önceden bilgi sahibi olmayan yöntemler olarak iki ana gruptan oluşur (Mery, 2006). Referans yöntemler, hareketsiz görüntüler üzerinden karşılaştırmaya dayalıdır. Bir test görüntüsü ile referans görüntü karşılaştırılır. Önemli bir fark tespit edilirse, test parçası kusurlu olarak sınıflandırılır. Yapı hakkında önceden bilgi sahibi olmayan yöntemler, örüntü tanıma, uzman sistemler, yapay sinir ağları veya genel filtrelerin kullanıldığı yöntemlerdir. Referans yöntemlerin temel dezavantajları, karmaşık konfigürasyonlar ve tasarımdaki değişikliklere karşı esnek olmamalarıdır. Bununla birlikte, referans yöntemler, yapı hakkında önceden bilgi sahibi olmayan yöntemlere göre çok daha etkilidirler. Bunun sebebi ikinci gruptaki yöntemlerin tasarım değişikliklerine uyum sağlamak için kullanılan otomatik uyarılama süreçlerinin kusursuz olmamasıdır (Mery, 2021).



**Şekil 2.1.** Derin öğrenme tabanlı nesne dedektörleri kronolojik sırası (Parlak ve Emel, 2023)

Mery (2003), görüntünün ortasında yer alan her bir potansiyel kusuru kesen düz çizgiler boyunca çıkarılan özelliklere dayanan bir yaklaşım önermiştir. Hernández ve diğerleri (2004), kusurları tespit etmek için sinirsel bulanık (neuro-fuzzy) yöntemine dayalı bir kusur tespit yaklaşımı önermişlerdir. Li ve diğerleri (2006), döküm parçalarının X-ışını görüntülerini kullanarak iç kusurlarını tespit etmek için Dalgacık (Wavelet) tekniğinden

faydalanmışlardır. Tang ve diğerleri (2009), dökümlerdeki tipik iç kusurların tespiti için bir bölütleme yöntemi önermişlerdir. Önerdikleri yöntem, bulanık küme teorisinden yararlanmaktadır. Ramírez ve Allende (2013), alüminyum dökümlerde kusur tespiti için iki üretken (generative) ve ayırt edici (discriminant) örüntü tanıma algoritmasını karşılaştırmışlardır.

Cogranne ve Retraint (2014), hipotez testi yaklaşımı ile radyografik görüntülerin içeriğine uyarlanmış bir istatistiksel modelden yararlanarak kusurları tespit etmişlerdir. Zhao ve diğerleri (2014), radyografi görüntülerindeki döküm kusurlarını otomatik olarak tespit etmek için gri tonlamalı düzenleme çiftleri (GAP) özelliğine dayalı bir yöntem önermişlerdir. Zhao ve diğerleri (2015), radyografi görüntülerindeki gizli döküm kusurlarının tespiti ve sınıflandırılması için güçlü, seyrek temsile dayalı bir inceleme sistemi sunmuşlardır. Sistemde çatlaklar, gaz boşlukları, çekinti ve porozite dahil olmak üzere dört yaygın döküm kusuru türü dikkate alınmaktadır. Li ve diğerleri (2016), otomobil jantında kusur görüntülerinin tespiti ve tanınmasının doğruluğunu iyileştirmek için bazı klasik yöntemleri kullanmışlardır. Janttaki kusurun bölgesini çıkarmak için geliştirilmiş bir tepe konumu (peak location) algoritması, jant kusurunu sınıflandırmak ve tanımak için sinir ağı ile birlikte önerilmiştir. Kamalakannan ve Rajamanickam (2017), gaz boşluğu ve soğuk dolguyu (cold fill) tespit etmek için uzamsal tabanlı etkili bir bölütleme yöntemi önermişlerdir. Mery ve Arteta (2017), taşıt araçları parçalarında kusurlu ve kusursuz 32×32 piksel boyutlarında yaklaşık 47 500 adet kırılmış X-ışını görüntüsü içeren yeni bir veri seti yayınlamışlardır. Bu veri setini kullanarak 24 farklı bilgisayarlı görme tekniğini değerlendirip karşılaştırmışlardır. En iyi performansın, %97 kesinlik (precision) ve %94 duyarlılık (recall) ile Support Vector Machines (SVM)-doğrusal sınıflandırıcıya sahip basit bir LBP tanımlayıcısı ile elde edildiğini rapor etmişlerdir.

Düşük basınçlı dökümde, alüminyum alaşımlı jantlar, gaz boşlukları ve çekinti boşlukları gibi, kaliteyi sağlamak için X-ray denetimi gerektiren iç kusurları barındırmaya eğilimlidir. Zhang ve diğerleri (2018) bu problemi çözmek için X-ışını görüntüleri üzerinden uyarlanabilir eşik (adaptive threshold) bölütleme algoritması önermişlerdir. Jin ve diğerleri (2020), Relief algoritmasına ve Adaboost-SVM algoritmasına dayalı bir iç çatlak kusurlarını tespit yöntemi önermişlerdir. İlk olarak radyografi görüntülerine gri tonlamalı dönüşüm uygulanmıştır. Ayrıca görüntüler bazı filtreler ve görüntü bölütleme



tekniki ile önışlemlere tabii tutulmuştur. Daha sonra yönlendirilmiş gradyanların histogramı (HOG) özniteliđi, deđişmez moment özniteliđi ve LBP özniteliđi çıkarılmış ve Relief algoritması ile hassas öznitelik kümesi seçilmiştir. Son olarak, Adaboost-SVM, çatlak tespitini yüksek doğrulukla gerçekleştirmek için iç çatlak tespit modelini olarak kullanılmıştır.

Çoklu görüşlere dayalı yöntemler, tek bir görüşün belirsizliğinin yanlış yorumlamaya yol açabileceđi karmaşık nesnelere incelerken çok etkili olabildikleri için, birden çok görüşe dayalı yaklaşımlar olarak önerilen yöntemlerdir. Bu yöntemlerin tipik olarak iki temel adımı vardır. İlk adım, her görüşte olası kusurların belirlenmesidir. Bu adımı gerçekleştirmek için manuel verilen veya önceden öğrenilmiş öznitelikler kullanılabilir. İkinci adımda, olası kusurlar eşleştirilir ve birden çok görüşte izlenir. Bu yöntemdeki temel fikir, izlenemeyen olası kusurları yanlış alarm olarak değerlendirmektir.

Mery ve Filbert (2002), kusurları otomatik olarak tespit etmek için çoklu görüşe dayalı tek bir filtre kullanan iki aşamalı bir teknik önermişlerdir. Pieringer ve Mery (2010), 3B uzayda kayan pencereden elde edilen bir noktadan tüm projeksiyonları toplayarak, simüle edilmiş kusurları tespit etmeyi öğrenmek için bir sınıflandırıcıyı eğitmişlerdir. Carrasco ve Mery (2011), otomatik çoklu görüş incelemesi adımı verdikleri bir yöntem kullanarak farklı konumlardan alınan görüntüler ile alüminyum jantların incelenmesi için bir yöntem sunmuşlardır. Yöntem, kalibre edilmemiş görüntü dizilerine uygulanabilir. Bu nedenle kalibre edilmiş sistemlerde normalde mevcut olan optik ve geometrik parametrelerin belirlenmesi gerekli değildir. Mery ve diğerleri (2013), yüksek düzeyde tanımlanmış şekil ve boyutlara sahip belirli nesnelere tanımak için çoklu X-ışını görüntülerine dayalı otomatik bir yöntemin kullanılmasını önermişlerdir. Mery (2015), çoklu X-ışını görüntülerini kullanarak karmaşık bir nesnenin içindeki ilgi çekici kısımları belirlemek için beş adımdan oluşan yeni bir yaklaşım sunmuştur.

Diđer yöntem gruplarının aksine bilgisayarlı tomografiye dayalı yöntemler, test nesnesinin hacimsel olarak yeniden yapılandırmasını üretir. Bir 3B hacim, yani test edilen nesnenin dilimlerinin bir dizi 2B görüntüsü, yeniden yapılandırma yaklaşımları kullanılarak tahmin edilir. BT, yeterli sinyal-gürültü oranları için minimum ölçüm süresinin yanı sıra istenen yerel çözünürlük için minimum sayıda projeksiyon gerektiren

çok zaman yoğun bir süreç olabilir (Domingo Mery, 2021). Sonuç olarak bu gibi yöntemler gerçek-zamanlı muayene süreci için her zaman uygun olmayabilir.

Hangai ve diğerleri (2010), parlatma makinesi ve optik mikroskop içeren bir seri kesit sistemi geliştirmişlerdir. Seri kesit sistemi kullanılarak yapılan gözlem, bir kümelenmiş çekinti gözenekleri son derece karmaşık bir şekil oluşturan birbirine bağlı birçok küçük gözenekten oluştuğunu açıkça göstermiştir. Li ve diğerleri (2019) araştırmalarında, bilgisayarlı tomografinin net şekle yakın döküm parçalarındaki iç döküm kusurlarını belirlemek için etkili bir yöntem olduğunu göstermişlerdir. Bandara ve diğerleri (2020), endüstriyel alüminyum alaşımlarının kalıp dökümünün iç kusurlarını incelemek, süper sızdırmazlık reçinelerinin küçük delikleri doldurmadaki yeterliliğini doğrulamak ve metallerin döküm gözenekliliğini azaltmak için BT kullanımını açıklamışlardır.

Bu tez çalışmasında kullanılan tespit yöntemi derin öğrenme yaklaşımlarına dayalı bir yöntem olduğu için derin öğrenme tabanlı yöntemler farklı açılardan ele alınmıştır. Çizelge 2.1’de döküm kusurları tespiti için kullanılan derin öğrenme tabanlı yöntemlerin kullanıldığı çalışmalar yer almaktadır. Literatürde yer alan çalışmaların önemli bir kısmında alüminyum parçalarının X-ışını görüntüleri kullanılmıştır. Buna ek olarak TM topluluğu için açık kaynak olan yalnızca iki adet döküm parçalarına ait X-ışını görüntülerinden oluşan veri seti mevcuttur. Bunlardan ilki birçok çalışmada da kullanılmış olan GDXray (Mery ve diğerleri, 2015) veri setidir. Bu veri setinde beş farklı veri grubu vardır: Döküm (alüminyum jantlardan ve mafsallardan oluşan toplam 2727 görüntü), kaynak, bagaj, doğa ve dekor. Veri setinde, alüminyum döküm parçalarının döküm tekniği ile ilgili bir bilgiye yer verilmemiştir. İkinci bir veri seti ise bu tez çalışması kapsamında oluşturulan Al-Cast (Parlak ve Emel, 2023) veri setidir. Bu veri seti, taşıt aracı bileşenleri olarak YBD tekniği ile üretilen alüminyum döküm parçalarının X-ışını görüntülerini içermektedir. Veri setinde yedi farklı parçaya ait görüntüler yer almakta olup veri seti toplam 5025 adet görüntü içermektedir.

**Çizelge 2.1.** Döküm kusurlarının derin öğrenme yöntemleriyle tespiti ve bölütlenmesi

<b>Kaynak</b>	<b>Döküm malzemesi</b>	<b>Açıklamalar</b>
Ferguson ve diğerleri (2017)	Alüminyum	GDXray veri seti ile Daha hızlı R-CNN, R-FCN ve SSD kullanarak kusur tespiti yapılmıştır.
Mery ve Arteta (2017)	Alüminyum	GDXray veri seti ile 24 farklı CNN yapısı (XNet, VGG, AlexNet ve GoogleNet dahil) kullanılarak görüntülerin kusurlu veya kusursuz olarak sınıflandırılması yapılmıştır.
Ferguson ve diğerleri (2018)	Alüminyum	X-ışını görüntülerindeki kusurların tespiti için Mask R-CNN kullanılmıştır.
Lin ve diğerleri (2018)	Malzeme belirtilmemiş	Görsel dikkat mekanizmasına ve öznitelik haritasının derinlemesine öğrenilmesine dayalı güçlü bir tespit yöntemi önerilmiştir.
Du ve diğerleri (2019)	Alüminyum	Faster R-CNN ve FPN kullanarak özgün bir veri seti ile kusur tespiti yapılmıştır.
Hu ve Wang (2020)	Malzeme belirtilmemiş	İki modülden oluşan bir CNN yaklaşımı kusur tespiti için önerilmiştir.
Mery (2020)	Alüminyum	Geliştirdikleri ve Xnet-II adını verdikleri bir CNN modeli ile GDXray veri setindeki görüntülerde yer alan kusurların tespiti gerçekleştirilmiştir.
Duan ve diğerleri (2021)	Malzeme belirtilmemiş	İki farklı kusur tespiti için (farklı boyutlar için ayrı ayrı) YOLOv3 ve değiştirilmiş YOLOv3 kullanılmış ve yöntemler karşılaştırılmıştır.
Mery (2021)	Alüminyum	GDXray görüntüleri üzerinden sekiz farklı modern nesne tespit algoritmasının (YOLOv3, YOLOv5, RetinaNet, EfficientDet dahil) performansları karşılaştırılmıştır.
Jiang ve diğerleri (2021)	Alüminyum	VGG-16 ile özgün bir veri seti üzerinden kusur tespiti gerçekleştirilmiştir.
Tang ve diğerleri (2021)	Malzeme belirtilmemiş	Kusur tespiti için uzamsal dikkat mekanizmasına sahip bir CNN yapısı önerilmiştir.
García Pérez ve diğ. (2022)	Alüminyum	GDXray veri seti kullanılarak FPN eklenmiş RetinaNet yöntemi ile kusur tespiti yapılmıştır.
Yagüe ve diğerleri (2022)	Magnezyum	Detectron2 nesne tespit yöntemi ile kusur bölütlemesi gerçekleştirilmiştir.
Liu ve Kim (2022)	Alüminyum ve Magnezyum	Piramit modeline dayalı bir ön işleme yöntemi ile UNet++ bölütleme yöntemi bir arada kullanılarak kusur bölütlemesi yapılmıştır.
Du ve diğerleri (2022)	Alüminyum	IXNet adını verdikleri interaktif kusur bölütleme yöntemi önerilmiştir.
Parlak ve Emel (2023)	Alüminyum	YBD tekniği ile üretilen alüminyum parçalarından oluşan özgün bir veri seti ile modern nesne tespit yöntemleri kusur tipini de içerecek şekilde eğitilmiş ve performansları karşılaştırılmıştır.

Çizelge 2.1’de yer alan çalışmalarda kullanılan yöntemler üç farklı derin öğrenme yaklaşımı altında sınıflandırılabilir:

1. Yalnızca CNN kullanılarak yapılan sınıflandırmalar
2. Sınırlayıcı kutular kullanılarak eğitilen nesne tespiti
3. Piksel bazında gerçekleştirilen bölütleme

CNN kullanılarak yapılan çalışmalarda görüntüler bütünsel olarak ele alınmaktadır ve her bir görüntü, CNN çıktısında kusurlu veya kusursuz olarak sınıflandırılmaktadır. Bu yaklaşımla yapılan kusur tanıma işlemleri kusurla ilgili konum bilgisi içermediği için çok kullanışlı değildir. Ancak diğer yöntemlere göre temel-gerçek (ground-truth) olarak adlandırılan kutuların oluşturulmasını gerektirmediğinden oldukça hızlı uygulanabilir. Bu gruba ait çalışmalar Çizelge 2.1’de az sayıda yer almaktadır. Nesne tespiti yaklaşımlarından faydalanılan çalışmalar Çizelge 2.1’de en sık rastlanan çalışmalardır. Nesne tespit yöntemlerinin çıktısında hem nesnenin konum bilgileri hem de nesneyi sıkı bir şekilde kapsayan bir sınırlayıcı kutu yer almaktadır. Böylece hem parça üzerindeki kusurun konumuna hem de kusuru içeren çerçeve boyutlarına ulaşılabilir. Bu avantajı sayesinde bu yöntem grubu oldukça pratiktir. Ancak yöntemlerin eğitimi için bir uzman tarafından temel-gerçek kutularının çizilmesi yani etiketleme işleminin yapılması gerekmektedir. Bununla birlikte ilk grup yöntemlere göre eğitim süreleri çok daha uzundur.

Çizelge 2.1’de yer alan çalışmalarda kullanılan son yöntem grubu da bölütleme yöntemleridir. Bu gruba ait yöntemler nesnenin ya da daha özel olarak kusurun yalnızca yerini vermekle kalmaz aynı zamanda piksel bazında kusuru gösteren bir çıktı da sunar. Tespit yöntemlerine göre çıktı açısından çok daha pratik bilgiler sunsa da bölütleme birçok dezavantaja sahiptir; örneğin diğer iki gruba göre çok daha yüksek eğitim sürelerine sahiptir. Buna ek olarak tespit yöntemlerinde kutular çizilerek yapılan etiketleme işlemi bu yöntemlerde kusurun sınırlarında piksel bazında çizilmek zorundadır. Bu da etiketleme işlemi ciddi anlamda yorucu ve uzun bir sürece çevirmektedir. Tüm bu sebeplerden dolayı kusur tespiti için genellikle nesne tespit yöntemleri tercih edilmiştir. Nesne tespitine ek olarak, bu tez çalışmasında kusurların

tipleri ile birlikte seviyeleri üzerine tanımlanmış ASTM standartlarına uygun olarak yapılan bir çalışma, bilindiği kadarıyla literatürde mevcut değildir.

### **2.3. Basınçlı Döküm Süreci Parametrelerinin Optimizasyonu**

Farklı ayar seviyelerine sahip çeşitli döküm süreci parametreleri, kaliteli döküm parça üretiminde en uygun parametre değerlerini içeren bir model geliştirmeyi son derece zorlaştırmaktadır. Geleneksel deneme yanılmaya dayalı döküm kalıp süreci geliştirme görevi, kapsamlı deneysel çalışma gerektirdiğinden pahalı ve zaman alıcıdır (Cica ve Kramar, 2018). Basınçlı döküm süreci gibi karmaşık ve çok değişkenli imalat problemlerinde ise deneysel tasarım ile deney sayısını azaltarak önemli kalite avantajları elde edilebilir. Anastasiou (2002), birçok döküm parametresi arasından önemli olduğunu ileri sürdüğü tutma fırını sıcaklığı, döküm metalin kalıp sıcaklığı, piston hızı (birinci ve ikinci aşama) ve hidrolik basınç (üçüncü aşama) gibi parametreleri üzerinde Taguchi yöntemi (Taguchi, 1987) ile deneysel tasarım yapmıştır ve parametrelerin farklı seviyelerdeki ayarlarının döküm kalitesi üzerindeki etkilerini incelemiştir. Diğer parametreler tüm deney boyunca sabit tutulmuştur. Benzer bir şekilde Syrcos (2003),  $AlSi_9Cu_{13}$  alüminyum alaşımının döküm sürecinin optimal parametre ayarlarını elde etmek için Taguchi yöntemini kullanmıştır. Bu çalışmada ele alınan parametreler piston hızı (birinci ve ikinci aşama), metal sıcaklığı, dolum süresi ve hidrolik basınçtır. Deney sonucunda ele alınan parametrelerin döküm ürününün yoğunluğunu önemli ölçüde etkilediklerini göstermiştir. Tsoukalas ve diğerleri (2004), gözenek oluşumunda süreç parametrelerinin önemini araştırmak için Taguchi yaklaşımına dayalı bir deneysel tasarım gerçekleştirmişlerdir. Ayrıca, optimum parametre ayarları altında beklenen kalite iyileştirmesi ve maliyet tasarruflarını değerlendirmişlerdir. Wang ve diğerleri (2014), Al-Cu-temelli ZL205A alaşımındaki ciddi çekinti gözenekliliği eğilimini azaltmak için düşük basınçlı döküm ile üretilen silindirik kabuk ZL205A alaşım parçalarının proses parametrelerini optimize etmek amacıyla Taguchi yöntemi kullanmışlardır. Dökme sıcaklığının, doldurma süresinin ve paketlenme basıncının ZL205A dökümlerinde büzülme gözenekliliği üzerindeki etkisi araştırılmıştır. Kittur ve ark. (2015), döküm sürecindeki girdi-çıkı ilişkilerini Yanıt Yüzeyi (Response Surface) yöntemi kullanarak incelemişlerdir. Hızlı atış hızı, enjeksiyon basıncı, faz geçiş noktası ve bekletme süresi

gibi proses parametreleri modele girdi olarak kabul edilmiştir. Gerçekleştirilen deneylere dayalı olarak, merkezi kompozit tasarım ve Box-Behnken tasarımı kullanılarak iki doğrusal olmayan model geliştirilmiştir. Bunlara ek olarak maliyetleri azaltmak ve verimliliği artırmak için bazı araştırmacılar, basınçlı döküm işleminin fiziksel olaylarını daha iyi anlamak adına simülasyon araçlarını kullanmışlardır (Ding ve diğerleri, 2014; Keste ve diğerleri, 2016; Yourui ve diğerleri, 2016).

Yaklaşık olarak son 20 yılda, yüksek basınçlı dökümün çeşitli süreç parametrelerinin tahmini ve optimizasyonu için birçok araştırmacı tarafından yapay zekaya dayalı yöntemler kullanılmıştır. Yarlagadda ve Chiang (1999), geri yayılım, momentum ve uyarlanabilir öğrenme algoritması olmak üzere üç farklı algoritma kullanarak basınçlı döküm işlemi için süreç parametrelerini oluşturmak adına yapay sinir ağları (YSA) tabanlı bir model geliştirmişlerdir. Shafyei ve diğerleri (2006), Al-Si döküm alaşımlarındaki gözeneklilik yüzdesini tahmin etmek için YSA tabanlı bir yöntem geliştirmişlerdir. Rai ve diğerleri (2008), optimal yüksek basınçlı döküm proses parametrelerinin gerçek zamanlı tahmini için genel bir YSA tabanlı model önermişlerdir. Krimpenis ve diğerleri (2006), döküm proses parametreleri ile kusur ve katılma süresi gibi çıktı değişkenleri arasındaki ilişkiyi simüle etmek için YSA meta-modellerini uyguladılar. Eğitilmiş YSA, kalıp döküm süreçlerini optimize etmek için bir genetik algoritmanın (GA) amaç fonksiyonu olarak kullanılmıştır. Zhang ve Wang, (2013), düşük basınçlı döküm işlemini optimize etmek için birleştirilmiş bir YSA ve GA yöntemi önermişlerdir. YSA modeli, basınçlı döküm sürecinin parametreleri ve kalite indeksleri arasındaki karmaşık ilişkiyi haritalamak için kullanılırken; GA, eğitilmiş YSA modeline dayalı amaç fonksiyonu ile basınçlı döküm süreci parametrelerini optimize etmek için kullanılmıştır. Anijdan ve diğerleri (2006), Al-Si döküm alaşımlarındaki minimum gözeneklilik yüzdesi ile ilişkili optimal süreç parametrelerini bulmak için birleşik GA-YSA yöntemini geliştirmişlerdir. Tsoukalas (2008), AlSi<sub>9</sub>Cu<sub>3</sub> alüminyum alaşımlı dökümlerin minimum gözeneklilik yüzdesi ile ilişkili optimal süreç parametrelerini bulmak için etkili yöntemler olarak lineer regresyon ve GA önermiştir. Manjunath Patel ve diğerleri (2016), genetik algoritma kullanarak döküm işleminin çok amaçlı optimizasyonunu gerçekleştirmişlerdir. Chiang ve diğerleri (2008), magnezyum alaşımlı parçaların döküm süreci parametrelerinin optimal ayarını oluşturmak için gri ilişkisel analiz ve bulanık mantığı kullanmışlardır. Rao ve diğerleri (2014), basınçlı döküm

işleminin parametre optimizasyonu için öğretim-öğrenme tabanlı optimizasyon adı verilen yeni bir optimizasyon yöntemi geliştirmişlerdir. Cica ve Kramar (2018), alüminyum basınçlı dökümlerin poroziteden kaynaklanan hurda miktarını en aza indirerek kalitesini artırmak için geliştirilen tahmine dayalı modellerin tasarım ve uygulama sorunlarını incelemiştir. Çalışmada, döküm parçalarının gözenekliliğini, karşı basınç, birinci faz hızı, birinci faz uzunluğu, ikinci faz hızı, birinci soğutma periyodu ve ikinci soğutma periyodunun bir fonksiyonu olarak ifade edilmiştir. Geliştirdikleri GA ve TB tabanlı bulanık sistemlerin, basınçlı dökümlerde gözeneklilik konusunda büyük tahmin kabiliyetine sahip olduğunu belirtmişlerdir. Martínez-Pastor ve Hernández-Ortega (2022), bazı optimizasyon modellerini kullanarak, kaliteli parçaların üretilmesini sağlayan makine parametrelerinin kurulumunu oluşturan bir Karar Destek Sistemi (KDS) tasarlanmışlardır. Bu KDS'yi kullanarak kullanıcı, döküm malzemesi, kalıp veya döküm makinesi ile ilgili girdi değişkenlerinin değerlerini ayarlayabilir ve daha sonra sistem tarafından ilgili hesaplamalar yapılır ve sonuçlar, diğerleri arasında maksimum doldurma süresi, maksimum doldurma oranı veya piston hız profili gibi belirli çıktı değişkenleri cinsinden ifade edilir. Ancak bu çalışmaların arasında otomatik bir şekilde bulunan kusurun detayı ele alınarak döküm süreç parametrelerinin optimizasyonu ile ilgili bir çalışma mevcut değildir.

Bu tez çalışmasında YBD tekniği ile alüminyum döküm parçaları üreten bir dökümhaneden alınan X-ışını görüntüleri kullanılarak kusur tespiti yapılmıştır. Kusurlar ASTM standartlarına göre tanımlanmış ve dökümhanede karşılaşılan ve bu standartlara uyan üç farklı kusur tipi ele alınmıştır. Ek olarak yine ASTM standartlarına uygun bir şekilde bir makine öğrenmesi algoritması ve farklı görüntü işleme algoritmaları kullanılarak kusurların seviyesi belirlenmiştir. Tüm bu işlemler bilgisayar tarafından otomatik olarak yapılmaktadır. Bu da muayene sürecinin otomasyonu için önemli bir adımdır. Ayrıca verilerin alındığı dökümhanede tek bir parça tipi için farklı üretim parametreleri kullanılarak bir dizi üretim gerçekleştirilmiştir. Bu üretimin amacı kusurlu parçalarla döküm süreci parametreleri arasında ilişki kurmaktır. Böylece tespit edilen ve seviyeleri belirlenen kusurların süreç parametreleri kullanılarak kusurun önüne geçebilecek döküm ayarlarının önerisinin otomatik olarak yapılması amaçlanmaktadır ve bu çalışma tüm bu adımlarıyla akıllı bir muayene ve üretim süreci için önemli bir potansiyel taşımaktadır.

### 3. MATERYAL ve YÖNTEM

Yerel bir dökümhanede mevcut durumda mühendisler tarafından gerçekleştirilen muayene sürecinin bu çalışmada otomatize edilmesi hedeflenmiştir. Bu muayene süreci genel olarak kusurların tespit edilmesi ve uluslararası bir standarda göre kusur seviyelerinin belirlenmesini içerir. Çalışma bu sürecin yanı sıra, döküm süreci parametrelerini de kullanarak kusur bilgisinden bu parametrelerin en uygun değerlerinin belirleneceği bir deneysel tasarım ve optimizasyonu da içermektedir.

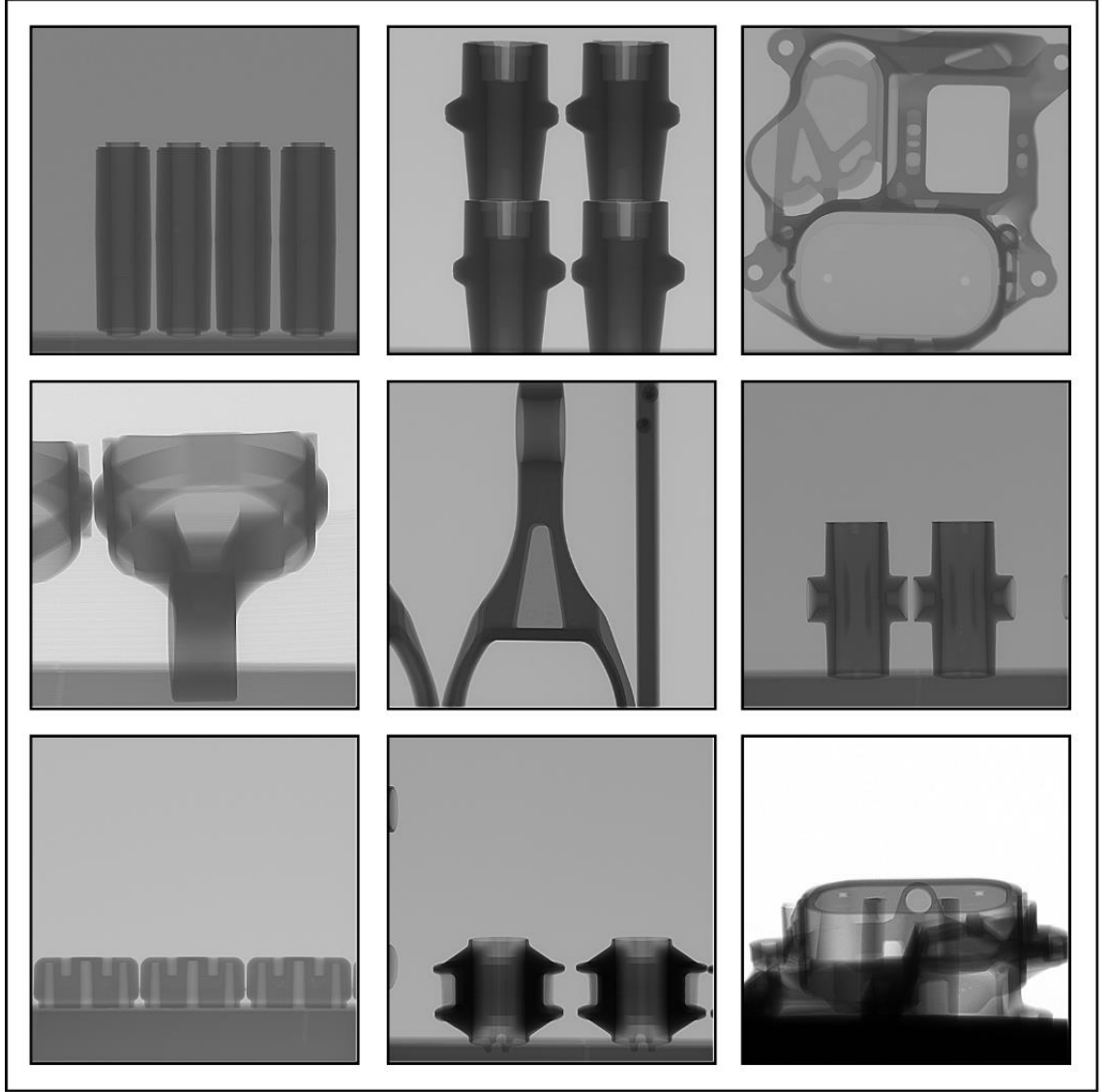
#### 3.1. Materyal

Bu bölümde çalışmada kullanılan veri setine ait detaylar ve bu verilerin kullanım şekli ele alınmıştır. Bununla birlikte döküm kusurlarının karakteristik özellikleri ve çalışmada kullanılan ASTM standartlarına ait bilgiler yine bu bölümde irdelenmiştir.

Tez çalışmasında yer alan veriler otomobillerde kullanılmak üzere YBD tekniği ile üretilen alüminyum döküm parçalarına ait 2B X-ışını görüntülerinden oluşmaktadır. Görüntüler YXLON MU2000-D cihazı ile elde edilmiştir. Her bir görüntünün piksel çözünürlüğü 1000×1000'dir ve görüntüler TIFF formatında alınmıştır. Ayrıca görüntülerin tamamı 8-bit gri tonlamalıdır. Veri seti yedi farklı parça tipine ait görüntüler içermektedir. Görüntülerdeki parçaların bir kısmında herhangi bir kusur bulunmazken, çoğu parçada farklı sayı, boyut ve türde kusurlar bulunmaktadır. Tüm parçalar için X-ray görüntüleri oluşturmak adına yapılan işlemlerde kaynaktan filme ve parçadan filme mesafeler eşit olarak alınır. Böylece tüm görüntüler için standart bir ölçüm sağlanmıştır. Şekil 3.1'de veri setinde yer alan bazı örnek X-ışını görüntüleri verilmiştir.

Alüminyum alaşımları için diğer döküm yöntemleri gibi, YBD tekniği de "kusur oluşturan bir süreç" olarak kabul edilebilir. Döküm sonucunda parça içerisinde birçok döküm süreci parametresinden kaynaklanan çeşitli kusurlarla karşılaşılabilir. Bu kusurlar farklı özelliklere sahiptir ve ayrıca çeşitli boyut ve yayılımlarda olabilir. Veri setindeki kusurları analiz etmek için ASTM International (Amerikan Test ve Malzeme Derneği) standartları kullanılmıştır.





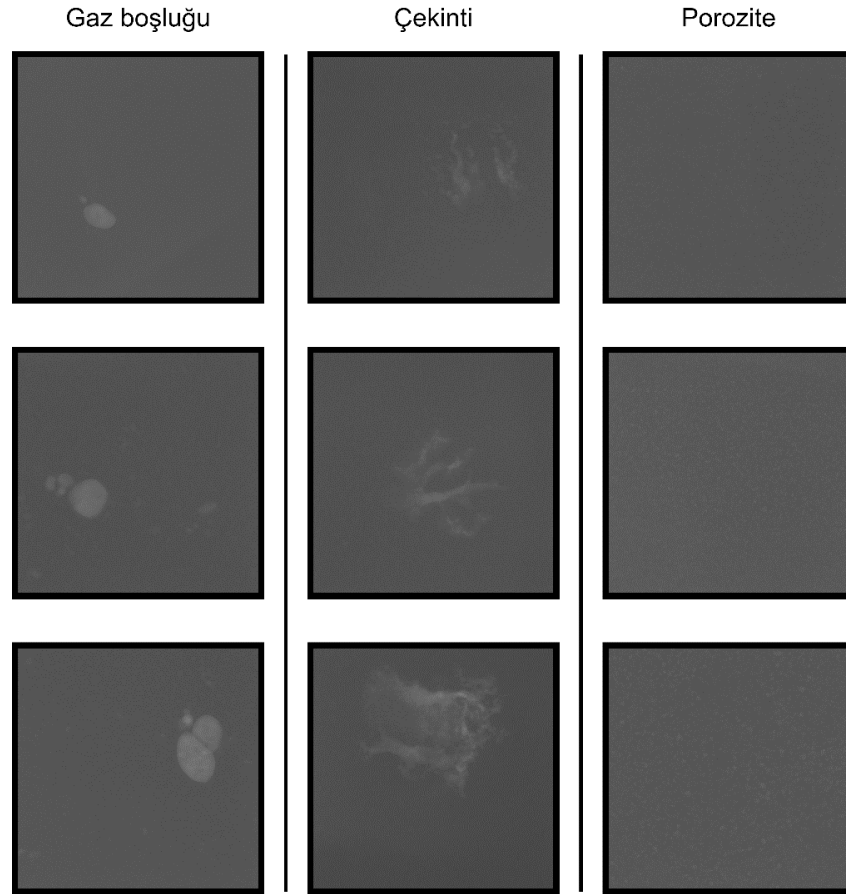
**Şekil 3.1.** Veri setinde yer alan bazı görüntü örnekleri

ASTM, endüstriler için performans ve güvenlik standartları geliştiren tanınmış bir kuruluştur. Alüminyum dökümlerin muayenesi için ASTM E155 (ASTM International, 2014) standart referansı birçok iç kusur kategorisini tanımlamaktadır. Referans radyografiler 50 mm kalınlığa kadar alüminyum dökümlere uygulanabilir. Çekirdek kaydırma, soğuk kapatma ve yüzey kusurları bu standartlarda gösterilmemiştir. Bu tür istenmeyen durumlar, görsel muayene veya farklı tekniklerle kolaylıkla tespit edilebilir. Bunun yanı sıra X-ışını enerjisi görüntü kalitesini etkiler ve kullanılan kaynağın enerjisine göre farklı avantajlar ve dezavantajlar ortaya çıkabilir. Veri setindeki görüntüler farklı piksel yoğunluklarına sahiptir. Bunun sebebi X-ışını cihazında farklı

parametrelerin kullanılmasıdır. Bu parametreler tez çalışmasında göz ardı edilmiştir. Kusur sınıflandırma ve tespit işlemi bu uluslararası standartlardaki referans bilgi ve görsellere uygun olarak yapılmıştır. Veri setinde yer alan görüntülerde üç farklı kusur tipi vardır:

- Gaz boşluğu (gas hole)
- Çekinti (shrinkage)
- Porozite (porosity)

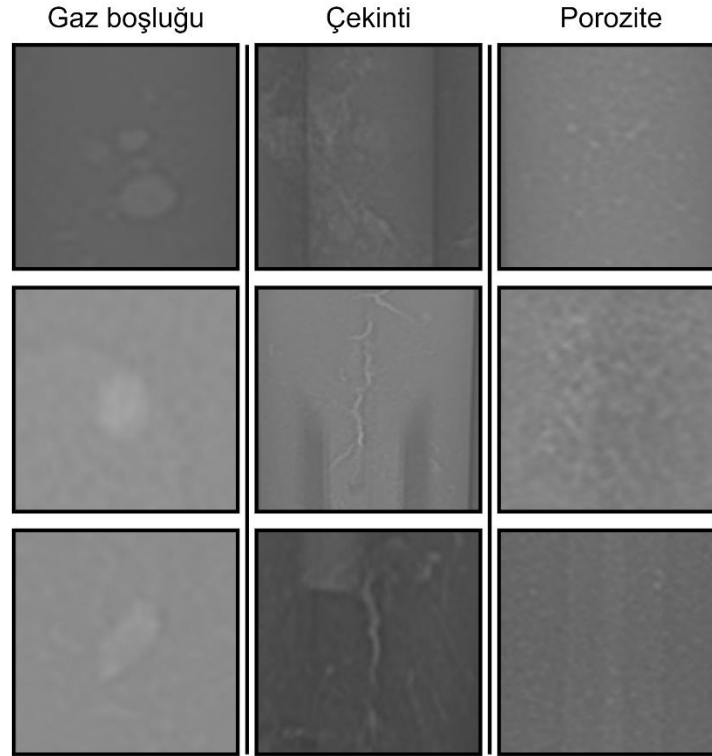
Bu üç kusur tipine ait ASTM referanslarında yer alan bazı görüntüler Şekil 3.2’de, Al-Cast veri setinde yer alan benzer kusurlar ise Şekil 3.3’te verilmiştir.



**Şekil 3.2.** ASTM referanslarında yer alan bazı kusur görüntüleri (ASTM, 2001)

Gaz delikleri tek tek veya kümeler halinde görünebilir ve genellikle dairesel veya eliptik bir görünüme sahip, sınırları net alanlardır. Porozite, genellikle geniş bir alana yayılan ve

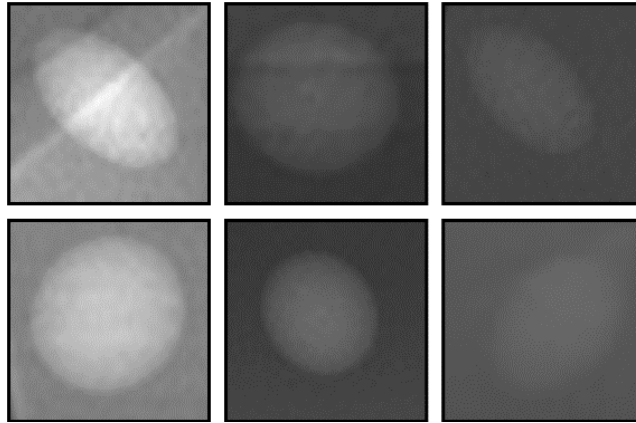
hafifçe yuvarlak veya uzun noktalar olarak görülebilen küçük boşluklardır. Genellikle diğer kusur tiplerine göre sınırlandırmaları ve görülmeleri kolay değildir. Çekinti, boşluk ve gözeneklilik olmak üzere iki farklı tipte gözlemlenebilir. Çekinti boşluğu dentritik, ipliksi veya çatlak şeklinde açık renkli alanlar olarak gözlenirken, çekinti gözenekliliği dantelli veya bal peteği görünümlü açık renkli alanlar olarak gözlenir. Bu çalışmada çekinti adı altında iki çekinti türü tek bir kusur tipi olarak ele alınmıştır.



**Şekil 3.3.** Veri setinde yer alan görüntülerden kırılmış bazı kusur örnekleri

Toplam görüntü sayısı 187 olan bu veri seti derin öğrenme yöntemlerinin eğitimi için yetersizdir. Bu nedenle toplam görüntü sayısını eğitim için yeterli olabilecek bir seviyeye çekmek adına iyi bilinen bazı veri artırım teknikleri kullanılmıştır. Her bir görüntü için rotasyon, kaydırma ve keskinleştirme olarak adlandırılan basit veri artırım teknikleri uygulanmıştır. Rotasyon işlemi ile 15°'lik açılarla döndürülerek her bir görüntüden 23 adet yeni görüntü ve kaydırma işlemi ile rassal olarak farklı sayılarda hem yatay ekseninde hem de dikey ekseninde kaydırmalar yaparak yeni görüntüler elde edilmiştir. Oluşan yeni görüntüler üzerinden rastgele seçilen görüntülere keskinleştirme işlemi uygulanmıştır. Bununla birlikte veri setinde yer alan kusur tiplerinin sayıları hesaplanmıştır. Gaz boşluğu

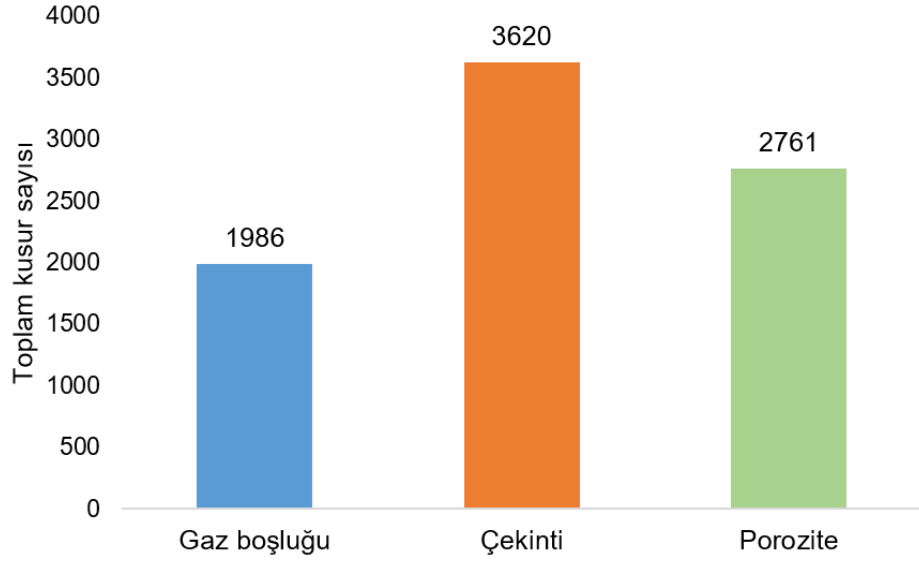
kusurlarının sayısı diğer kusur tiplerinin sayısının çok altında olduğu için elipsoit kusurların simülasyonu<sup>2</sup> (Mery, 2021) tekniği kullanılarak gaz boşluklarının sayısı rastgele seçilen görüntüler üzerinde artırılmıştır. Herhangi bir boyut ve oryantasyondaki elipsoit şekilli bir kusurun simülasyonu, dökümün herhangi bir konumunda gerçekleştirilebilmektedir. Simülasyonda, elipsoitin boyutunun, simüle edilen kusurun yansıtıldığı dökümün kalınlığından daha büyük olmaması gerekir. Yöntem rastgele boyutlarda oluşturulan elipsoitin 2B görüntü üzerine izdüşümünün eklenmesi (superimposition) şeklinde uygulanır. Yöntem basit ve kullanışlı olduğu için ve yöntemle elde edilen yapay kusurlar gaz boşluğu kusurlarına yüksek benzerlik gösterdiği için veri artırımı amacıyla kullanılmıştır. Şekil 3.4'te elipsoit kusurlarının simülasyonu ile elde edilen kusur görüntülerine ait örnekler gösterilmektedir. Tüm bu işlemler uygulanarak toplam 5025 adet X-ışını görüntüsü elde edilmiştir. Şekil 3.5'te veri artırım teknikleri ve simülasyon sonucunda oluşan görüntülerdeki kusur tiplerinin toplam sayıları verilmiştir. Ayrıca Şekil 3.6'da bir görüntü üzerine uygulanan tüm bu teknikler sonucunda elde edilen görüntüler yer almaktadır.



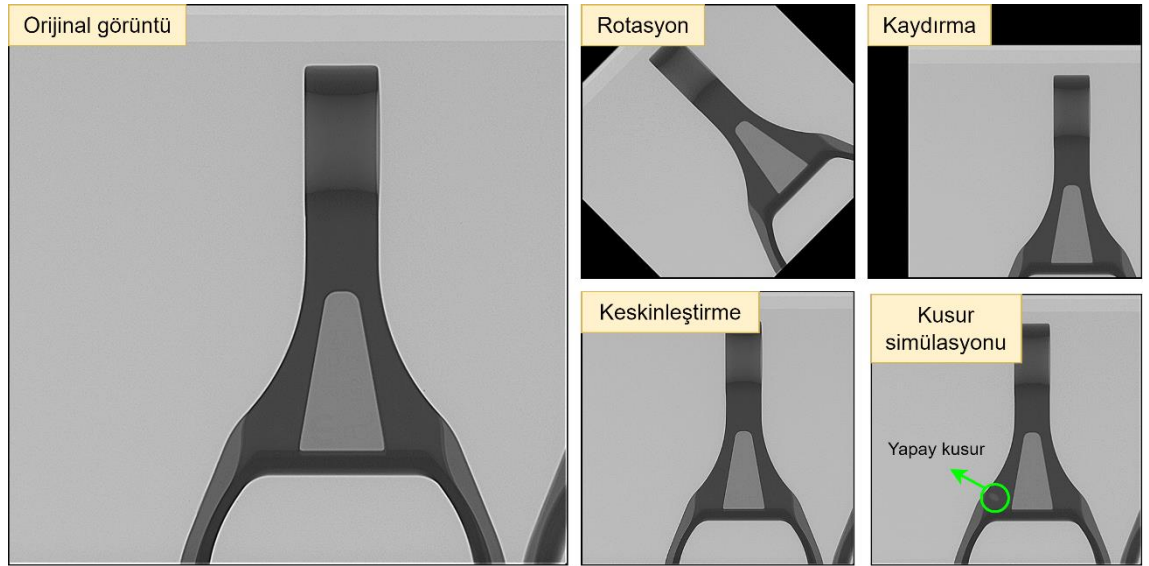
**Şekil 3.4.** Elipsoit kusur simülasyonu ile oluşturulan yapay kusur örnekleri

---

<sup>2</sup> <https://github.com/domingomery/Xvis/blob/master/commands/Xsimdefect.m>



**Şekil 3.5.** Kusur tiplerinin veri setindeki toplam sayıları



**Şekil 3.6.** Veri artırım teknikleri ile oluşturulan görüntüler

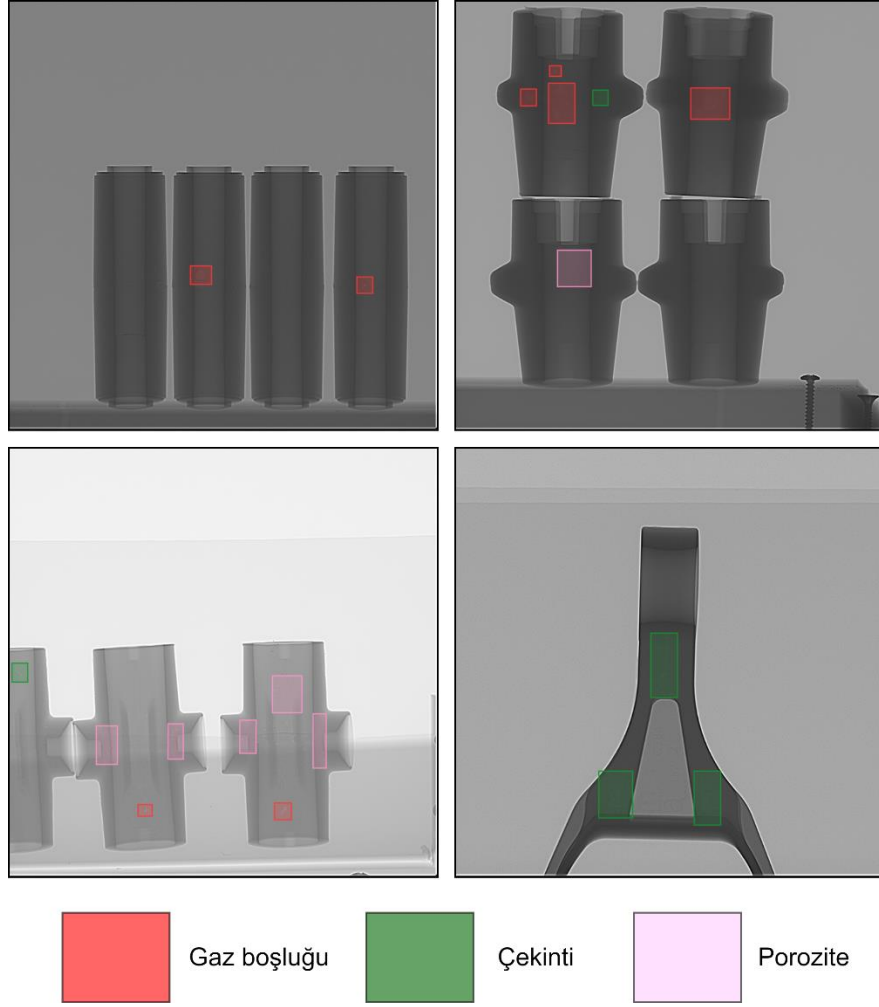
Veri setinde yer alan X-ışını görüntüleri kusur tespiti yönteminin eğitiminde kullanılmak üzere etiketlenmiştir. Etiketleme işlemi, kusurların dijital ortamda bir çerçeve ile sıkıca kapsanması ve bu çerçeveye ait bilgilerin ortaya çıkarılması olarak açıklanabilir. Oluşturulan çerçeveler temel-gerçek (ground-truth) kutusu olarak adlandırılır. Bu işlem ile görüntülerdeki kusurlara ait konumsal ve boyutsal bilgiler ortaya çıkarılmaktadır. Böylece eğitim sırasında yöntemin kusurların gerçek konumlarını kullanarak öğrenmesi gerçekleştirilmektedir. İnsan, araba, kedi, köpek ve ağaç gibi herkes tarafından rahatlıkla

ayırt edilebilen canlılar ve nesnelere kolaylıkla bir görüntü üzerinde etiketlenebilmektedir. Ancak alüminyum dökümlerindeki kusurların etiketlenmesi belirli bir eğitim ve uzmanlık gerektirmektedir. Bu alandaki çalışmaların sayıca fazla olmamasının sebeplerinden biri de budur. Etiketleme için gereken uzmanlık sorunu, dökümhanede muayene işlemini gerçekleştiren mühendislerden alınan bilgiler ve destekler ile çözülmüştür. Böylece 5025 adet görüntünün hepsi bir web uygulaması<sup>3</sup> (Skalski, 2019) ile etiketlenmiştir. Etiketleme işlemine ait görüntü örnekleri Şekil 3.7’de gösterilmektedir. Burada dikkat edilmesi gereken hususlardan biri etiketleme işlemi yaparken oluşturulan kutunun kusurun sınırlarını sıkıca kavrayacak şekilde oluşturulmasıdır.

Etiketleme işlemi oldukça uzun bir süre almakla birlikte görüntülerde yer alan kusurların tamamı tipleriyle birlikte etiketlenerek, tüm görüntüler için *text* dosyaları elde edilmiştir. Eğer görüntü en az bir kusur içeriyorsa bir görüntü için bir adet *text* dosyası oluşturulur. Bu *text* dosyasında yer alan her bir satır, görüntüdeki bir kusura denk gelmektedir. Örneğin bir görüntü üç kusur içeriyorsa dosyada üç satır olacaktır. Her bir satırda beş farklı bilgi yer almaktadır. İlk bilgi sınıfı belirtir ve sıfırdan başlar. Kusur etiketleme işleminde gaz boşluğu için “0” değeri, çekinti için “1” değeri ve “porozite için “2” değeri kullanılmıştır. İkinci ve üçüncü bilgi ise kusurun görüntü üzerindeki koordinatlarını belirtir. Bu koordinatlar etiketlemede kullanılan sınırlayıcı kutunun merkez koordinatlarıdır. Görüntünün orijin noktası sol üst köşedir. Satırda yer alan son iki terim ise boyutlarla ilgilidir. İlk terim kutunun genişliğinin yarısını, ikinci terim de kutu yüksekliğinin yarısını ifade eder. Son olarak etiketlenen bu 5025 adet görüntü sırasıyla %70, %20 ve %10 olacak şekilde eğitim, validasyon ve test gruplarına ayrılmıştır.

---

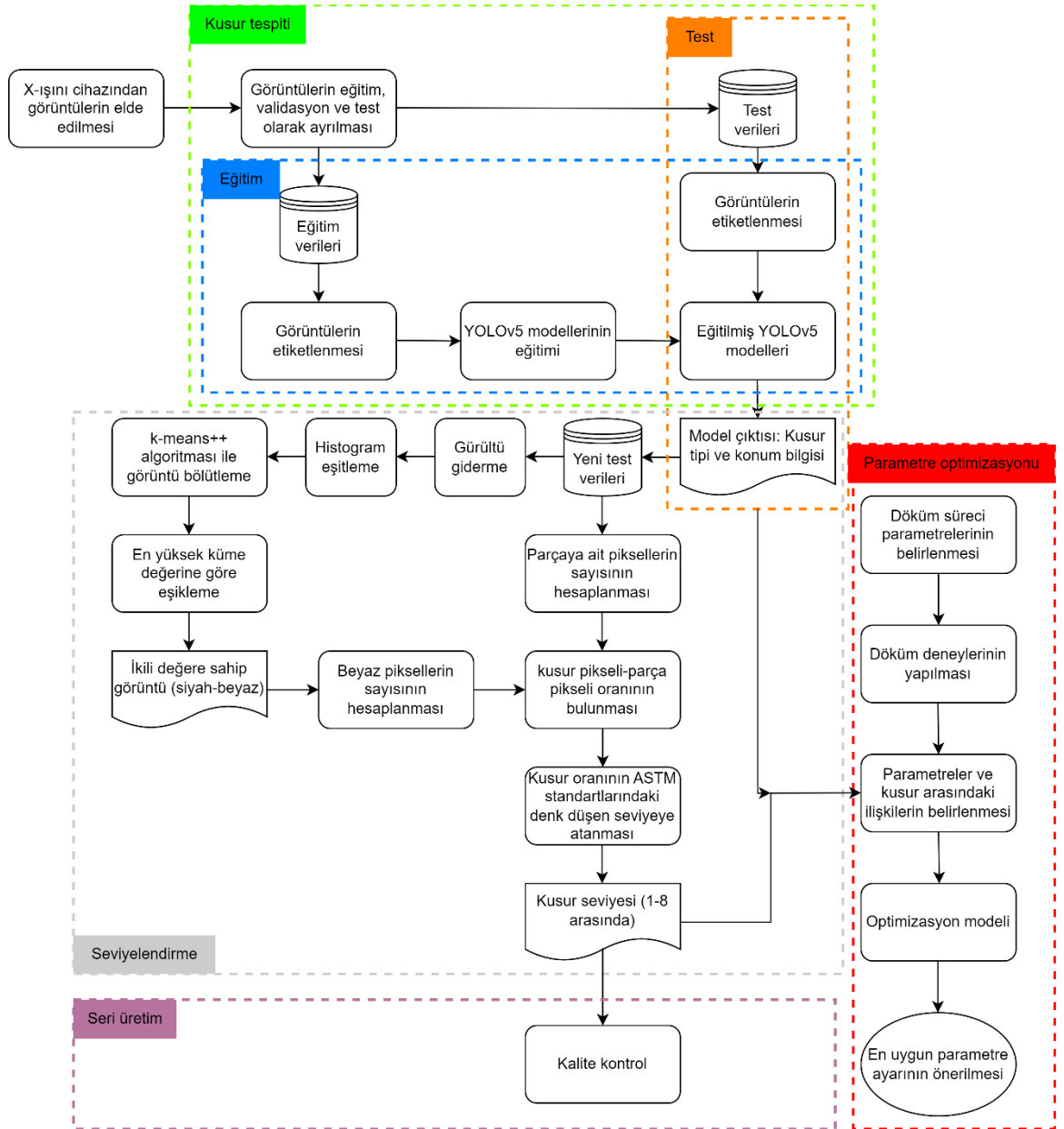
<sup>3</sup> <https://www.makesense.ai/>



**Şekil 3.7.** Etiketleme örnekleri

### 3.2. Yöntem

Bu bölümde çalışmada kullanılan YOLOv5 yönteminin yapısı eski versiyon YOLO yöntemleri de göz önünde bulundurularak detaylıca anlatılmıştır. Daha sonra bölütleme, eşikleme ve seviyelendirme yöntemlerine yer verilmiştir. Ayrıca tüm yöntemlerde kullanılan performans metriklerine ait bilgiler ayrı bir alt başlıkta verilmiştir. Şekil 3.8’de kusur tespiti, seviyelendirmesi ve parametre optimizasyonu için önerilen yöneme ait şema verilmiştir.



**Şekil 3.8.** Önerilen yöntemin şeması

Şekil 3.8’de görüldüğü gibi önerilen yöntemin ilk adımı X-ışını cihazından alüminyum döküm parçalarına ait görüntülerin elde edilmesidir. Elde edilen bu X-ışını görüntüleri çalışmanın ilk büyük aşaması olan kusur tespiti için tek tek uzmanlar tarafından etiketlenir. Etiketlenen bu görüntüler ile ilk olarak derin öğrenme tabanlı nesne tespit yöntemi eğitilir ve daha sonra test edilerek performansı değerlendirilir. Başarılı tespit sonucunda çalışmadaki ikinci büyük aşamaya geçilir. İkinci aşama kusurların tiplerine göre uluslararası bit standardında uygun olarak seviyelendirilmesidir. Bu aşama ilk



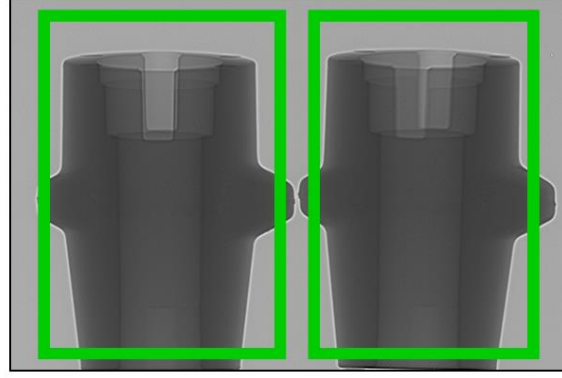
aşamadan elde edilen kusur çerçevelerinin bazı gürültü giderme ve histogram eşitleme gibi görüntü işleme teknikleriyle işlenmesiyle başlar. İşlenen görüntüler otomatik olarak küme sayısının seçildiği bir kümeleme algoritmasıyla eşiklenir. İkili değerlere dönüştürülen görüntüdeki pikseller kullanılarak kusur oranı hesaplanır ve bu değer, ilk orijinal görüntüdeki parçanın bütününe piksel sayısına oranlanır. Bu oran daha önceden belirlenen referans kusur aralığına atanır ve kusur seviyesi belirlenmiş olur. Üçüncü aşamada ikinci aşamadan elde edilen kusur oranı bilgisi kullanılarak farklı döküm süreci parametrelerinin değerlendirilmesini içerir. Bu aşamada amaç kusur oranının en düşük olabileceği döküm süreci parametrelerinin seviyelerinin belirlenmesidir. Böylece önerilen yöntem yalnızca kalite kontrol sürecine değil, aynı zamanda üretim sürecine de katkıda bulunmaktadır. Bu üç aşama birlikte denetim ve kontrol sürecini seri üretime uygun olarak otomatize etme potansiyeli taşımaktadır.

### **3.2.1. Derin öğrenme, konvolüsyonel sinir ağları ve nesne tespiti**

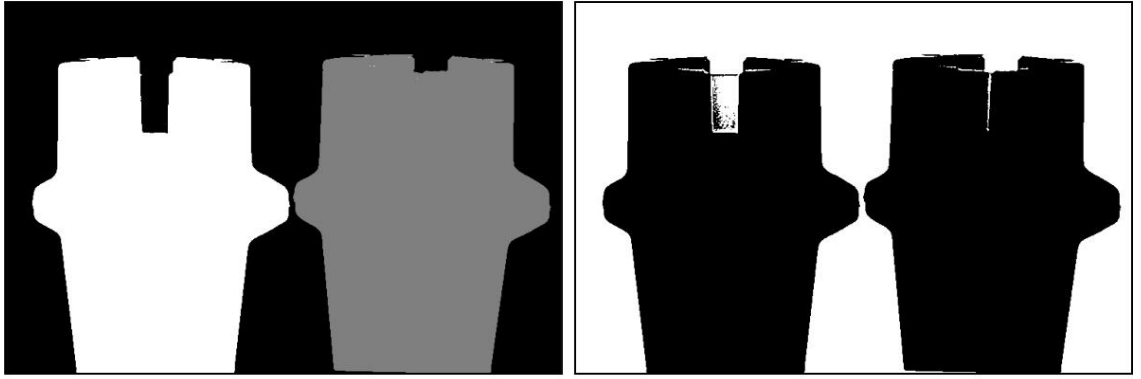
Nesne tespiti, bilgisayarla görme alanındaki en önemli araştırma alanlarından biridir. Semantik bölütleme, sınıflandırma, lokalizasyon ve örnek (instance) bölütleme, nesne tespitinin yanı sıra görüntüler üzerinde çalışan diğer yöntemlerdir. Bilgisayarla görmenin bu görevlerinin aralarında büyük farklılıklar bulunmakla birlikte her yöntem probleme göre seçilir. Bu bilgisayarlı görme yöntemlerinden nesne tespiti, örnek bölütleme ve semantik bölütleme yöntemlerinin aralarındaki farklılıkları özetleyen bir görsel Şekil 3.9'da gösterilmektedir. Sınıflandırma tek bir CNN ile yapılabilen ve yalnızca görüntünün bütünü hakkında yapılan bir bilgisayarlı görme işleviyken lokalizasyon görüntüdeki nesnenin yerini de bulmak için kullanılabilir. Ancak bu yöntemler bir görüntüdeki tek bir nesneye odaklanmaktadırlar. Nesne tespiti ise bir görüntü içinde birden fazla nesneyi (farklı sınıflar da dahil) bulup sınıflandırmak için kullanılmaktadır (Şekil 3.9a). Örnek bölütleme aynı işlemi arka plandan ayırarak piksel temelinde yapmaktadır ve aynı sınıfa ait olan nesnelere bile farklı renklerde ayrıştırılmaktadır (Şekil 3.9b). Semantik bölütlemeye ise farklı sınıflara ait nesnelere arka plandan ayrılma işlemi tıpkı örnek bölütlemeye gibi olmasına rağmen aynı sınıfa ait nesnelere aynı renkle gösterilmektedir (Şekil 3.9c).

Nesne tespit alanında uzun bir süredir birçok araştırma ortaya konmuştur. Son yıllarda Convnet öznetelik çıkarıcıların yükselişi (Liu ve diğerleri, 2022) ve önceki bilgileri aktarma yöntemi olarak kullanılan Transfer Öğrenmesi nedeniyle önemli sonuçlar elde edilmiştir. İlk nesne dedektörleri, genellikle kayan pencereler (Dalal ve Triggs, 2005) gibi tekniklerden faydalanmışlardır. Bilgisayarla görme bağlamında kayan pencere, bir görüntü boyunca "kayan" sabit genişlik ve yükseklikte bir dikdörtgendir. Dikdörtgenin geçtiği her bölgede nesne olup olmadığı kontrol edilir. Bu yöntemde kayan pencerenin üzerine geldiği her bir bölge ayrı ayrı öznetelik çıkarıcıdan geçirilir. Özellikle yüksek çözünürlüğe sahip bir görüntü üzerinden küçük nesnelere tespit etme durumu söz konusu olduğunda bu işlem oldukça uzun bir zaman alır.

Modern nesne tespit yöntemleri ise kayan pencere gibi erken dönem teknikler yerine çok daha verimli yöntemleri kullanırlar. Modern derin öğrenme tabanlı nesne tespit yöntemleri genelde tek aşamalı ve iki aşamalı olmak üzere iki grupta incelenebilir. İki aşamalı dedektörler arasında en iyi bilinen yöntemlerden biri R-CNN'nin gelişmiş bir versiyonu olan Daha Hızlı R-CNN'dir. Bu tip yöntemler yüksek doğrulukta tespit yaparken tespit süresi açısından tek aşamalıya göre daha yavaştır. Bunun yanı sıra tek aşamalı dedektörler arasında en yaygın olarak kullanılan yöntemler arasında YOLO'nun farklı sürümleri yer almaktadır. İlk sürümler oldukça hızlı tespit yaparken, doğru tespit açısından iki aşamalı yöntemlere göre birçok uygulamada daha düşük performans gösterirler. Ancak son sürüm tek aşamalı yöntemler hızlı tespitin yanı sıra oldukça yüksek doğrulukta tespit görevini de yapmaktadırlar. Özellikle YOLO yöntemleri ile birçok uygulamada başarılı sonuçlar elde edilmiştir. Kusur tespiti göz önünde bulundurulduğunda YOLO'nun farklı versiyonları karşılaştırmalı sonuçlarda diğerlerine göre üstün başarı göstermektedirler (Parlak ve Emel, 2023); Duan ve diğerleri, 2021; Mery, 2021).



(a)

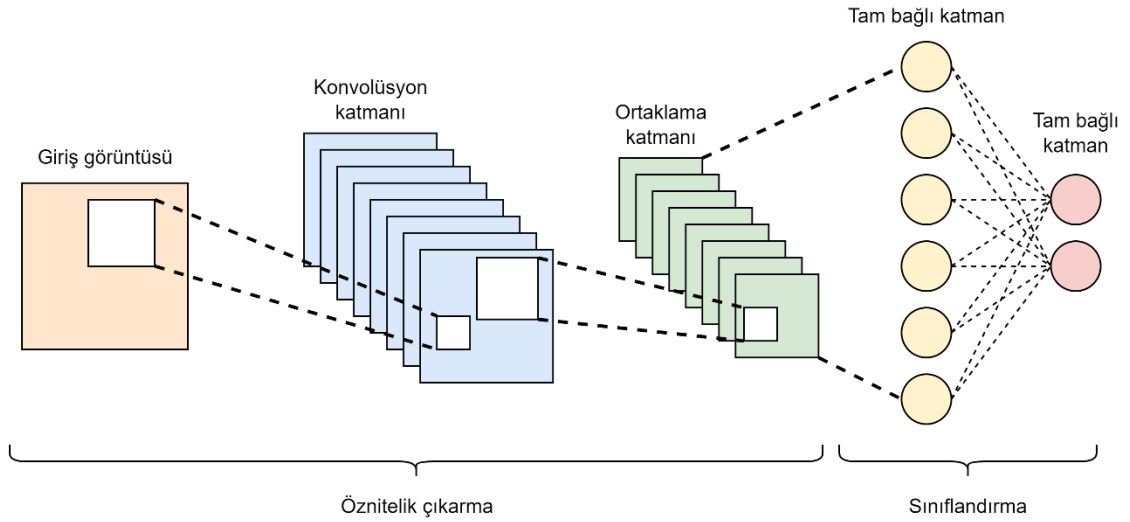


(b)

(c)

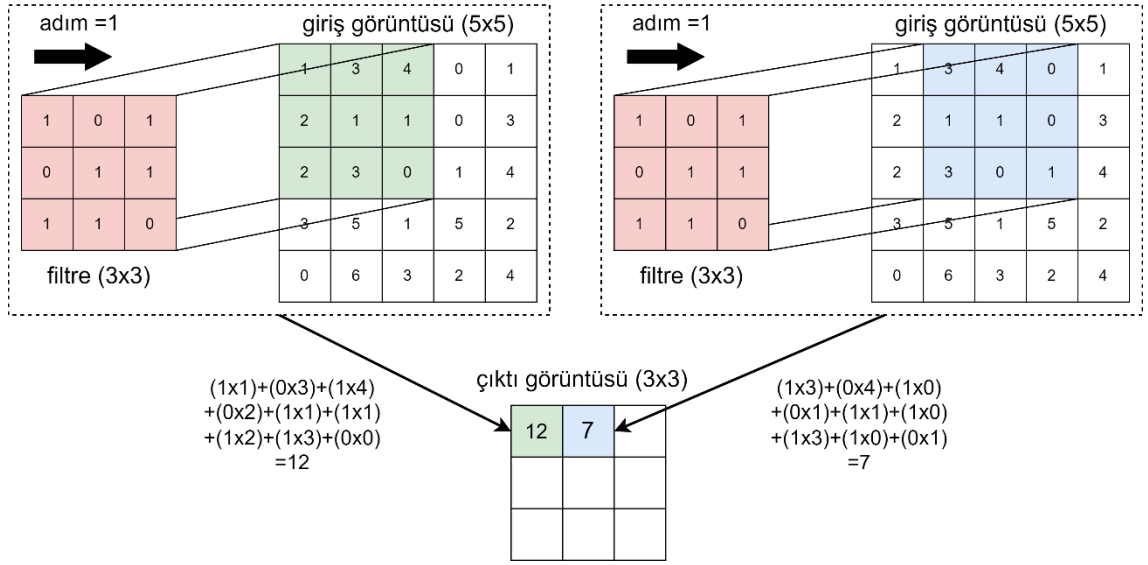
**Şekil 3.9.** Farklı bilgisayarlı görme görevleri; (a) nesne tespiti, (b) örnek bölütleme ve (c) semantik bölütleme

Nesne tespit yöntemlerine girmeden önce CNN'lerin bazı temel yapılarından bahsetmek gerekir. CNN, derin öğrenmenin en temsili modelidir (LeCun ve diğerleri, 2015). Bir CNN tipik olarak üç katmana sahiptir: Bir konvolüsyon katmanı, bir ortaklama katmanı ve tam bağlı bir katman (Şekil 3.10). CNN'nin her katmanı bir özellik haritası olarak bilinir. Giriş katmanının özellik haritası, farklı renk kanalları (ör. RGB) için piksel yoğunluklarından oluşan 3B bir matristir ve bu tip matrisler tensör olarak adlandırılır.



**Şekil 3.10.** Basit bir CNN yapısının görünümü

Konvolüsyon katmanı, CNN'nin temel yapı taşı olarak bilinmektedir. Ağın hesaplama yükü genel olarak bu katmana aittir. Bu katmanda iki matris arasında bir iç çarpım gerçekleştirilir. Bu matrislerden biri kernel ya da filtre olarak adlandırılabilir ve aynı zamanda ağırlık değerlerini içeren bir matristir. Filtre, bir görüntüden uzamsal anlamda daha küçük ancak daha derinlemesindedir. Bu ifade, görüntü üç kanallı ise, filtre yüksekliği ve genişliğinin uzamsal olarak küçük olacağı, ancak derinliğin üç kanalın tümüne kadar uzanacağı anlamına gelir. İleri geçiş (forward pass) sırasında filtre, görüntünün yüksekliği ve genişliği boyunca kayar ve bu, alıcı bölgenin (receptive field) görüntü temsiliyi üretir. Çekirdeğin tanımlanan bir ölçü kadar hareket etmesine adım (stride) denir. Bir girdi görüntüsü üzerinden filtrenin bir adım hareket etmesi Şekil 3.11'de gösterilmiştir. Şekil 3.11'de filtre hareketi sonucunda oluşan çıktı görüntüsü öznitelik haritası olarak adlandırılır. Şekil 3.11'de yer alan örnek gösterim tek kanallı bir görüntü için geçerlidir.



**Şekil 3.11.** Giriş görüntüsü üzerinde filtrenin bir adım ilerlemesi

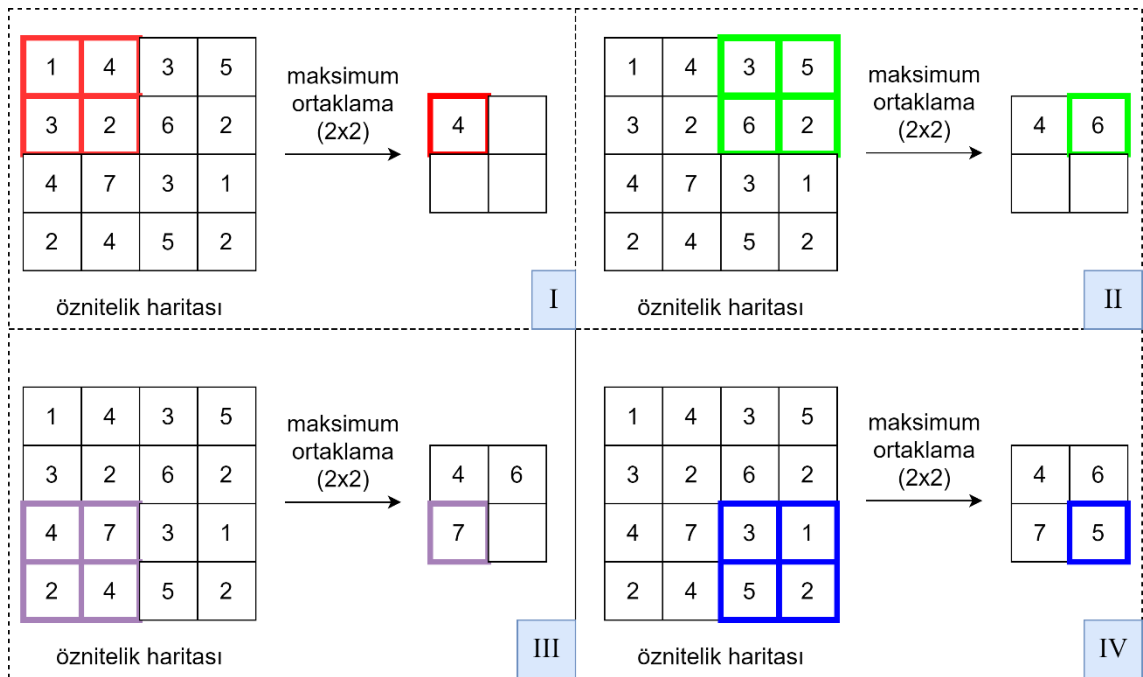
$S$  adım büyüklüğü,  $N \times N$  girdi görüntüsünün ve  $F \times F$  girdi görüntüsüne uygulanan filtrenin boyutu olmak üzere çıktı görüntüsünün boyutu  $O \times O$  Denklem (3.1) kullanılarak hesaplanır.

$$O \times O = \left(1 + \frac{N-F}{S}\right) \times \left(1 + \frac{N-F}{S}\right) \quad (3.1)$$

Konvolüsyon işleminden sonra girdi ile çıktı görüntüsü arasındaki boyut farkını yönetmek için piksel ekleme (padding) uygulanır. Girdi görüntüsünde ilk satırın üstüne ve ilk sütunun soluna  $\left\lfloor \frac{F-1}{2} \right\rfloor$ , son satırın altına ve son sütunun sağına  $\left\lfloor \frac{F}{2} \right\rfloor$  kadarlık bir piksel ekleme yaparsak girdi görüntüsüne uygulanan konvolüsyon işlemi sonucunda elde edilen öznitelik haritasının boyutu ile girdi görüntüsünün boyutu aynı olur (J. Wu, 2017). Eklenen bu piksellerin değeri orijinal tensörün özelliklerini değiştirmemek için sıfır olarak ayarlanır.

Yukarıda açıklanan işlemler ile elde edilen öznitelik haritaları bir sonraki adımda ortaklama/havuzlama (pooling) katmanında işlem görür. Bu katmanda genellikle maksimum ortaklama (max pooling) yöntemi kullanılır. Maksimum ortaklama dışında kullanılan diğer bir yaygın yöntem ise ortalama ortaklık yöntemidir. Ağın bu katmanında

öğrenilen herhangi bir parametre yoktur. Öznitelik haritasının yükseklik ve genişliğini azaltır. Hesaplama karmaşıklığını azaltmak için kullanılan bir adımdır. Maksimum ortaklamada öznitelik haritasındaki ilgili alanın maksimum piksel değeri seçilir. Şekil 3.12’de maksimum ortaklama örneği verilmiştir. Şekil 3.12’de yer alan öznitelik haritasına  $2 \times 2$ ’lik bir maksimum ortaklama işlemi uygulanmıştır ve bu işlem sonucunda daha küçük boyuta sahip bir matris elde edilmiştir. Burada  $S = 2$  seçildiği için  $4 \times 4$  boyutuna sahip matris,  $2 \times 2$  boyutuna sahip bir başka matrise dönüşmüştür.



**Şekil 3.12.** Maksimum ortaklama örneği

Şekil 3.10’da yer alan CNN’de ortaklama katmanından sonra tam bağlı katman gelmektedir. Bu katmandaki nöronlar, bir önceki katman ve bir sonraki katmandaki tüm nöronlarla tam bağlantıya sahiptir. Tam bağlı katman, girdi ve çıktı arasındaki temsili haritalamaya yardımcı olur. Tam bağlı katmanlar genellikle CNN’in sonunda bulunur. Tam bağlı katmanlar sayesinde sınıf skorları elde edilir.

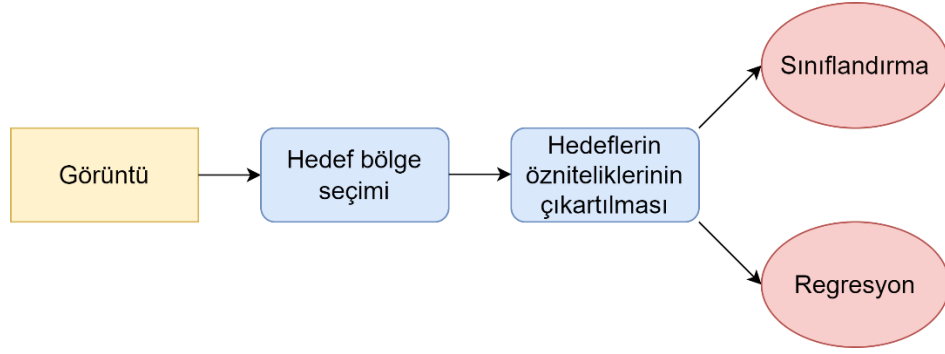
Genellikle, tüm öznitelikler tam bağlı katmana bağlandığında, eğitim veri setinde ezberleme (overfitting) durumuna neden olabilir. Bir model eğitim seti üzerinde başarılı bir performans göstermesine rağmen yeni karşılaştığı test verileri üzerinde kötü bir performans ortaya koyuyorsa ezberleme durumu söz konusu olabilir (Li ve diğerleri,

2019). Bu sorunun üstesinden gelmek için, eğitim süreci sırasında sinir ağından birkaç nöronun düştüğü ve modelin boyutunun küçüldüğü bir bırakma (dropout) katmanı kullanılır. Bırakma, ağı daha basit hale getirerek fazla ezberlemeyi önlediği için modelin performansında artış gözlemlenebilir. Yöntem eğitim esnasında tam bağlı katmandaki nöronların belirli bir kısmını rastgele seçip ağdan çıkarma şeklinde uygulanır (Srivastava ve diğerleri, 2014).

Son olarak, CNN modelinin en önemli kavramlarından biri aktivasyon fonksiyonudur. Ağın değişkenleri arasındaki her türlü sürekli ve karmaşık ilişkiyi öğrenmek ve tahmin etmek için kullanılırlar. Basit bir deyişle, modelin hangi bilgilerinin ileri yönde tetiklenmesi ve hangilerinin ağın sonunda olmaması gerektiğine karar verir. Aktivasyon fonksiyonları, ağa doğrusal olmama özelliği ekler. ReLU, softmax, tanh ve sigmoid fonksiyonları, aktivasyon fonksiyonu olarak yaygın bir şekilde kullanılmaktadır. İkili sınıflandırma söz konusu olduğunda sigmoid fonksiyonları yaygın tercih edilirken, çok sınıflı modellerde softmax yaygın olarak kullanılır. Tanh fonksiyonu sigmoid fonksiyonuna çok benzerdir. Tek farkı orijine göre simetrik olmasıdır. ReLU fonksiyonunu diğer aktivasyon fonksiyonlarına göre kullanmanın ana avantajı, tüm nöronları aynı anda aktive etmemesidir. Bir CNN modelindeki aktivasyon fonksiyonları, bir nöronun aktive edilip edilmeyeceğini belirler. Matematiksel işlemleri kullanarak tahminde bulunmanın önemli olup olmadığına karar verirler (Sharma ve diğerleri, 2020).

Son on yılda önemli bazı CNN mimarileri sunulmuştur (Dhillon ve Verma, 2020; Khan ve diğerleri, 2020). Özellikle CNN mimarilerindeki en yeni gelişmeler ağ derinliğinin kullanımı konusunda gerçekleştirilmiştir. İlk CNN modeli örneği LeNet (LeCun ve diğerleri, 1995) olarak bilinmektedir (Khan ve diğerleri, 2020). Bu modelden sonra yüzlerce farklı yapıda CNN modelleri geliştirilmiştir. En iyi bilinenlerinden bazıları AlexNet (Krizhevsky ve diğerleri, 2012), VGG (Simonyan ve Zisserman, 2015), GoogLeNet (Szegedy ve diğerleri, 2015), Inception-V3 (Szegedy ve diğerleri, 2016), Inception-V4 (Szegedy ve diğerleri, 2017), ResNet (He ve diğerleri, 2016) ve DenseNet (Huang ve diğerleri, 2017) olarak sayılabilir. CNN'ler bir görüntüye ait bilgileri çıkarma konusunda oldukça başarılıdır ancak tek başlarına görüntülerdeki nesnelere tespit etmek için yetersiz kalmaktadırlar. Tespit görevi yalnızca nesnenin görüntü üzerinde var olup olmadığıyla değil aynı zamanda görüntü üzerinde nerede olduğunu da içerir. Bununla birlikte tespit görevi, sınıflandırma ve lokalizasyondan farklı olarak bir görüntü

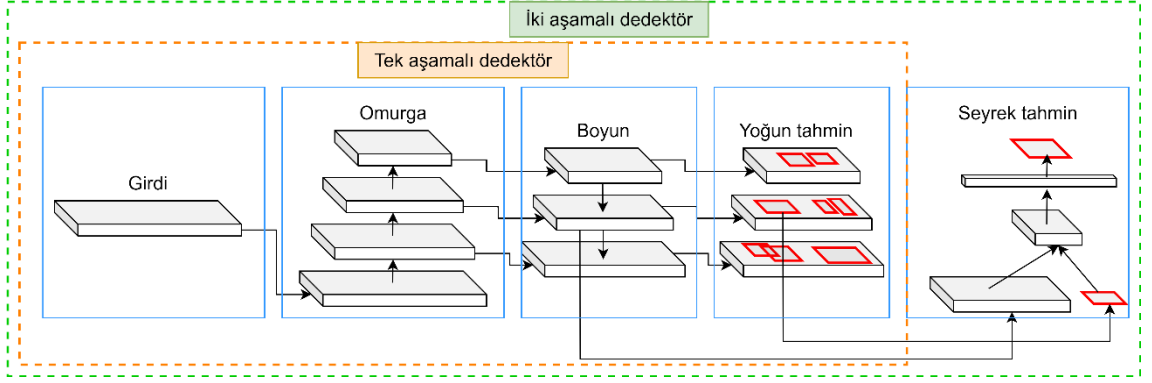
üzerindeki birden fazla nesneyi sınıflarına göre ayırt edip konumlandırmayı da içerir. Şekil 3.13'te basit bir nesne tespit yöntemine ait şematik gösterim verilmiştir.



**Şekil 3.13.** Genel bir nesne tespit algoritmasının gösterimi

Modern bir nesne tespit yöntemi ya da farklı bir deyişle nesne dedektörü genellikle ImageNet ile önceden eğitilmiş bir “omurga” ve sınıflarla birlikte nesne sınırlayıcı kutularını tahmin etmek için kullanılan bir “baş” olmak üzere iki bölümden oluşur (Bochkovskiy ve diğerleri, 2020b). Omurga kısmı bir CNN’den oluşur ve öz nitelikleri çıkarmak için kullanılır. GPU platformunda çalışan dedektörler için omurgalar VGG, ResNet (He ve diğerleri, 2016) veya DenseNet (Huang ve diğerleri, 2017) olabilirken, CPU platformunda çalışan dedektörler için omurgalar SqueezeNet (Iandola ve diğerleri, 2016), MobileNet (Howard ve diğerleri, 2019, 2017; Sandler ve diğerleri, 2018; Tan ve diğerleri, 2019) veya ShuffleNet (Ma ve diğerleri, 2018; X. Zhang ve diğerleri, 2018) olabilir. Baş kısmı genellikle tek aşamalı nesne dedektörü ve iki aşamalı nesne dedektörü olmak üzere iki türe ayrılır. Bu kısım sınıflandırma ve regresyondan sorumlu tahmin modülüdür. Son yıllarda geliştirilen nesne dedektörleri genellikle omurga ve baş arasında bazı katmanlar yerleştirir. Bu katmanlar genellikle farklı aşamalardan öz nitelik haritaları toplamak için kullanılır. Bu bölüm “boyun” olarak adlandırılabilir (Bochkovskiy ve diğerleri, 2020a). Şekil 3.14’te temsili bir nesne dedektörünün mimarisi hem tek aşamalı hem de iki aşamalı olarak gösterilmiştir. Genellikle bir boyun birkaç aşağıdan yukarıya ve yukarıdan aşağıya yoldan oluşur (Şekil 3.14). Bu mekanizma ile donatılmış ağlar arasında Öz nitelik Piramit Ağı (FPN) (Lin ve diğerleri, 2017a), Yol Toplama Ağı (PAN) (Liu ve diğerleri, 2018), BiFPN (Tan ve diğerleri, 2020a) ve NAS-FPN (Ghiasi ve diğerleri, 2019) bulunur.





Şekil 3.14. Nesne dedektörlerinin mimarisi (Bochkovskiy ve diğerleri, 2020b)

### 3.2.2. Modern nesne tespit yöntemleri ve YOLO

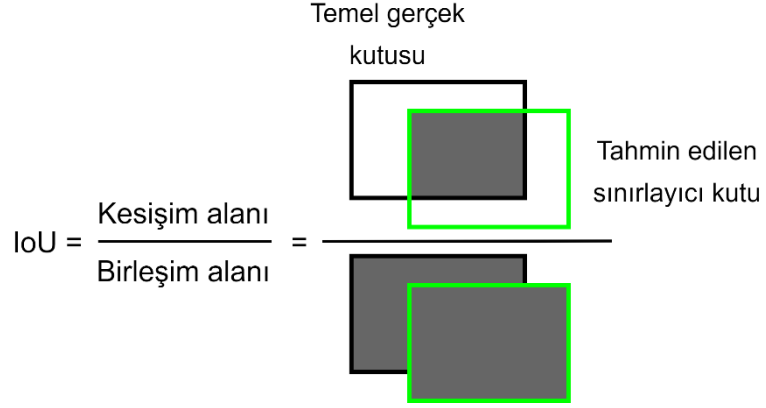
Bu kısımda ilk olarak R-CNN'e yer verilmiştir. R-CNN ve sonrasında geliştirilen Hızlı R-CNN ve Daha Hızlı R-NN yöntemleri açıklanarak çalışmada kullanılan YOLO'nun anlaşılabilir olması hedeflenmiştir. Gerekli bazı yöntemler anlatıldıktan sonra YOLO'nun beş farklı versiyonu kronolojik sırayla incelenmiştir. Tüm bunlardan önce nesne tespit yöntemlerinin bazı temel kavramları ele alınmıştır.

**Sınırlayıcı kutu:** İlgilenilen nesneyi çevrelemek için kullanılan sıkı bir dikdörtgendir. Genellikle dört değer kullanılarak tanımlanır:  $(b_x, b_y, b_h, b_w)$ .  $(b_x, b_y)$  sınırlayıcı kutunun merkez koordinatları ve  $b_h, b_w$  sırasıyla 0-1 ölçeğinde ölçülen sınırlayıcı kutunun yüksekliği ve genişliğidir.

**Bağlantı (anchor) kutusu:** Bunlar, belirli bir yükseklik ve genişlikte önceden tanımlanmış bir dizi sınırlayıcı kutudur. Bu kutular, tespit edilmesi istenen belirli nesne sınıflarının ölçeğini ve en-boy oranını yakalamak için tanımlanır ve genellikle eğitim verileri kümesindeki nesne boyutlarına göre seçilir. Tespit sırasında, önceden tanımlanmış bağlantı kutuları görüntü boyunca döşenir. Ağ, döşenen her bağlantı kutusu için olasılığı ve arka plan, birleşim üzerinden kesişim ( $IoU$ ) ve uzantılar (offset) gibi diğer nitelikleri tahmin eder. Tahminler, her bir bağlantı kutusunu hassas bir şekilde ayarlamak için kullanılır.

**Temel gerçek kutusu:** Eğitim sırasında modelin nesneye ait konum ve boyut bilgilerini kullanabilmesi için önceden elle tanımlanan sıkı dikdörtgenlerdir.

**Birleşim üzerinden kesişim (IoU):** Temel-gerçek kutusuna göre tahmin edilen sınırlayıcı kutunun doğruluğunu kontrol etmek amacıyla kullanılan bir değerlendirme metriğidir (Şekil 3.15). *IoU* 0-1 arasında bir değer alır ve 1'e yakın olması iyi bir tespit anlamına gelmektedir.



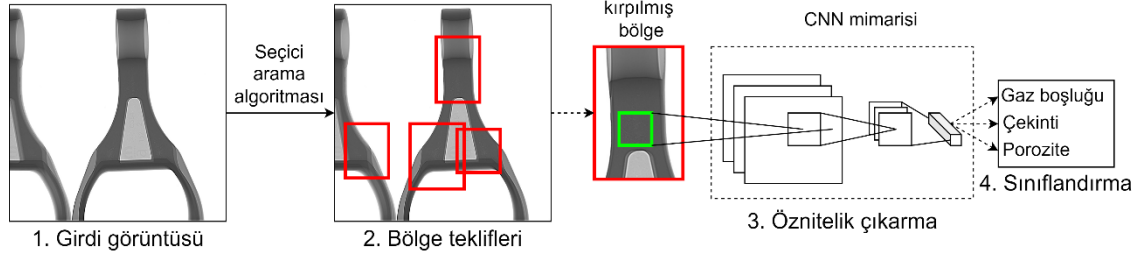
**Şekil 3.15.** Birleşim üzerinden kesişim (*IoU*)

**Maksimum olmayan bastırma (non-max suppression):** Belirli bir nesne için birden fazla sınırlayıcı kutu tahmin edildiyse bu teknik, maksimum *IoU* değerine sahip olan sınırlayıcı kutu dışındaki tüm kutuları ortadan kaldırır.

Bu kavramlar tanımlandıktan sonra bazı önemli nesne dedektörleri aşağıda açıklanmıştır.

**R-CNN:** İlk derin öğrenme tabanlı nesne tespit modellerinden biri olan R-CNN nesne tespiti alanında önemli bir adımdır (Girshick ve diğerleri, 2014). R-CNN, bölge tabanlı bir CNN dedektörüdür. R-CNN dedektörü dört modülden oluşur. İlk modül sınıflardan bağımsız bölge önerileri üretir. İkinci modül, her bölge teklifinden sabit uzunlukta bir öznitelik vektörü çıkarır. Üçüncü modül, görüntüdeki nesnelere sınıflandırmak için kullanılan bir SVM'dir. Son modül, sınırlayıcı kutu tahmini için kullanılan bir regresördür. Bölge önerileri oluşturmak için seçici arama yöntemi kullanılmaktadır. Bu modülde yaklaşık olarak 2000 adet bölge önerisi yapılır. Daha sonra, her bölge teklifinden 4096 boyutunda bir öznitelik vektörünü çıkarmak için bir CNN kullanılır. Tam bağlı katman, sabit uzunlukta giriş vektörlerine ihtiyaç duyduğundan, bölge önerisi öznitelikleri aynı boyuta sahip olmalıdır. Yazarlar, CNN'nin giriş boyutunu sabit ve  $227 \times 227$  olarak belirlemişlerdir. CNN öncesinden gelen farklı boyuttaki bölge

tekliflerinin çözünürlüğü  $227 \times 227$  olarak ayarlanır. Öznitelik çıkarma ağı, beş konvolüsyon katmanından ve iki tam bağlı katmandan oluşur. R-CNN'in en büyük dezavantajı, ilgili öznitelikleri çıkarma görevinde önerilen bölgelerin miktarının çok olması nedeniyle yavaş olmasıdır. Tüm bu bölgeler, binlerce ileri beslemeli hesaplama ve büyük bir bilgi işlem yükü gerektiren CNN ağından geçirilir ve bu da onu gerçek zamanlı nesne algılama uygulamaları için olanaksız hale getirir (Jiao ve diğerleri, 2019).

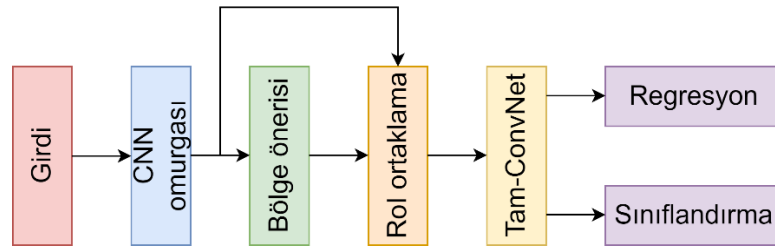


**Şekil 3.16.** R-CNN mimarisi (Girshick ve diğerleri, 2014)

**Hızlı R-CNN:** Tespit görevini R-CNN'den daha hızlı bir şekilde gerçekleştirir (Girshick, 2015). R-CNN, hesaplamayı paylaşmadan her bölge teklifi için bir ConvNet ileri geçişi gerçekleştirdiğinden, SVM ile sınıflandırma uzun zaman alır. Hızlı R-CNN, tüm bir giriş görüntüsünden öznitelikler çıkarır. Daha sonra sınıflandırma ve sınırlayıcı kutu regresyonu öncesindeki tam bağlı katmanların girdisi olacak sabit boyutlu öznitelikleri elde etmek için önceki öznitelikler, ilgi bölgeleri (RoI) adı verilen bir ortaklama katmanından geçer. Öznitelikler tüm görüntüden bir kez çıkarılır ve bir seferde sınıflandırma ve yerleştirme için CNN'e gönderilir. Diğer bir gelişme, Hızlı R-CNN'nin farklı büyüklükteki bölge tekliflerinden sabit boyutlu bir öznitelik haritası çıkarmak için bir RoI ortaklama katmanı kullanmasıdır. Önerilen bölgeleri belirli bir çözünürlüğe sabitlemeye gerek duymadan bu işlem, bölge önerilerinin özniteliklerinin konumsal bilgilerini saklı tutar. PASCAL VOC 2007 veri setinde Hızlı R-CNN'nin %66,9 *mAP*, R-CNN'in ise %66,0 *mAP* performansı göstermiştir. Eğitim süresi, R-CNN'de 84 saat ve Hızlı R-CNN'de 9,5 saattir. Ayrıca test süresinde Hızlı R-CNN 213 kat daha hızlıdır (Jiao ve diğerleri, 2019).

**Daha Hızlı R-CNN:** Hızlı R-CNN'nin önerilmesinden üç ay sonra, Daha Hızlı R-CNN (Ren ve diğerleri, 2017) bölge bazlı CNN yapısını daha da iyileştirir. Hızlı R-CNN, yavaş

olan ve tespit ağıyla aynı çalışma süresine ihtiyaç duyan RoI'yi önermek için seçici aramayı kullanır. Daha hızlı R-CNN, çok çeşitli ölçekler ve en-boy oranları ile bölge önerilerini verimli bir şekilde tahmin etmek için tamamen konvolüsyonel bir ağ olan yeni bir Bölge Öneri Ağı (RPN) ile değiştirir. RPN, tespitten görevli ağ ile görüntünün tamamının konvolüsyonel özneteliklerini ve ortak bir konvolüsyon katmanları kümesini paylaştığı için bölge önerileri oluşturma hızını artırır. Ayrıca, farklı boyutlardaki nesnelere tespiti için yeni bir yöntem önerilmiştir. Bu yöntem referans olarak çok ölçekli bağlantı (anchor) kutularının kullanılmasındadır. Bağlantı kutuları çeşitli büyüklükteki bölge önerileri oluşturma sürecini büyük ölçüde basitleştirebilir. Yazarlar, bağlantı kutularının üç farklı ölçek ve üç farklı en-boy oranına sahip olduğunu tanımlamışlardır. Bölge önerisi, bir referans bağlantı kutusuna göre parametrelendirilir. Ardından, tahmin edilen kutunun konumunu optimize etmek için tahmin edilen kutu ile buna karşılık gelen temel-gerçek kutusu arasındaki mesafe ölçülür. PASCAL VOC 2007 test setinde, Daha Hızlı R-CNN, %66,9'luk Hızlı R-CNN'ye kıyasla %69,9'luk *mAP* değerine ulaşmıştır. Ayrıca, Daha Hızlı R-CNN'nin (198 ms) toplam çalışma süresi, aynı VGG omurgasına sahip Hızlı R-CNN'den (1830 ms) yaklaşık 10 kat daha düşüktür. Daha hızlı R-CNN'de saniye başına kare (FPS) değeri 5,0 olarak elde edilirken, Hızlı R-CNN'de bu değer 0,5 olarak elde edilmiştir (Ren ve diğerleri, 2017). Şekil 3.17'de Daha Hızlı R-CNN mimarisinin basit bir gösterimi verilmiştir.



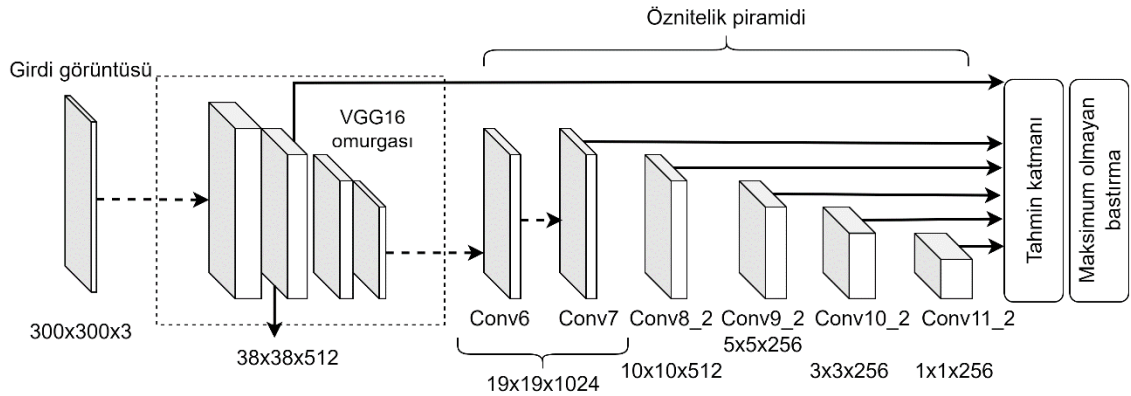
**Şekil 3.17.** Daha Hızlı R-CNN mimarisi (Ren ve diğerleri, 2017)

Yukarıda anlatılan üç yöntem iki aşamalı dedektörlere ait en iyi bilinen yöntemlerden bazılarıdır. Bundan sonraki anlatılacak yöntemlerin tamamı tek aşamalıdır. İlk olarak bazı modern nesne tespit yöntemleri kısaca anlatılıp, çalışmada tespit için kullanılan YOLO ve farklı versiyonlarına daha detaylı bir şekilde yer verilmiştir.

**SSD:** Nesne lokalizasyon ve sınıflandırma görevleri, ağın tek bir ileri geçişinde yapılır. DeepMultiBox (Erhan ve diğerleri, 2014), sınırlayıcı kutu regresyonu için kullanılan

teknikinin adıdır. Ağ, tespit edilen nesnelere de sınıflandıran bir nesne dedektörüdür. Şekil 3.18’de SSD dedektörünün mimarisi verilmiştir. Ayrıca Şekil 3.18’de yer alan VGG1-6 ağına ait mimari Şekil 3.19’da yer almaktadır. Burada kullanılan CNN ağları farklılıklar gösterebilmektedir. SSD’deki ana yaklaşım, sabit boyutlu bir sınırlayıcı kutu koleksiyonu üreten ve bu kutularda nesne sınıfı örneklerinin varlığı için puanlar üreten ileri beslemeli bir konvolüsyonel ağa dayalıdır. Bu ağın ardından nihai tespitleri üretmek için maksimum olmayan bastırma adımı gelir.

**EfficientDet:** EfficientDet, hafif, ölçeklenebilir bir tespit ağıdır ve D0–D7 olmak üzere toplam sekiz farklı model içerir (Tan ve diğerleri, 2020b). D0’dan D7’ye, model boyutuyla birlikte modelin doğruluğu ve zaman karmaşıklığı artar. Sekiz model, geniş bir kaynak kısıtı yelpazesini karşılayabilir. EfficientDet’in omurgası, modeli daha hafif hale getirmek için EfficientNet (Tan ve Le, 2019) ağını kullanır. Ayrıca, ağın boyun kısmı çift yönlü ağırlıklı özellik piramit ağı (BiFPN) kullanır. Yol toplama ağından (PANet) (Liu ve diğerleri, 2018) başlayarak BiFPN, yalnızca bir giriş kenarına sahip düğümleri kaldırır. Giriş ve çıkış düğümleri aynı seviyedeysen, bir kısayol ekler. Amaç, hesaplama maliyetini artırmadan daha fazla özneteliği birleştirmektir.

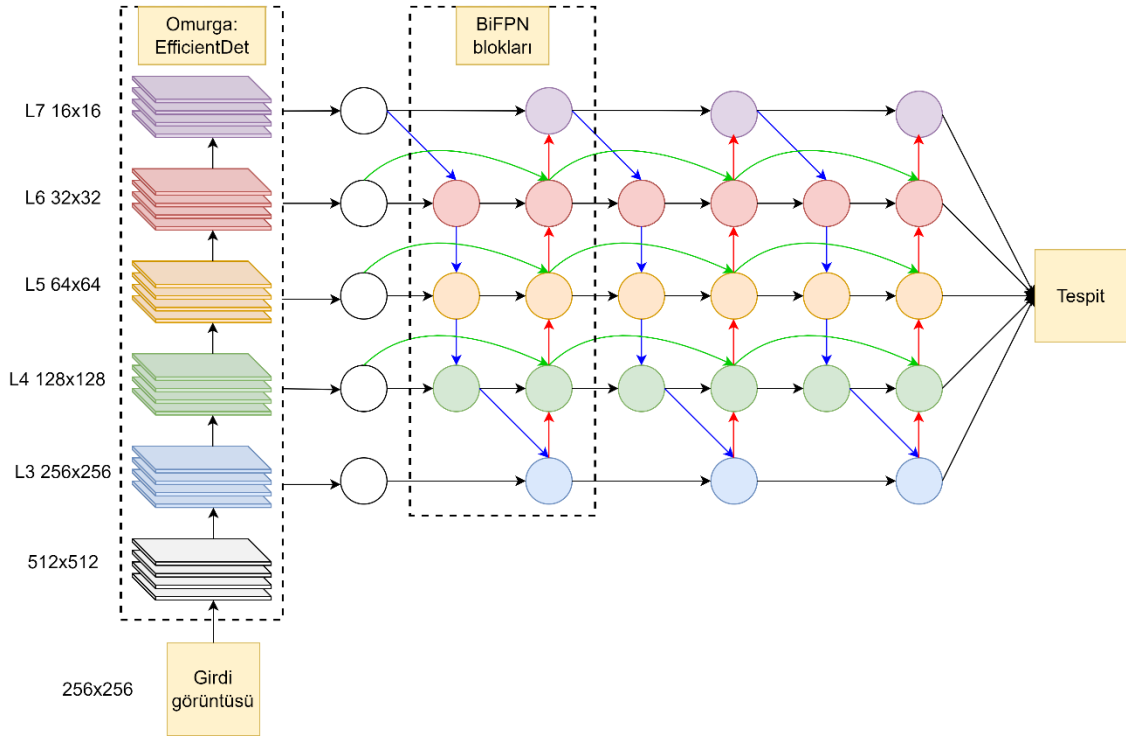


Şekil 3.18. SSD mimarisi (Liu ve diğerleri, 2016b)



Şekil 3.19. VGG16 ağının katmanları (Simonyan ve Zisserman, 2015)

**RetinaNet:** Bu yöntem hızlı tespit süresini korurken mevcut iki aşamalı dedektörlerle tespit doğruluğu boşluğunu azaltmak için tek aşamalı bir nesne dedektörü olarak önerilmiştir. Lin ve diğerleri (2020) tarafından geliştirilen bu modelin yer aldığı çalışmada tek aşamalı dedektörler ve iki aşamalı dedektörler arasındaki doğruluk farkının temel olarak, eğitimde kullanılan kolay örnekler ve zor örneklerin yanı sıra olumlu ve olumsuz örneklerin sayısından kaynaklandığı belirtilmiştir. Çok sayıda kolay örnek, yozlaşmış bir modelle sonuçlanan kayıp fonksiyonuna neden olmuştur. Bu problem, kolay örneklerin ağırlıklarını uyarlamalı olarak azaltmak için odak kaybı (focal loss) fonksiyonu adı verilen yeni bir kayıp fonksiyonu getirilerek çözülmüştür. Odak kaybı, zorlu eğitim örneklerine odaklanır ve eğitim sırasında dedektörü yozlaştıran ve yoran çok sayıda kolay olumsuz örnekten kaçınır. RetinaNet, önceki tek aşamalı dedektörlerin yüksek hızını devralırken, dengesiz pozitif ve negatif örnekler ile eğitmenin zor olduğu tek aşamalı dedektörlerin dezavantajını büyük ölçüde ortadan kaldırır (Lin ve diğerleri, 2017b).



**Şekil 3.20.** EfficientDet mimarisi (Tan ve diğerleri, 2020b)

**YOLO:** Redmon ve diğerleri (2016), hem çoklu sınıflar hem de sınırlayıcı kutular için güven puanlarını tahmin etmek amacıyla en üstteki öznetelik haritasının tamamını

kullanan *Yalnızca Bir Kez Bakarsın* (YOLO) adlı bir mimari önermiştir. YOLO, birleşik bir nesne tespit modelidir. Görüntüler üzerinden nesne tespit görevini bir regresyon problemi olarak kabul eder ve doğrudan sınırlayıcı kutu koordinatlarını ve görüntü piksellerinden ilgili sınıf olasılığını önerir. Eğitim sırasında doğrudan girdi görüntüleri üzerinden öğrenen tek bir konvolüsyonel uçtan uca eğitilebilir modülden oluşur. Bu birleşik model, geleneksel nesne tespit yöntemlerine göre çeşitli avantajlara sahiptir. YOLO ilk olarak giriş görüntüsünü  $S \times S$ 'lik ızgaralara böler. Bu ızgaralar  $3 \times 3$ ,  $5 \times 5$  ve  $9 \times 9$  gibi farklı şekillerde seçilebilir. Orijinal YOLO'da ızgara hücreleri  $7 \times 7$  olarak alınmıştır. Bir nesnenin merkezi bir ızgara hücresine düşerse, bu ızgara hücresi o nesneyi algılamaktan sorumludur.

Her ızgara hücresi,  $B$  adet sınırlayıcı kutu (orijinal YOLO'da  $B = 2$  olarak verilmiştir) ve bu kutular için güven puanlarını tahmin eder. Bu güven puanları, modelin sınırlayıcı kutunun bir nesne içerdiğinden ne kadar emin olduğunu ve ayrıca kutunun tahmin ettiğinin ne kadar doğru olduğunu yansıtır. Denklem (3.2)'de güven değerinin ( $GD$ ) hesaplanması için kullanılan formül verilmiştir:

$$GD = \Pr(\text{nesne}) * IoU_{\text{tahmin}}^{\text{gerçek}} \quad (3.2)$$

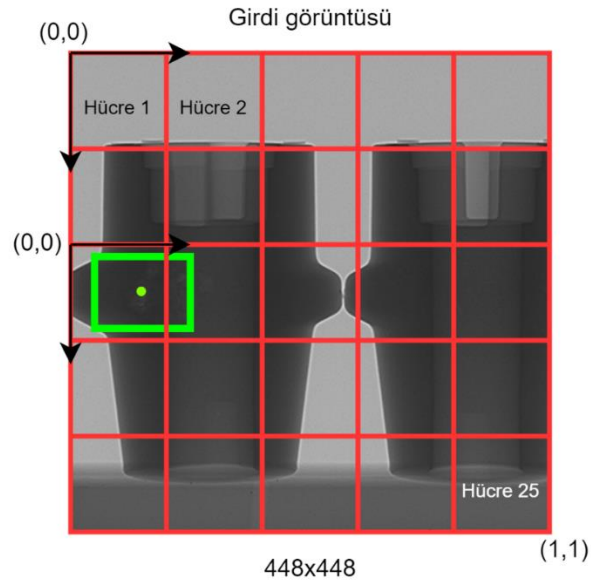
Denklem (3.2)'de yer alan  $\Pr(\text{nesne})$ , bir sınırlayıcı kutuda nesne olma olasılığını;  $IoU_{\text{tahmin}}^{\text{gerçek}}$  ise sınırlayıcı kutu ile temel-gerçek kutu arasındaki birleşim üzerinden kesişim değerini temsil etmektedir. Eğer ilgili ızgara hücresinde hiçbir nesne yoksa güven puanı sıfır olur. Aksi halde güven puanının tahmin edilen sınırlayıcı kutu ile temel-gerçek arasındaki  $IoU$  değerine eşit olması beklenir (hücrede nesne varsa  $\Pr(\text{nesne}) = 1$  olur).

Her sınırlayıcı kutu beş adet terim içerir. Bu terimler sırasıyla  $x$ ,  $y$ ,  $w$ ,  $h$  ve güven puanıdır.  $(x, y)$  koordinatları, ızgara hücresinin sınırlarına göre sınırlayıcı kutunun merkezini temsil eder. Genişlik ( $w$ ) ve yükseklik ( $h$ ), görüntünün tamamına göre tahmin edilir. Son olarak güven tahmini, tahmin edilen kutu ile herhangi bir temel-gerçek kutusu arasındaki  $IoU$  değerini temsil eder. Her bir ızgara hücresi ayrıca  $C$  adet sınıfın koşullu olasılıklarını  $\Pr(\text{sınıf}_i | \text{nesne})$  olarak tahmin eder. Sınırlayıcı kutu sayısı  $B$ 'den

bağımsız olarak, ızgara hücresi başına yalnızca bir sınıf olasılıkları kümesi tahmin edilir. Test aşamasında koşullu sınıf olasılıkları ve bireysel sınırlayıcı kutu güven tahminlerinin çarpımı Denklem (3.3)'de verilmiştir (Redmon ve diğerleri, 2016).

$$\Pr(sınıf_i|nesne) * \Pr(nesne) * IoU_{tahmin}^{gerçek} = \Pr(sınıf_i) * IoU_{tahmin}^{gerçek} \quad (3.3)$$

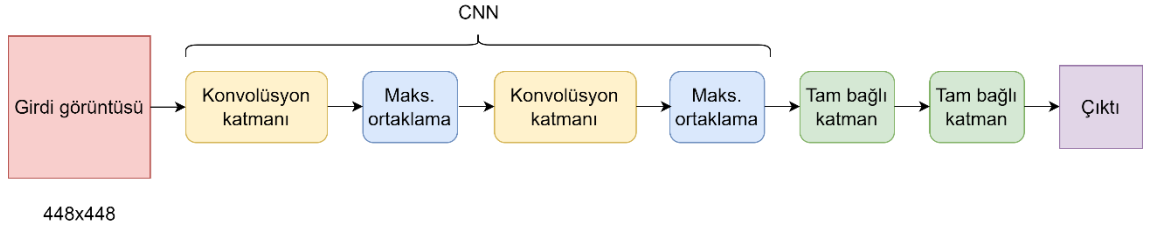
Denklem (3.3) ile her bir sınırlayıcı kutu için sınıfa özel güven puanı elde edilir. Bu puan, hem ilgili sınıfa ait nesnenin kutuda bulunma olasılığının hem de tahmin edilen kutunun nesneye ne kadar iyi uyduğunun göstergesidir. Şekil 3.21'de 5×5'lik ızgara hücrelerine bölünmüş bir YOLO girdisi yer almaktadır. Şekilde toplam 25 adet ızgara hücresi bulunmakla birlikte on birinci ızgara hücresine kusurun merkezi denk geldiği için bu kusurun tespitinden on birinci ızgara hücresi sorumludur. Her ızgara hücresinin sol üst köşesi (0,0) koordinatlarını ve sağ alt köşesi (1,1) koordinatlarını temsil etmektedir. YOLO'da çıktı tensörü  $S \times S \times (B * 5 + C)$  olarak kodlanır.  $S = 5$  ve  $B = 2$  için üç sınıflı ( $C = 3$ ) bir tespit işleminde toplamda  $5 \times 5 \times (2 * 5 + 3) = 5 \times 5 \times 13$  boyutunda bir tensör elde edilir.



**Şekil 3.21.** YOLO girdisinin ızgara hücrelerine bölünmesi



YOLO'daki ağ mimarisinde, görüntü sınıflandırması için kullanılan GoogLeNet (Szegedy ve diğerleri, 2015) modelinden esinlenilmiştir. Ağ, 24 konvolüsyon katmanı ve hemen peşinden gelen iki tam bağlı katmandan oluşur. GoogLeNet tarafından kullanılan başlangıç modülleri yerine, Lin ve ark. (2013) tarafından sunulan modele benzer şekilde  $1 \times 1$ 'lik indirgeme katmanları ve ardından  $3 \times 3$ 'lük konvolüsyon katmanları kullanılmıştır. YOLO'da kullanılan ağın yapısı Şekil 3.22'de verilmiştir.



**Şekil 3.22.** YOLO ağ mimarisinin basit gösterimi (Redmon ve diğerleri, 2016)

Ağdaki son katman hem sınıf olasılıklarını hem de sınırlayıcı kutu koordinatlarını tahmin eder. Sınırlayıcı kutu genişliği ve yüksekliği, görüntü genişliği ve yüksekliği ile 0-1 arasında olacak şekilde normalleştirilir. Son katman için doğrusal bir etkinleştirme fonksiyonu kullanır ve diğer tüm katmanlar Denklem (3.4) ile verilen *Leaky ReLU* aktivasyon fonksiyonunu kullanır.

$$\phi(x) = \begin{cases} x, & \text{eğer } x > 0 \text{ ise} \\ 0,1x, & \text{diğer durumlarda} \end{cases} \quad (3.4)$$

Denklem 3.4'te yer alan  $x$  değişkeni katmanlardaki çıktıları temsil etmektedir. Modelin çıktısındaki hata karelerinin toplamı için optimizasyon gerçekleştirilir. İdeal olmayabilecek sınıflandırma hatası ile yerleştirme hatasına eşit ağırlık verilmiştir. Ayrıca, her görüntüde birçok ızgara hücresi herhangi bir nesne içermez. Bu durum, bu hücrelerin güven puanlarını sıfırlar ve genellikle nesnelere içeren hücrelerden gelen gradyanı bastırır. Bu, model kararsızlığına yol açarak eğitim sürecini olumsuz etkiler (Redmon ve diğerleri, 2016). Bu durumun üstesinden gelmek için sınırlayıcı kutu koordinat tahminlerinden kaynaklanan kayıp artırılır ve nesne içermeyen kutular için

güven tahminlerinden kaynaklanan kayıp azaltılır. Bu amacı gerçekleştirmek için  $\lambda_{koordinat}$  ve  $\lambda_{nesneyok}$  olmak üzere iki parametre kullanılmaktadır (iki parametre sırasıyla 5 ve 0,5 olarak ayarlanmıştır). Hata karelerinin toplamı aynı zamanda büyük sınırlayıcı kutulardaki ve küçük sınırlayıcı kutulardaki hataları eşit olarak ağırlıklandırır. Hata metriği, büyük kutulardaki küçük sapmaların küçük kutulardan daha az önemli olduğunu yansıtmalıdır. Bunu kısmen ele almak için, doğrudan genişlik ve yükseklik yerine sınırlayıcı kutu genişliği ve yüksekliğinin karekökü tahmin edilir.

YOLO, her bir ızgara hücresi başına birden çok sınırlayıcı kutu tahmin eder. Eğitim esnasında, her nesneden yalnızca bir sınırlayıcı kutunun sorumlu olması istenir. Bunun için en yüksek mevcut *IoU* değerine sahip olan tahmine dayalı bir seçim işlemi yapılır. Bu, sınırlayıcı kutu tahmin edicileri arasında uzmanlaşmaya yol açar. Her tahmin edici, belirli boyutları, en-boy oranlarını veya nesne sınıflarını tahmin etmede daha iyi hale gelir. Eğitim sırasında Denklem (3.5)'teki çok terimli kayıp fonksiyonu optimize edilir.

$$\begin{aligned}
& \lambda_{koordinat} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbf{1}_{ij}^{nesne} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
& + \lambda_{koordinat} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbf{1}_{ij}^{nesne} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbf{1}_{ij}^{nesne} (C_i - \hat{C}_i)^2 \\
& + \lambda_{nesneyok} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbf{1}_{ij}^{nesneyok} (C_i - \hat{C}_i)^2 \\
& + \sum_{i=0}^{S^2} \mathbf{1}_i^{nesne} \sum_{c \in sınıflar} (p_i(c) - \hat{p}_i(c))^2
\end{aligned} \tag{3.5}$$

Denklem (3.5)'te yer alan  $\mathbf{1}_i^{nesne}$  terimi  $i$  hücresinde nesne görünüp görünmediğini,  $\mathbf{1}_{ij}^{nesne}$ ,  $i$  hücresindeki  $j$  sınırlayıcı kutu tahmincisinin bu tahminden sorumlu olduğunu

belirtir. Kayıp fonksiyonu sadece ilgili ızgara hücresinde bir nesne varsa sınıflandırma hatasını cezalandırmaktadır. Ayrıca, yalnızca bu tahminci temel-gerçek kutusundan sorumluyorsa (yani, o ızgara hücresindeki herhangi bir tahmincinin en yüksek *IoU* değerine sahipse) sınırlayıcı kutu koordinat hatasını cezalandırır.

YOLO, her ızgara hücresi yalnızca iki kutu öngördüğü ve bu kutular yalnızca bir sınıfa sahip olabileceği için sınırlayıcı kutu tahminlerinde bazı limitlere sahiptir. Özellikle kuş sürüsü veya yerdeki yapraklar gibi birbirine yakın küçük nesnelere tahmin edilmesi sayıca sınırlandırılmıştır. Ayrıca model küçük nesnelere tespit etme konusunda da kötü bir performans ortaya koymaktadır (Redmon ve diğerleri, 2016). Model verilerden sınırlayıcı kutuları tahmin etmeyi öğrendiğinden, ilk defa karşılaştığı veya olağandışı en-boy oranlarındaki nesnelere genelleme yapmakta zorlanır. Son olarak kayıp fonksiyonu yapısından dolayı küçük sınırlayıcı kutulardaki hatalara büyük sınırlayıcı kutulardaki hatalarda olduğu gibi davranır. Büyük bir kutudaki küçük bir hata genellikle zararsız olmakla birlikte, küçük bir kutudaki küçük bir hatanın *IoU* üzerinde çok daha büyük bir etkisi vardır (Redmon ve diğerleri, 2016).

**YOLOv2:** YOLO, son teknoloji algılama sistemlerine göre çeşitli yetersizliklere sahiptir. YOLO'nun Hızlı R-CNN ile karşılaştırıldığında hata analizi, YOLO'nun önemli miktarda lokalizasyon hatası yaptığını göstermektedir. Ayrıca, YOLO, bölge önerisine dayalı yöntemlere kıyasla nispeten düşük hatırlamaya (recall) sahiptir. Bu nedenle Redmon ve Farhadi (2017), sınıflandırma doğruluğunu korumaya çalışırken esas olarak hatırlama ve lokalizasyonu geliştirmeye odaklanmışlardır.

Yığın (batch) normalizasyonu, derin öğrenme modellerinde en popüler normalizasyon yöntemlerinden biridir (Thuan, 2021). Bu normalizasyon işlemi, eğitim sırasında girdi katmanlarının dağılımını stabilize ederek derin sinir ağlarının daha hızlı ve daha kararlı eğitimini sağlar (Ioffe ve Szegedy, 2015). Bu yaklaşımın amacı, öznelikleri (aktivasyon fonksiyonundan geçtikten sonra her katmanın çıktısı) standart sapması bir olan sıfır ortalamalı bir duruma normalize etmektir. Yığın normalizasyonu YOLOv2'nin tüm konvolüsyon katmanlarına uygulanmıştır. Bu teknik sadece eğitim süresini kısaltmakla kalmayıp, aynı zamanda ağı genelleştirilmesini de artırır. YOLOv2'ye uygulanan yığın normalizasyonu, *mAP* değerini yaklaşık %2 artırmıştır (Redmon ve Farhadi, 2017).

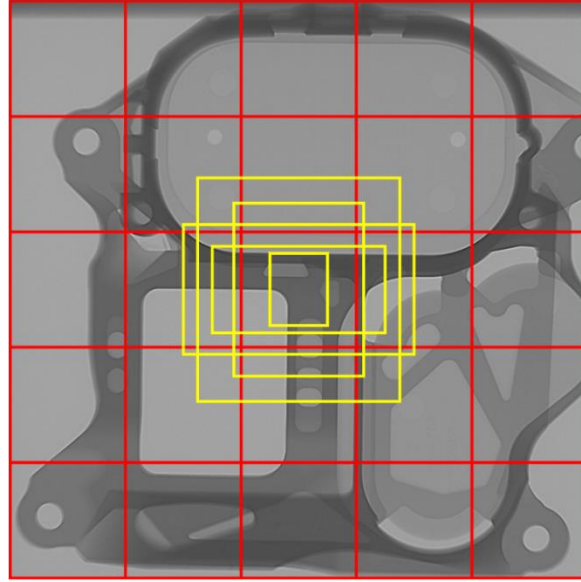
Orijinal YOLO'da,  $224 \times 224$  giriş görüntüsü ile öznitelik çıkarıcıyı (sınıflandırma ağı) eğitmek için ilk 20 konvolüsyon katmanı kullanılmıştır. Daha sonra kalan dört konvolüsyon katmanı ve iki tam bağlı katman eklenmiştir. Giriş görüntüsünün çözünürlüğü nesne dedektörü olarak kullanılmak üzere eşzamanlı olarak  $448 \times 448$ 'e yükseltilmiştir (Redmon ve diğerleri, 2016). YOLOv2'de ise,  $224 \times 224$ 'lük giriş görüntüsü ile öznitelik çıkarıcının eğitim aşamasını tamamladıktan sonra model, nesne dedektörü eğitimi için mimariyi kullanmadan önce öznitelik çıkarıcıyı  $448 \times 448$ 'lik giriş görüntüsü ile on *epoch*'tan fazla eğitmeye devam eder. Bu, öznitelik çıkarıcının eğitim aşamasından nesne tespiti eğitim aşamasına geçildiğinde görüntü boyutunu aniden artırmak yerine modelin  $448 \times 448$ 'lik büyük bir çözünürlüğe uyarlanmasına yardımcı olur. Bu yüksek çözünürlüklü sınıflandırma ağı, yaklaşık %4'lük bir *mAP* artışı sağlamıştır (Redmon ve Farhadi, 2017).

YOLOv1'in ana fikri, merkezi ilgili ızgara hücrelerinin içinde olan bir nesneyi tespit etmekten sorumlu olacak bir ızgara hücresi kullanmaktır. Bu nedenle, merkezi aynı ızgara hücrelerinin içinde olan iki veya daha fazla nesne olduğunda hatalı veya eksik tahmin olabilir. Bu problemi çözmek için yazarlar, bir ızgara hücrelerinin birden fazla nesneyi tahmin etmesine izin vermeye çalışmışlardır. YOLOv2'de, YOLOv1'deki gibi tam bağlı katmanlar kullanmak yerine sınırlayıcı kutuları tahmin etmek için bir bağlantı kutusu mimarisi tanıtılmıştır (Redmon ve Farhadi, 2017). Bağlantı kutusu, istenen nesnelere en iyi eşleşen önceden tanımlanmış kutulardır. Sınırlayıcı kutular yalnızca temel-gerçek kutularına dayalı olarak değil, aynı zamanda önceden tanımlanmış  $k$  adet bağlantı kutusuna göre de tahmin edilir. Şekil 3.23'te bağlantı kutusu örnekleri gösterilmiştir. Burada  $5 \times 5$ 'lik ızgara hücrelerine (kırmızı renkte) bölünmüş bir görüntü kullanılmıştır. En ortada yer alan hücrede toplamda beş adet bağlantı kutusu (sarı renkte) oluşturulmuştur. Bu bağlantı kutularının farklı ölçeklerde ve en-boy oranlarında olmalarının sebebi farklı boyut ve şekillerdeki nesnelere tespit etme kabiliyetini artırmak içindir. Böylece ince, küçük veya büyük nesnelere tespit etmek amaçlanmıştır. Ayrıca bir hücre içinde beş farklı bağlantı kutusu kullanılmasıyla aynı hücre içinde beş nesne tespit edilebilmektedir.

En uygun bağlantı kutularını manuel olarak seçmek yerine, benzer şekillere sahip sınırlayıcı kutuları kümelemek ve ardından en yakın merkez noktasına sahip ortalama

$IoU$ 'yu çizmek için eğitim setindeki sınırlayıcı kutularda (tüm temel-gerçek kutuları dahil)  $k$ -ortalamalar kümeleme algoritması kullanılmıştır. Ancak yazarlar, Öklid mesafesini kullanmak yerine, sınırlayıcı kutu ile ağırlık merkezi arasında  $IoU$ 'yu kullanmışlardır. Ayrıca orijinal YOLOv2’de yazarlar bağlantı kutularının sayısını  $k = 5$  olarak önermişlerdir.

YOLO, bir kutunun sınırlayıcı konumunu tahmin etme konusunda herhangi bir kısıtlamaya sahip değildir. Parametreler rastgele başlatıldığında, sınırlayıcı kutu görüntünün herhangi bir yerinde tahmin edilebilir. Bu, modeli eğitimin ilk aşamalarında kararsız hale getirir. Sınırlayıcı kutunun konumu, sınırlayıcı kutuyu tahmin etmekten sorumlu ızgara hücrelerinden uzakta olabilir. YOLO’daki her ızgara hücresi 0-1 ölçeğinde belirtilmiştir. YOLO’da bir ızgara hücresinin sol üst noktanın koordinatı (0,0) ve sağ alt nokta (1,1)'dir. 0-1 aralığında sınırlayıcı kutu merkezinin değerini sınırlamak için sigmoid fonksiyonu ( $\sigma$ ) kullanılmaktadır. YOLOv2 sınırlayıcı kutuyu yalnızca ızgara hücresine değil, aynı zamanda temel-gerçek kutusu ile en yüksek  $IoU$ 'ya sahip bağlantı kutularından birine atar. Kutunun merkez koordinatları, bir sigmoid ( $\sigma$ ) fonksiyonu kullanılarak filtre uygulamasının konumuna göre tahmin edilir (Redmon ve Farhadi, 2017).



Bağlantı kutusu      Izgara hücresi

**Şekil 3.23.** YOLOv2’de kullanılan bağlantı kutularının temsili gösterimi

Model, orijinal YOLO'da olduğu gibi her bir sınırlayıcı kutu için  $(x, y, w, h, \text{güven puanı})$  karşılık gelen beş parametre değerini  $(t_x, t_y, t_w, t_h, t_o)$  tahmin eder. Ancak, bu parametreler önceden tanımlanmış bağlantı kutusuyla olan korelasyona dayalı olarak yeniden hesaplanır. Hücre görüntünün sol üst köşesinden  $(c_x, c_y)$  kadar kaydırılırsa ve verilen bağlantı kutusunun genişliği ve yüksekliği  $(p_x, p_y)$  ise, tahminler şu şekilde hesaplanır (Denklem (3.6)-(3.10)):

$$b_x = \sigma(t_x) + c_x \quad (3.6)$$

$$b_y = \sigma(t_y) + c_y \quad (3.7)$$

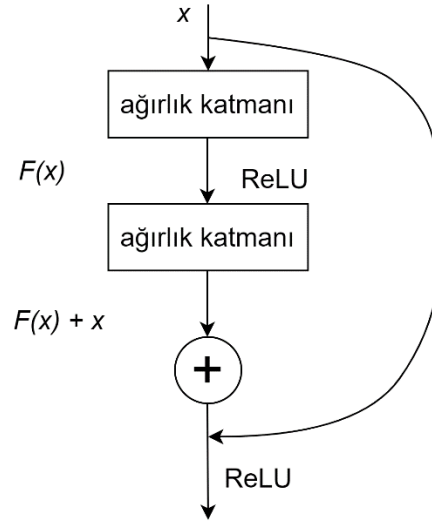
$$b_w = p_w e^{t_w} \quad (3.8)$$

$$b_h = p_h e^{t_h} \quad (3.9)$$

$$\Pr(\text{nesne}) * IoU(b, \text{nesne}) = \sigma(t_o) \quad (3.10)$$

Böylece, son tahmin edilen sınırlayıcı kutu  $(b_x, b_y, b_w, b_h, \sigma(t_o))$  olacaktır. Model konum tahminini sınırladığından, parametrelendirmenin öğrenilmesi daha kolaydır ve bu da ağı daha kararlı hale getirir. YOLOv2'de, bağlantı kutuları ile yaklaşık %5'lik *mAP* iyileştirilmiştir (Redmon ve Farhadi, 2017).

**YOLOv3:** YOLOv2, Darknet mimarisi için 30 konvolüsyon katmanı kullanmaktadır. Derin sinir ağları için daha fazla katman, daha fazla doğruluk anlamına gelir. Ancak, giriş görüntüsü daha derin katmanlara iletilirken alt örnekleme (downsampling) yapılır ve küçük detaylara sahip özneliklerin kaybolmasına neden olur. YOLOv2'nin genellikle küçük nesnelere tespit etme konusunda başarısızlığının sebebi budur. ResNet, aktivasyonların gradyan kaybolmadan (bu problem “gradient vanishing” olarak geçmektedir) daha derin katmanlar arasında yayılmasına yardımcı olmak için atlama bağlantıları (skip connections) fikrini getirmiştir (Şekil 3.24) (He ve diğerleri, 2016b). Bu fikrin detaylı bir incelemesi Shrivastava ve diğerleri (2016) tarafından yapılan çalışmada yer almaktadır.



**Şekil 3.24.** ResNet atlama bağlantıları mimarisi (He ve diğerleri, 2016b)

YOLOv3'te kullanılan öznetelik çıkarıcı yani CNN mimarisi; YOLOv2, Darknet-53 (53 konvolüsyon katmanlı) ve artık ağların (ResNet) bir melezidir (Redmon ve Farhadi 2018b). ResNet'in artık blokları sayesinde, üst üste gelen katmanlar ağ performansını düşürmez. Ayrıca, daha derin katmanlar, sık katmanlardan doğrudan daha fazla bilgi alır. Böylece küçük detayları temsil eden öznetelikler kaybolmaz. Darboğaz yapısı ( $1 \times 1$  ve ardından  $3 \times 3$  konvolüsyonel katman) içeren beş artık bloklu Darknet-53 mimarisi Şekil 3.25'te gösterilmektedir. Model, orijinal olarak eğitim aşamasında öznetelikleri çıkarmak için 53 katmanlı ağa sahip Darknet-53 mimarisini kullanmaktadır. Ayrıca, nesne dedektörü eğitimi için tespit aşamasında (baş kısmında) 53 katman daha eklenir ve YOLOv3, toplam 106 tamamen konvolüsyonel katmanlı bir mimariye sahip olur (Redmon ve Farhadi, 2018b).

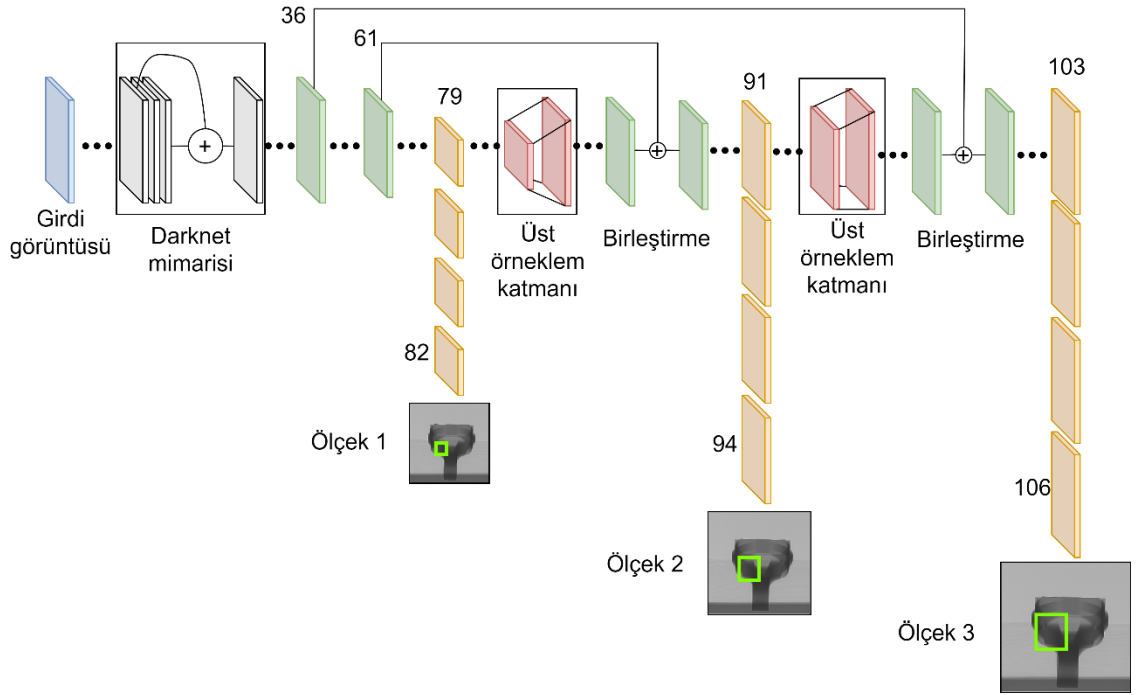
YOLO'nun önceki iki versiyonunda, Darknet mimarisine sahip özellik çıkarıcının eğitiminden sonra, girdi birkaç katmana daha iletilir ve son olarak nesne dedektörünün son katmanlarında tahminler yapılır. Ancak YOLOv3, önceki gibi tahminleri son katmanlarda yapmak yerine farklı katmanlarda gerçekleştirmektedir. YOLOv3 önceki versiyonlardan farklı olarak üç farklı ölçekte tahmin görevini gerçekleştirmektedir. Daha detaylı olarak YOLOv3, adım adım sırasıyla ölçeği 32, 16 ve 8 olarak verilen 82'nci, 94'üncü ve 106'ncı katmanlarda tahminler yapar (Şekil 3.26). Her bir dedektör için YOLOv3'ün çıktı tensörü  $S \times S \times (B \times (5 + C))$  olur. İlk tespit 82. katman tarafından

yapılır. Girdi görüntüsünün çözünürlüğü varsayılan olarak  $416 \times 416$  alınmıştır. İlk 81 katman boyunca iletdikten sonra girdi görüntüsü 32'lik adımlarla alt örneklenir ve elde edilen öznitelik haritası, ızgara hücrelerine karşılık gelen  $13 \times 13$  boyutunda olur (Şekil 3.26) (Kathuria, 2018). Her bir tespit katmanında, öznitelik haritalarına  $1 \times 1$  tespit filtreleri uygulanarak tespit yapılır.  $1 \times 1$ 'lik filtre, öznitelik haritasının her ızgara hücresi için  $B$  adet sınırlayıcı kutuyu tahmin etmekten sorumludur. YOLOv3,  $B = 3$  (her hücre için üç sınırlayıcı kutu) ve  $C = 80$  (80 sınıf) ile COCO veri kümesi üzerinde eğitilmiştir. Bu nedenle filtre boyutu  $1 \times 1 \times (3 \times (5 + 80)) = 1 \times 1 \times 255$ 'tir. İlk tespit katmanında, nihai öznitelik haritası  $13 \times 13 \times 255$  olacaktır (Thuan, 2021). Benzer şekilde diğer katmanlardaki tespitler de Şekil 3.26'daki gibi yapılır.

	Katman tipi	Filtre sayısı	Boyut	Çıktı
	Konvolüsyonel	32	3x3	256x256
	Konvolüsyonel	64	3x3/2	128x128
1x	Konvolüsyonel	32	1x1	
	Konvolüsyonel	64	3x3	
	Artık			128x128
	Konvolüsyonel	128	3x3/2	64x64
2x	Konvolüsyonel	64	1x1	
	Konvolüsyonel	128	3x3	
	Artık			64x64
	Konvolüsyonel	256	3x3/2	32x32
8x	Konvolüsyonel	128	1x1	
	Konvolüsyonel	256	3x3	
	Artık			32x32
	Konvolüsyonel	512	3x3/2	16x16
8x	Konvolüsyonel	256	1x1	
	Konvolüsyonel	512	3x3	
	Artık			16x16
	Konvolüsyonel	1024	3x3/2	8x8
4x	Konvolüsyonel	512	1x1	
	Konvolüsyonel	1024	3x3	
	Artık			8x8
	Ortalama ortaklama		Global	
	Tam bağlı		1000	
	Softmax			

**Şekil 3.25.** Darknet-53 mimarisi (Redmon ve Farhadi, 2018b)





**Şekil 3.26.** YOLOv3 mimarisindeki üç farklı ölçekte yapılan tahmin (Kathuria, 2018)

Farklı ölçek katmanlarındaki tespitler, YOLOv2'de karşılaşılan bir problem olan küçük nesnelerin tespit edilememesi problemini çözmeye yardımcı olur. Daha büyük boyutlu bir öznetelik haritası daha fazla detay barındırır. Bu nedenle, büyük ölçekli tespit katmanı ( $52 \times 52$ ) küçük nesnelerin algılanmasından sorumluyken, küçük ölçekli tespit katmanı ( $13 \times 13$ ) daha büyük nesnelere algılar (Redmon ve Farhadi, 2018b).

**YOLOv4:** YOLOv4'ün geliştiricileri, mimarinin her bir parçası için bilgisayarlı görmenin en gelişmiş yenilik fikirlerinin birçoğuyla bir dizi deney gerçekleştirmiştir (Bochkovskiy ve diğerleri, 2020b). YOLOv4 modelinin omurgası (öznetelik çıkarıcı) yazarlar tarafından üç seçenek arasında değerlendirilmiştir: CSPResNext53, CSPDarknet53 ve o zamanın en gelişmiş CNN'i olan EfficientNet-B3. Teorik gerekçelere ve birçok deneye dayanarak, CSP Darknet53 sinir ağının en uygun model olduğu belirlenmiştir (Bochkovskiy ve diğerleri, 2020b). CSPResNext50 ve CSPDarknet53 (CSP - Cross Stage Partial) önceki girdiyi kullanan ve yoğun katmana geçmeden bu girdiyi önce mevcut girdiyle birleştiren DenseNet (Huang ve diğerleri, 2017b) mimarisinden türetilmiştir. DenseNet, kaybolan gradyan problemlerini (ResNet gibi)

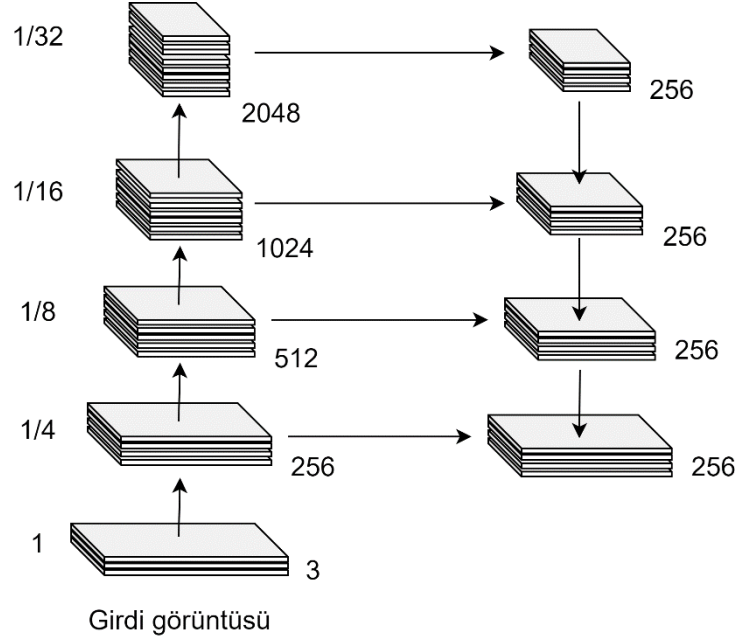
hafifletmek amacıyla çok derin bir sinir ağında katmanları birbirine bağlamak için tasarlanmıştır.

CSP, temel katmanda tam boyutlu girdi öznitelik haritasını kullanmak yerine girdinin iki bölüme ayrılması dışında DenseNet ile aynı prensibe dayanmaktadır. Bu fikirleri YOLOv3'teki Darknet-53 mimarisiyle birleştirerek, kalan blokların yerini tam bağlı bloklar almıştır. CSP, yayılma yoluyla özellikleri korur, ağı özellikleri yeniden kullanmaya teşvik eder ve ağ parametrelerinin sayısını azaltır. Ayrıca CSP, daha derin katmanlara daha verimli bilgiler iletmek için küçük detaylı özniteliklerin korunmasına yardımcı olur. Yoğun bağlantılı konvolüsyon katmanlarının aşırı artışının tespit hızında düşüşe neden olabileceği göz önüne alındığında, Darknet-53 omurga ağındaki yalnızca daha zengin anlamsal özellikleri çıkarabilen son konvolüsyon bloğu, yoğun bir blok olacak şekilde geliştirilmiştir (Huang ve diğerleri, 2020).

CSPDarknet53 omurgasının çıktılarındaki öznitelik haritaları boyundaki öznitelik toplama mimarisine iletilmeden önce, alıcı alanı artırmak (receptive field) ve en önemli öznitelikleri ayırmak için ek bir bloğa, yani Uzamsal Piramit Ortaklama (SPP) bloğuna (He ve diğerleri, 2015) (Uzamsal Piramit Ortaklama) gönderilir. Birçok CNN tabanlı model, yalnızca belirli boyutlardaki girdi görüntülerini kabul eden tam bağlı katmanlar içerir. SPP, girdi boyutundan bağımsız olarak sabit boyutlu bir çıktı üretmek amacıyla doğmuştur. Aynı zamanda SPP, kendisinin çok ölçekli sürümlerini bir havuzda toplayarak önemli özniteliklerin çıkarılmasına da yardımcı olur.

Giriş görüntüsü, omurga üzerinden iletdikten sonra, görüntü özellikleri anlamsal özniteliklere (veya öğrenilen özniteliklere) işlenir. Başka bir deyişle, giriş görüntüsü düşük seviyeli katmanlardan ne kadar derine inerse, anlamsal özniteliklerin karmaşıklığı daha fazla artacak ve alt örnekleme nedeniyle öznitelik haritalarının uzamsal çözünürlüğü daha fazla düşecektir (Thuan, 2021). Bu durum, küçük detaylara sahip özniteliklerin yanı sıra uzamsal bilgi kaybına yol açar. Bu küçük detaylara sahip öznitelikleri korumak için (Redmon & Farhadi, 2018b) YOLOv3'ün boyun bölümünde Özellik Piramit Ağı (FPN) mimarisi fikri uygulanmıştır (Redmon ve Farhadi, 2018b). FPN mimarisi (Şekil 3.27), semantik özellikleri (yüksek seviyeli katmandan) aktarmak için yukarıdan aşağıya bir yol uygular ve daha sonra bunları büyük ölçekli dedektörde küçük nesnelere tahmin etmek

için küçük detaylara sahip özniteliklere (omurgadaki düşük seviyeli katmandan) birleştirir (Bochkovskiy ve diğerleri, 2020b).



**Şekil 3.27.** FPN mimarisi

Yol Toplama Ağı (PAN), FPN'nin gelişmiş bir sürümüdür (Liu ve diğerleri, 2018). FPN mimarisindeki akış yukarıdan aşağıya bir yoldur. Bu nedenle yalnızca FPN'deki düşük seviyeli katmanlardan gelen büyük ölçekli dedektör, yüksek seviyeli katmanlardan semantik öznitelikleri ve yan omurgadaki düşük seviyeli katmanlardan detaylı özellikleri aynı anda alabilir. FPN'deki yüksek seviyeli katmanlardan gelen küçük ölçekli dedektör, nesnelere tespit etmek için yalnızca anlamsal öznitelikleri kullanır. Küçük ve orta ölçekli dedektörün performansını iyileştirmek için, anlamsal öznitelikleri ve detaylı öznitelikleri üst düzey katmanlarda birleştirme fikri incelenmiştir (Bochkovskiy ve diğerleri, 2020b). Ancak günümüzdeki derin sinir ağlarının omurgası çok sayıda katman içermektedir. Bu nedenle, FPN'de, detaylı özniteliklerin düşük seviyeden yüksek seviyeli katmanlara geçiş için uzun bir yol alması gerekir. PAN mimarisinin yazarları, FPN'de kullanılan yukarıdan aşağıya yolun yanına bir aşağıdan yukarıya artırım yolu eklemeyi önermişlerdir (Liu ve diğerleri, 2018). Böylece, detaylı öznitelikleri düşük seviyeli katmanlardan en üsttekilere

doğrudan bağlamak için bir “kısayol” oluşturulmuştur. Bu kısayol 10 adetten daha az katman içerir ve bilgi akışını kolaylaştırır (Liu ve diğerleri, 2018).

PAN mimarisinde aşağıdan yukarıya artırım yolu, her aşamada aynı uzamsal boyutlarda öznitelik haritaları üreten katmanlar içeren FPN’ye ait yukarıdan aşağıya yolun bir kopyası gibi görünebilir. Bu öznitelik haritaları, eleman bazında toplama işlemi ile yanal mimariye bağlanır. Buna rağmen YOLOv4 için değiştirilmiş PAN mimarisinde, yazarlar bunu birleştirme işlemi ile değiştirmiştir.

Tek aşamalı dedektörlerde, baş bölümünün işlevi yoğun tahminler yapmaktır. Yoğun tahmin, tahmin edilen sınırlayıcı kutu koordinatlarını (merkez, yükseklik, genişlik), güven skorunu ve olasılık sınıflarını içeren bir vektörden oluşan son tahmindir. YOLOv4, çapa tabanlı tespit adımları ve üç ölçekte tespit ile YOLOv3’teki baş yapısının aynısını kullanmaktadır (Bochkovskiy ve diğerleri, 2020b).

Son olarak Bochkovskiy ve diğerleri (2020b), model içinde *mozaik veri artırımı* adı verilen yeni bir veri artırım tekniği kullanmaktadırlar. Bu yöntem rastgele seçilen dört adet (veya farklı sayılarda olabilir) görüntünün tek bir görüntü olarak birleştirilmesine dayalıdır. Bu sayede model kendi içinde de verileri çeşitlendirmektedir.

**YOLOv5:** YOLOv4’ün ortaya çıkışından bir ay sonra, YOLOv5 adlı YOLO ailesinin yeni bir sürümü daha geliştirilmiştir (Jocher, 2020). Jocher mozaik veri artırmanın mucididir ve bu Alexey Bochkovsky tarafından YOLOv4 makalesinde onaylanmıştır (Bochkovskiy ve diğerleri, 2020b). Ancak, YOLOv5 modeli, adı ve iyileştirmeleri nedeniyle bilgisayarlı görüş topluluğunda birçok tartışmaya neden olmuştur. Bu tartışmaların sebebi YOLOv4 ve YOLOv5’in birbirine çok yakın zamanlarda ortaya çıkmasından ve YOLOv5 için hakemli bir makalenin bulunmamasından kaynaklanmaktadır (Thuan, 2021).

Buna rağmen YOLOv5 mühendislikte bazı avantajlara sahiptir. YOLOv5, önceki sürümlerde olduğu gibi C yerine Python programlama dilinde yazılmıştır. Bu, IoT cihazlarında kurulum ve entegrasyonu kolaylaştırır. Ayrıca, PyTorch topluluğu da Darknet topluluğundan daha büyüktür, bu da PyTorch’un gelecekte daha fazla katkı ve büyüme potansiyeli alacağı anlamına gelir. İki farklı çerçevede iki farklı dilde yazılması nedeniyle, YOLOv4 ve YOLOv5 arasındaki performansı karşılaştırmanın doğruluğu

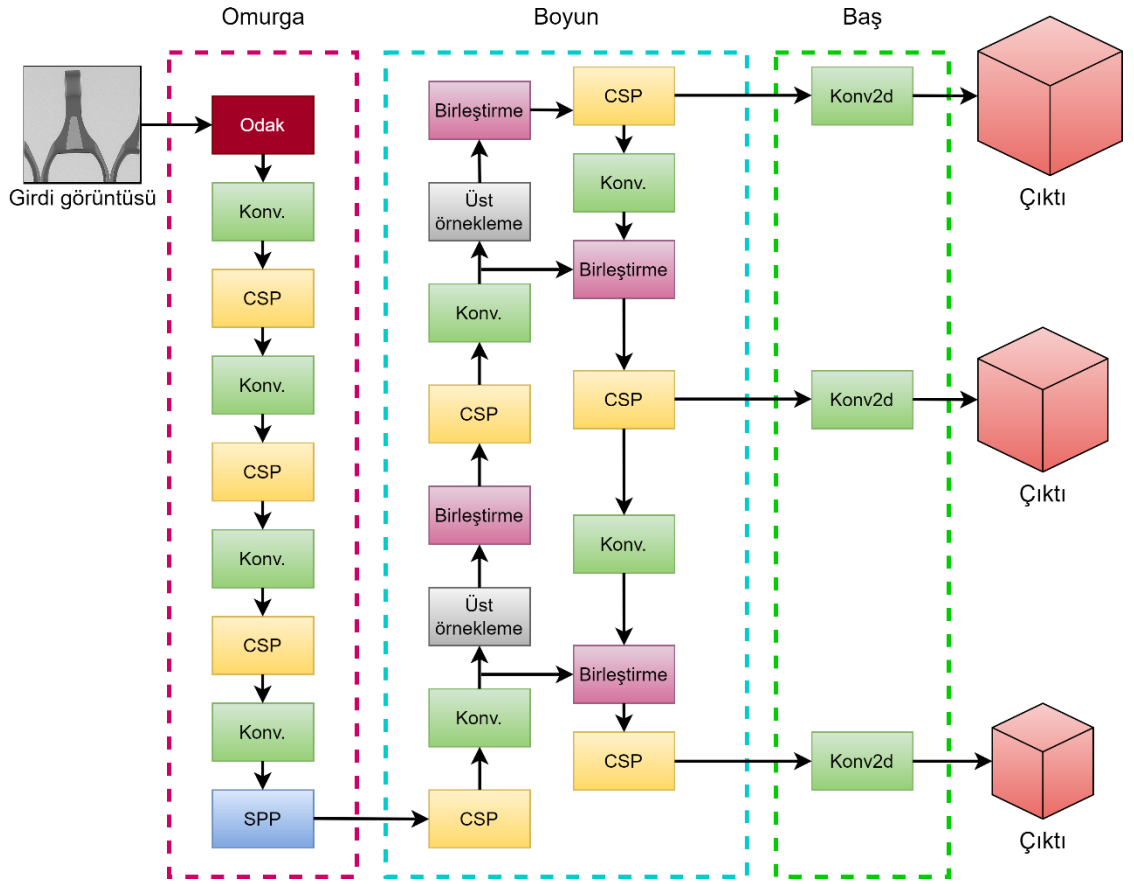
tartışılır. Ancak bir süre sonra YOLOv5, belirli koşullar altında YOLOv4'ten daha yüksek performans göstermiştir ve YOLOv4'ün yanı sıra bilgisayarlı görüş topluluğunda kısmen güven kazanmıştır.

Yukarıda bahsedildiği gibi, YOLOv5 mimarisi, YOLOv4 mimarisine benzer en son yenilikleri entegre etmiştir. Bu nedenle teoride çok açık farklılıklar yoktur. YOLOv5'in ayrıntılı bir makalesi yayınlanmamıştır. Ancak yönteme ait bir Github deposu mevcut olup, bu depo üzerinden iyileştirmeler yapılmaktadır. YOLOv5 modeli aşağıdaki gibi özetlenebilir (Jocher, 2020):

- Omurga: Odak yapısı, CSP ağı
- Boyun: SPP bloğu, PANet
- Baş: YOLOv3 baş bölümü, Genelleştirilmiş IoU (GIoU) kaybı

YOLOv5 yazarının değindiği dikkat çekici nokta, mühendislik farkıdır. Joseph Redmon, YOLOv2'deki bağlantı kutusu yapısını ve eğitim setindeki temel-gerçek kutularına çok benzeyen boyut ve şekilde bağlantı kutularını seçme prosedürünü tanıtmıştır. Jocher (2020), farklı  $k$  değerlerine sahip  $k$ -ortalamalar kümeleme algoritmasını kullanarak, COCO veri kümesi (80 sınıf içeren) için en uygun beş bağlantı kutusunu seçmiş ve bunları varsayılan olarak kullanmıştır. Bu, eğitim süresini azaltır ve ağın doğruluğunu artırır. Ancak, bu beş bağlantı kutusunu benzersiz bir veri kümesine (COCO veri kümesinde 80 sınıfa ait olmayan bir sınıf içeren) uygularken, bu bağlantı kutuları bu benzersiz veri kümesinin temel-gerçek kutularına hızlı bir şekilde uyum sağlayamaz. Bu problemi çözmek için, bilgisayarlı görüş mühendisleri, önce veri için en uygun bağlantı kutularını elde etmek adına genellikle benzersiz veri kümesinde  $k$ -ortalamalar kümeleme algoritmasını çalıştırır. Daha sonra bu parametreler YOLO mimarisinde manuel olarak yapılandırılacaktır (Thuan, 2021).

Jocher (2020), bağlantı kutusu seçim sürecini YOLOv5'e entegre etmeyi önermiştir. Sonuç olarak ağ, kullanılacak veri kümelerinden herhangi birini girdi olarak dikkate almak zorunda değildir ve o veri kümesi için en iyi bağlantı kutularını otomatik olarak öğrenir ve bunları eğitim sırasında kullanır (Solawetz, 2020). YOLOv5 mimarisi Şekil 3.28'de gösterilmiştir. Şekil 3.28'de yer alan *CSP*, öznel çıkarma ağını; *Konv.*, konvolüsyonel katmanı ve *SPP*, uzamsal piramit ortaklamayı göstermektedir.



Şekil 3.28. YOLOv5 mimarisi (Xu ve diğerleri, 2021)

### 3.2.3. Kusur görüntülerinin bölütlenmesi ve eşiklenmesi

Bu alt bölümde kusur görüntülerinde eşikleme (thresholding) yöntemiyle kusurların arka plandan ayrıştırılması ile ilgili yöntemler ele alınmıştır. İlk olarak görüntüleri iyileştirmek için kullanılan görüntü işleme teknikleri açıklanmıştır. Daha sonra eşikleme ve bölütleme yöntemleri ele alınmıştır.

Dijital görüntüler piksellerden oluşur ve pikseller siyah/beyaz, gri tonlu veya renk tonlu olabilir. Dijital bir görüntüdeki her piksel için gri tonu veya renk tonu bilgisini bit sayısı belirler. Bit sayısının fazla olması görüntüde daha fazla renk tonu anlamına gelir. Ayrıca bir görüntünün dosya boyutu bit derinliği ile doğrudan ilgilidir. Bir bit derinliğine sahip bir görüntüdeki piksellerin iki renk tonu olabilir: Siyah ve beyaz. Genellikle gri görüntüler 8-bit olarak kullanılmaktadır. Bu tip görüntülerde toplamda 256 farklı gri tonu vardır. RGB (red-green-blue) modundaki görüntüler üç renk kanalından oluşmaktadır.

Piksel başına 8-bit olan bir RGB görüntüsünde her kanal için 256 farklı renk tonu tanımlanmıştır. Bu da 8-bit RGB bir görüntüde yaklaşık 16 milyon farklı olası renk tonu olduğu anlamına gelmektedir. Tez çalışmasında yer alan veri setindeki görüntülerin tamamı 8-bit gri görüntülerdir.

**Görüntü işleme:** X-ışını kullanılarak oluşturulan radyografi görüntülerinde görüntü kalitesini olumsuz etkileyen piksel tabanlı gürültü bulunabilir. Ayrıca bazı görüntülerdeki kusurlar arka plandan görsel anlamda ayırt edilemeyebilir. Bu durumların üstesinden gelmek için görüntülere gürültü giderme ve histogram eşitleme işlemleri uygulanabilir. Bu çalışmada gri seviyeli görüntülerdeki gürültünün giderilmesi için Gauss bulanıklaştırma tekniği kullanılmıştır. Gauss bulanıklaştırma, bir gauss fonksiyonu kullanarak bir görüntüyü bulanıklaştırmak için kullanılan tekniktir. Bu teknik, tipik olarak, görüntü gürültüsünü ve ayrıntılarını en aza indirmek için bilgisayarla görme ve görüntü işleme uygulamalarında kullanılır. Teknik, bir pikselin değerini komşu piksellerin ortalaması ile güncellemeyi amaçlar. Denklem (3.11)'de bir boyut için Gauss fonksiyonu verilmiştir:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (3.11)$$

Denklem (3.12)'de iki boyut için Gauss fonksiyonu yer almaktadır:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.12)$$

Burada  $x$  yatay ekseninde merkezden uzaklık ve  $y$  dikey ekseninde merkezden uzaklıktır.  $\sigma$  denklemdaki bulanıklık parametresidir ve bu değer ne kadar yüksek olursa daha bulanık bir görüntüye neden olur. Bu fonksiyon bir görüntü üzerine uygulanırken kullanılacak filtrenin boyutu diğer bir önemli parametredir.

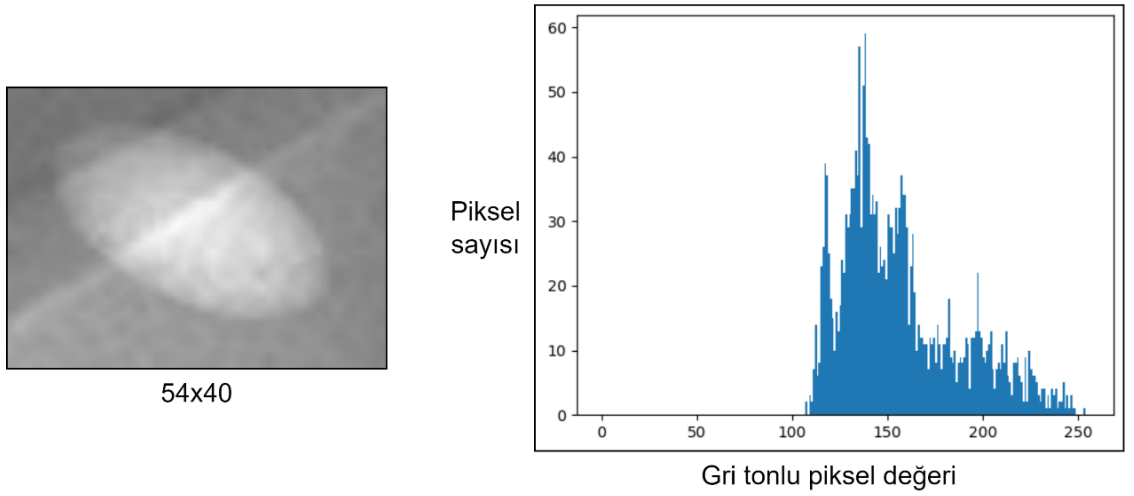
Histogram eşitleme, küresel kontrastı artırmak için yaygın olarak kullanılan bir yöntemdir. Histogram, görüntüdeki piksel değerlerin (piksel yoğunluğu da denilmektedir) dağılımının grafiksel olarak gösterimidir (Şekil 3.29). Şekil 3.29’da yer alan grafikteki yatay eksen görüntüdeki gri değerleri gösterirken, dikey eksen ise o gri değerdeki toplam piksel sayısını (frekans) gösterir. Yatay ekseninde sola doğru ilerledikçe siyaha yaklaşan koyu gri renkler temsil edilirken sağa doğru gidilirse beyaza yakın açık gri renkler temsil edilir. Histogram eşitlemede amaç, görüntü histogramının tekdüze dağılımda olması yani her bir piksel değeri için yaklaşık aynı sayıda piksel bulunmasıdır.

Sıradan histogram eşitleme, görüntü histogramından türetilen aynı dönüşümü kullanarak her pikseli dönüştürür. Piksel değerlerinin dağılımı tüm görüntü üzerinde benzer olduğunda, bu yöntem iyi sonuç verir. Görüntüde, görüntünün çoğundan belirgin şekilde daha açık veya daha koyu alanlar varsa, bu alanlardaki kontrast uygun şekilde iyileştirilmez. Bu problemin üstesinden gelmek için Uyarlamalı Histogram Eşitleme (AHE) kullanılabilir. AHE, komşu piksellerden türetilen bir dönüşümle bir pikseli dönüştürür (Hummel, 1977; Ketcham, 1976). Kontrast Sınırlı AHE (CLAHE), görüntü genişlemesi sorununu azaltmak için kontrast genişlemesinin sınırlı olduğu uyarlanabilir histogram eşitlemenin bir çeşididir (Pizer ve diğerleri, 1987; Zuiderveld, 1994). Bu çalışmada kontrast artışı için CLAHE tekniği tercih edilmiştir. Bu yaklaşım, görüntüyü neredeyse eşit büyüklükte, örtüşmeyen birkaç alana bölmeye dayanır. CLAHE, görüntünün tamamını işlemek yerine, karo adı verilen bölünebilen alanlarla çalışır. Yapay sınırlar daha sonra çift doğrusal enterpolasyon kullanılarak bitişik karoların birleştirilmesiyle ortadan kaldırılır. Pratikte, CLAHE uygulanırken iki parametre kullanılır: kesme limiti ve karo sayısı. İlk parametre kontrast sınırlama eşiğini ayarlamak için kullanılmaktadır. X-ışını görüntülerindeki kusurların daha belirgin olması için CLAHE yönteminden faydalanılmıştır.

**Görüntü eşikleme:** Kusur görüntülerinin yukarıda anlatılan görüntü işleme yöntemleriyle iyileştirmesinden sonra kusurun eşiklenmesi işlemi gelmektedir. Birçok görüntü işleme uygulamasında, nesneye ait olan piksellerin gri seviyeleri, arka plana ait olan piksellerin gri seviyelerinden önemli ölçüde farklıdır. Eşikleme işlemi nesnelere arka plandan ayırmak için basit ama etkili bir araçtır. Eşikleme işleminin çıktısı bir ikili (binary) görüntüdür. Uygulamaya bağlı olarak, bazı çalışmalarda ön plan yani nesne siyah (gri düzeyi 0) arka plan beyaz (gri düzeyi 255) tercih edilirken buradaki çalışmada



kusurlar daha aydınlık bölgeler olduğu için ve nesnelere karşılık geldiği için ön plan yani kusurlar beyaz arka plan siyah olmaktadır. Gürültü, ortam aydınlatması, nesne ve arka plandaki gri seviyelerin homojen olmayan yoğunluğu, yetersiz kontrast ve arka planla orantısız nesne boyutu gibi çeşitli faktörler, eşikleme işlemini zorlaştırır. Son olarak, çeşitli eşikleme algoritmalarının performansını değerlendirmek için nesnel ölçümlerin olmaması ve testlerin zorluğu diğer önemli engeller olmuştur (Sezgin ve Sankur, 2004).



**Şekil 3.29.** Bir kusur görüntüsüne ait histogram

En basit eşikleme yöntemleri, görüntü yoğunluğu  $I_{ij}$ ,  $T$  eşiği olarak adlandırılan sabit bir değerden küçükse görüntüdeki her pikseli siyah bir pikselle veya piksel yoğunluğu bu eşikten büyükse beyaz bir pikselle değiştirir. Bu  $T$  değeri basit eşikleme yönteminde elle seçilir. Ancak bu pratik bir yaklaşım değildir. Bunun yerine otomatik olarak  $T$  değeri belirleyip görüntüyü eşikleyen birçok yöntem literatürde yer almaktadır. Literatürde yer alan ve iyi bilinen bu yöntemlerden bazıları; Otsu (Otsu, 1979), MaxEntropy (Kapur ve diğerleri, 1985), Niblack (Niblack, 1985), Bernsen (Bernsen, 1986), Mean (Glasbey, 1993), Li (Li ve Lee, 1993; Li ve Tam, 1998), Shanbhag (Shanbhag, 1994), Huang (Huang ve Wang, 1995), Sauvola (Sauvola ve Pietikäinen, 2000) ve Phansalkar (Phansalkar ve diğerleri, 2011) olarak verilebilir. Bu yöntemler arasında en yaygın olarak kullanılan yöntem Otsu eşikleme yöntemidir.

Otsu yöntemi, iki sınıfın (arka plan ve ön plan) ağırlıklı bir toplamı olarak tanımlanan sınıf içi varyansı en aza indiren eşiği yinelemeli olarak arar. Gri tonlamalı renkler genellikle 0-255 arasındadır (ikili ise 0-1). Yani, 50'lik bir eşik seçilirse, değeri 50'den küçük olan tüm pikseller arka plan olur ve değeri 50'ye eşit veya daha büyük olan tüm pikseller görüntünün ön planı olur. Herhangi bir  $T$  eşiğinde sınıf içi varyansı bulma formülü Denklem (3.13)'te verilmiştir:

$$\sigma^2(t) = \omega_{ap}(t)\sigma_{ap}^2(t) + \omega_{öp}(t)\sigma_{öp}^2(t) \quad (3.13)$$

Denklem (3.13)'te yer alan  $\omega_{ap}(t)$  ve  $\omega_{öp}(t)$  sırasıyla arka plan ve ön plan için  $T$  eşiğinde her sınıf adına piksel sayısının olasılığını temsil eder ve  $\sigma^2$ , renk değerlerinin varyansını temsil eder. Bu olasılığın ne anlama geldiğini anlamak için bazı terimleri tanımlamak gerekmektedir.  $P_{toplam}$ , görüntüdeki toplam piksel sayısı;  $P_{ap}(t)$ ,  $T$  eşiğindeki arka plan piksellerinin sayısı ve  $P_{öp}(t)$ ,  $T$  eşiğindeki ön plan piksellerinin sayısı olsun.  $\omega$  değerleri böylece Denklem (3.14) ve (3.15)'teki gibi hesaplanabilir.

$$\omega_{ap}(t) = \frac{P_{ap}(t)}{P_{toplam}} \quad (3.14)$$

$$\omega_{öp}(t) = \frac{P_{öp}(t)}{P_{toplam}} \quad (3.15)$$

Varyans Denklem (3.16) kullanılarak hesaplanabilir:

$$\sigma^2(t) = \frac{\sum(x_i - \bar{x})^2}{N-1} \quad (3.16)$$

Denklem (3.16)'da yer alan  $x_i$ , ön plan ve arka plan gruplarındaki  $i$  noktasındaki piksel değeri;  $\bar{x}$ , ön plan ve arka plan gruplarındaki piksel değerlerinin ortalaması ve  $N$ , toplam

piksel sayısıdır (Muthukrishnan, 2020). Bu formüller kullanılarak otomatik olarak bir eşik değeri belirlenir ve görüntü bu değere göre eşiklenir (Otsu, 1979).

Sezgin ve Sankur (2004), eşikleme yöntemlerini, yararlandıkları bilgilere göre altı grupta sınıflandırmışlardır. Bu kategoriler şunlardır:

1. Histogram şekline dayalı yöntemler: Düzleştirilmiş histogramın tepe noktalarının, vadilerinin ve eğriliklerinin analiz edildiği yöntemler,
2. Kümeleme tabanlı yöntemler: Gri seviyeli örneklerin arka plan ve ön plan (nesne) olarak iki parça halinde kümelendiği yöntemler,
3. Entropi tabanlı yöntemler: ön plan ve arka plan bölgelerinin entropisini, orijinal ve ikili görüntü arasındaki çapraz entropiyi kullanan algoritmalarla eşiklemeyi gerçekleştiren yöntemler,
4. Nesne özniteliği tabanlı yöntemler: Bulanık şekil benzerliği, kenar çakışması, vb. gibi gri seviye ve ikili görüntüler arasında bir benzerlik ölçüsü arayan yöntemler,
5. Uzamsal yöntemler: Pikseller arasında daha yüksek dereceli olasılık dağılımı ve/veya korelasyon kullanan yöntemler,
6. Yerel yöntemler: Her piksel üzerindeki eşik değerini yerel görüntü özelliklerine uyarlayan yöntemler.

**Görüntü bölütleme:** Görüntü bölütleme, bir görüntüdeki pikselleri sınıflandırmak için kullanılan yöntemlerden biridir. Yöntem, yüksek benzerliğe sahip pikselleri gruplandırır. Özellikle sağlık alanında (röntgen görüntüleri ile) yaygın olarak kullanılmaktadır. Literatürde kenar tabanlı, bölge tabanlı, YSA tabanlı, bulanık teori tabanlı, eşik tabanlı ve kümeleme tabanlı gibi birçok görüntü bölütleme yöntemi mevcuttur (Khan, 2014; Zaitoun ve Aqel, 2015). k-ortalamlar kümeleme ve bulanık C-ortalama kümeleme gibi kümeleme tabanlı görüntü bölütleme yöntemleri, en verimli yöntemlerdendir (Dhanachandra ve diğerleri, 2015). Bu çalışmada kusurların bölütlenmesi için en yaygın kullanılan kümeleme algoritmalarından biri olan (Hartigan, 1975; Hartigan ve Wong, 1979) k-ortalamlar kümeleme algoritması (Berkhin, 2006; Rai ve Singh, 2010) kullanılmıştır.

k-ortalamlar kümeleme yaklaşımı,  $N$  adet veri noktası içeren bir veri kümesini  $k$  adet kümeye böler. Yöntemin çıktısında oluşturulan kümelerin, kümeler içinde maksimum, kümeler arasında minimum benzerliklere sahip olması hedeflenir.  $k$ , bölünecek küme sayısı ve  $\mathbf{X} = \{x_1, x_2, \dots, x_N\}$  veri kümesi olarak her biri  $n$  boyutlu bir vektör olsun.  $\mathbf{A} = \{A_1, A_2, \dots, A_k\}$ ,  $k$  kümelerinin bir dizisi olsun.  $A_j$ 'deki noktaların ortalaması  $\mu_j$ , Denklem (3.17)'deki gibi hesaplanır:

$$\mu_j = \frac{1}{|A_j|} \sum_{x_i \in A_j} x_i \quad (3.17)$$

Burada amaç,  $\mu_j$  değerini kullanarak kayıp fonksiyonunu ( $L$ ) Denklem (3.18)'deki gibi hesaplamaktır:

$$L = \arg \min_{\mathbf{A}} \sum_{j=1}^K \sum_{x \in A_j} \|x - \mu_j\|^2 \quad (3.18)$$

Yöntem, çekirdek (seed) sayılarının atanmasıyla başlar ve iteratif olarak devam eder. Optimal  $k$  değerinin belirlenmesinde en büyük problem çekirdek numarası atamasıdır. Yöntemin adımları aşağıdaki gibidir:

1. Küme merkezleri  $k$  adet küme için rastgele belirlenir (çekirdek sayılar kullanılarak).
2. Her bir veri noktasının küme merkezlerine olan uzaklıkları hesaplanır ve bu veri noktası en yakın kümeye atanır (bu çalışmadaki veri noktaları piksellerdir ve uzaklıklar piksellerin yoğunluk değerleri arasındaki farklardır).
3. Kümelerin merkezi, kümedeki veri noktalarının ortalama değerlerine göre güncellenir.
4. Küme merkez noktaları sabit değilse ikinci ve üçüncü adımlar tekrarlanır.

Yöntemdeki başlangıç küme merkezlerinin atanması genellikle veri noktalarından rastgele seçim ile gerçekleştirilir. Arthur ve Vassilvitskii (2007), k-ortalamlar için küme

merkezlerini seçmek ve merkez seçme probleminden kaçınmak için farklı bir teknik önermektedir.  $c_1$  ve  $c_2$  iki farklı küme merkezi olsun ve  $D(x)$  bir veri noktası ile ona en yakın seçilen merkez arasındaki en kısa mesafeyi göstere. Yöntemin adımları aşağıdaki gibidir:

1.  $\mathbf{X}$ 'ten rastgele bir  $c_1$  merkezi seç.
2.  $\frac{D(x)^2}{\sum_{x \in \mathbf{X}} D(x)^2}$  olasılığıyla  $\mathbf{X}$ 'ten bir veri noktası seçerek yeni bir  $c_i$  merkezi alın.
3. Toplam  $k$  adet merkez seçilene kadar önceki adımı tekrarlayın.
4. Standart  $k$ -ortalamalar algoritması ile devam edin.

Performans açısından  $k$ -means++ algoritması  $k$ -means algoritmasından daha iyi olduğu için (Arthur ve Vassilvitskii, 2007), çalışmada kusur bölütleme aşamasında kullanılmıştır. Ancak birçok kümeleme algoritmasında olduğu gibi burada da en iyi  $k$  değerinin seçimi büyük bir sorundur. Kümeleme algoritmalarında en iyi küme değeri seçimi için kullanılan birçok teknik vardır. En sık kullanılan yöntemlerden üçü aşağıda kısaca açıklanmıştır. Bu yöntemler kullanılarak  $k$  değeri seçimi manuel olarak değil otomatik olarak yapılabilir.

**Dirsek yöntemi:** Küme sayısının bir fonksiyonu olarak varyasyonu çizmek ve kullanmak için eğrinin dirseğini  $k$  olarak seçmek yöntemin temel prosedürüdür. Bir kümenin merkezi ile her kümenin veri noktaları arasındaki mesafenin karesi hesaplanır. Hata karelerinin toplamını (SSE) hesaplamak için her  $k$  değeri için Denklem (3.19) kullanılır:

$$SSE = \sum_{j=1}^k \sum_{x_i \in S_j} \|x_i - C_j\|^2 \quad (3.19)$$

Burada  $k$  küme sayısı,  $x_i$ ,  $i$ . veri noktası,  $S_j$ ,  $j$  kümesi ve  $C_j$ ,  $S_j$ 'nin merkezidir. Her  $k$  değeri için grafikte yer alan SSE değerleri üzerinden bir eğri çizilir. SSE değeri eğri üzerinde ani düşüyorsa, düştüğü noktadaki  $k$  değeri seçilir (Thorndike, 1953).

**Silhouette:** Bu yöntem, kümeler içindeki tutarlılığın yorumlanmasını sağlar ve tutarlılığın doğrulanması için kullanılabilir. Yöntem, her bir veri noktasının ne kadar iyi

sınıflandırıldığını göstermek için de kullanışlıdır. Silhouette yöntemi ile elde edilen değer, bir veri noktasının diğer kümelere kıyasla kendi kümesine ne kadar benzer olduğunun bir göstergesidir. Değer, -1 ile 1 arasında değişir ve 1'e yakın değerler, veri noktasının kendi kümesiyle yüksek düzeyde uyumlu olduğunu ancak komşu kümelerle yüksek düzeyde uyumsuz olduğunu gösterir. Bir kümedeki veri noktalarının çoğu yüksek değere sahipse, kümeleme başarılı olarak kabul edilebilir (Rousseeuw, 1987). Veri noktası  $i$  için Silhouette katsayısı Denklem (3.20)'deki gibi hesaplanabilir:

$$SC(i) = \frac{b(i)-a(i)}{\max \{a(i),b(i)\}} \quad (3.20)$$

Burada  $a(i)$ ,  $i$  veri noktası ile  $i$ 'nin ait olduğu kümedeki diğer tüm veri noktaları arasındaki ortalama mesafedir ve  $b(i)$ ,  $i$  veri noktasından ait olmadığı tüm kümelere olan ortalama mesafedir.

**Davies-Bouldin:** Bu yöntem aynı zamanda Davies-Bouldin indeksi (DBI) olarak da bilinir. Kümeleme algoritmalarında küme sayısını seçmek için kullanılabilir. DBI, her kümenin kendisine en yakın kümeyle ortalama benzerliğini gösterir. Benzerlik ne kadar düşükse, kümeler o kadar iyi ayrılır (Davies ve Bouldin, 1979). Denklem (3.21)'de  $i$  kümesinin dağılımı hesaplanır:

$$S_i = \left\{ \frac{1}{T_i} \sum_{j=1}^{T_i} |X_j - A_i|^q \right\}^{\frac{1}{q}} \quad (3.21)$$

Burada  $i$ , belirli bir tanımlanmış küme,  $T_i$ ,  $i$  kümesindeki gözlemlerin (vektörlerin) sayısı,  $X_j$ ,  $i$  kümesindeki  $j$ 'inci gözlem (vektör) ve  $A_i$ ,  $i$  kümesinin merkezidir. Temel olarak, küme içi dağılımı elde etmek için küme içindeki her bir gözlem ile onun ağırlık merkezi arasındaki ortalama mesafe hesaplanır. Genellikle  $q = 2$  olarak ayarlanır. Bu, kümenin merkezi ile her bir küme vektörü (gözlem) arasındaki öklid mesafesinin hesaplanmasını sağlar.  $i$  ve  $j$  kümeleri arasındaki ayırım Denklem (3.22)'deki gibi hesaplanır:

$$M_{ij} = \left\{ \sum_{k=1}^N |a_{ki} - a_{kj}|^p \right\}^{\frac{1}{p}} \quad (3.22)$$

Burada  $a_{ki}$ ,  $i$  kümesinin merkezi olan  $n$  boyutlu vektör  $a_i$ 'nin  $k$ . bileşenidir.  $M_{ij}$ 'in,  $i$  ve  $j$  kümelerini karakterize eden ağırlık merkezlerinin Minkowski metriği (Friedman ve Rubin, 1967) olduğu belirtilmelidir.  $p = 2$  olarak ayarlandığında Denklem (3.22),  $i$  ve  $j$  kümelerinin merkezleri arasındaki Öklid mesafesini hesaplar. Denklem (3.22) ayrıca aşağıdaki Denklem (3.23)'teki gibi de ifade edilebilir:

$$M_{ij} = \left\{ \sum_{k=1}^N |a_{ki} - a_{kj}|^p \right\}^{\frac{1}{p}} = \|A_i - A_j\|_p \quad (3.23)$$

$S_i$  ve  $M_{ij}$  değerleri kullanılarak  $i$  ve  $j$  kümeleri arasındaki benzerlik Denklem (3.24)'teki gibi hesaplanabilir:

$$R_{ij} = \frac{S_i + S_j}{M_{ij}} \quad (3.24)$$

Burada  $S_i$  ve  $S_j$ , sırasıyla  $i$  ve  $j$  kümelerinde küme içi sapmayı ve  $M_{ij}$ ,  $i$  ve  $j$  kümelerinin merkezleri arasındaki uzaklığı gösterir. Temel olarak burada, kümeler arasındaki benzerlik, ayırma ölçüsüne bölünen iki küme içi sapmanın toplamı olarak hesaplanır.  $R_{ij}$  ne kadar büyükse,  $i$  ve  $j$  kümeleri o kadar benzerdir. Muhtemelen, mümkün olan en iyi durumun, bu sayıların genel olarak mümkün olduğu kadar düşük olması olduğu söylenebilir. Son olarak her bir  $i$  kümesi için en benzer küme Denklem (3.25) ile bulunur.

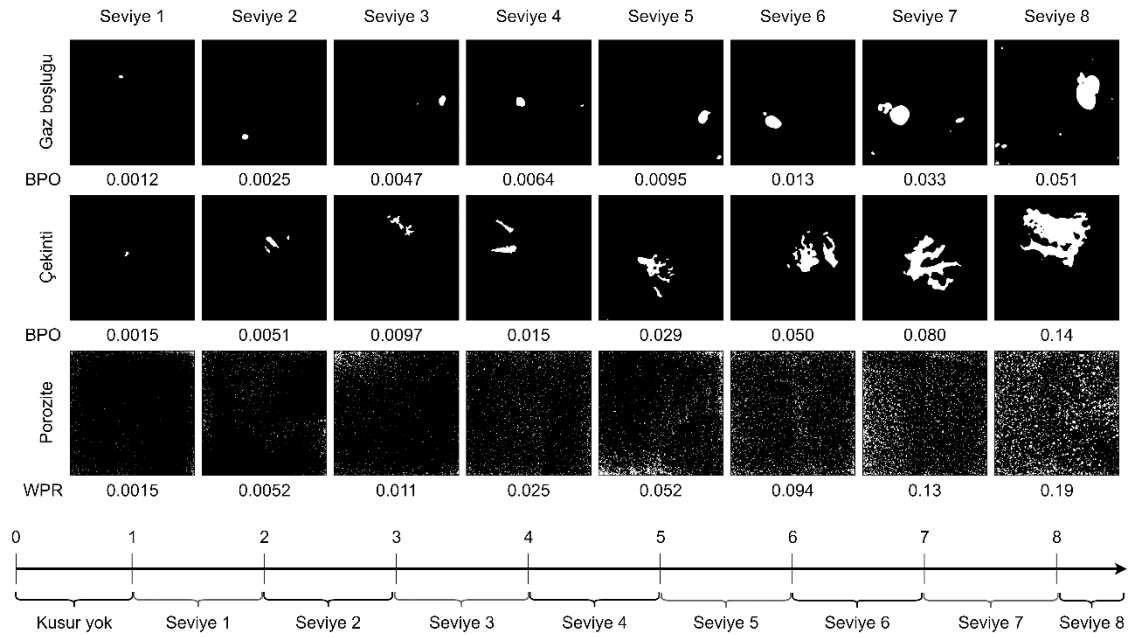
$$R_i \equiv \text{maksimum}(R_{ij}), \quad i \neq j \quad (3.25)$$

Burada her bir  $i$  kümesi için hesaplanan tüm  $R_{ij}$ 'lerden en yüksek olan bulunur (Davies ve Bouldin, 1979).

### 3.2.4. Kusur seviyelerinin belirlenmesi

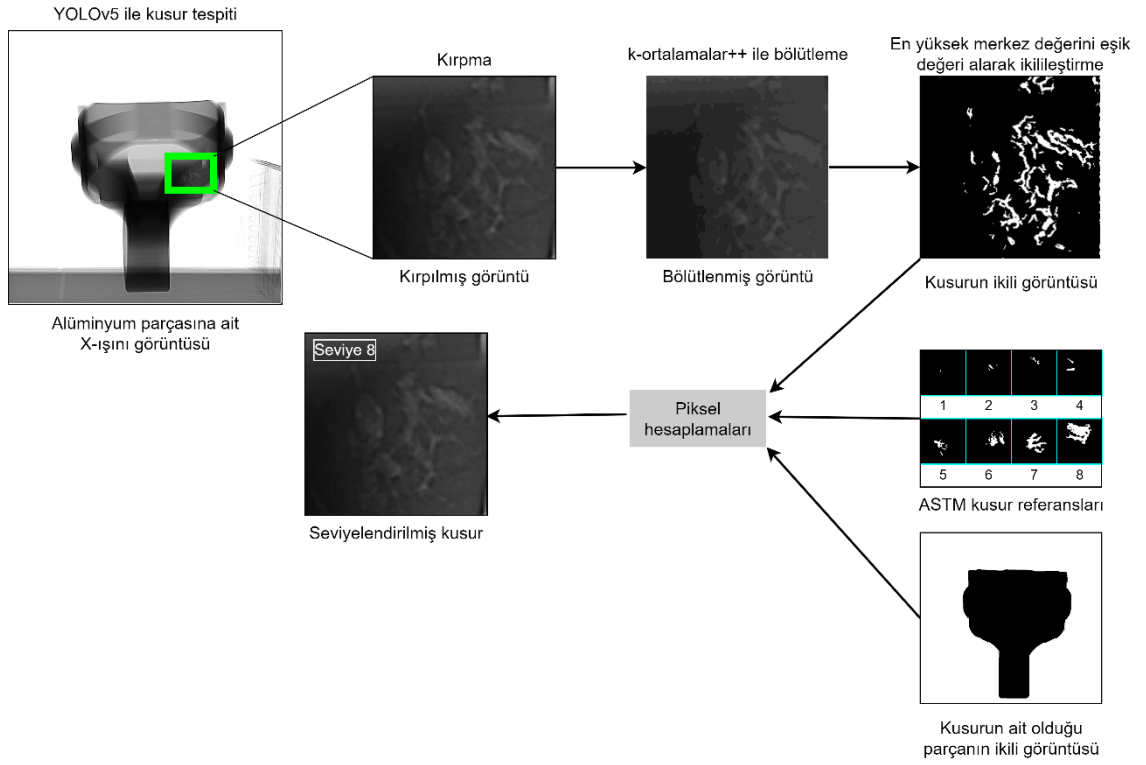
Seviyelendirme işlemi iki farklı bilgi gerektirir: ASTM kusur referansları ve parçadaki kusur alanının parça alanına oranı (piksel bazında). ASTM referanslarında yer alan ve her bir kusur tipi için ayrı ayrı verilen kusur referans bilgileri ve görseller sayesinde parça üzerindeki kusurların seviyeleri belirlenebilir. ASTM referanslarında her bir kusur tipi için sekiz farklı seviye belirtilmiştir.

YOLOv5 dedektörü ile tespit edilen kusurlar ilk olarak orijinal görüntüden kırpılır. İkinci aşama, bölütleme, kusur ayırma ve ardından eşiklemedir. İkili görüntüdeki kusurları temsil eden piksel sayısı hesaplanır. Kusur piksellerinin sayısı ile daha önceden hesaplanan kusurlu parçanın tamamının piksel sayısı oranlanarak parçadaki kusurlu bölgenin oranı elde edilir. Bu oran ASTM'deki kusur seviyelerinde ilgili aralığa (daha önceden ikili görüntüye çevirilerek beyaz piksellerin oranı (BPO) hesaplanmıştır) atanarak kusurun seviyesi belirlenir (Şekil 3.30). İşlem, Şekil 3.31'da gösterilmektedir.



Şekil 3.30. ASTM kusur seviyelerine ait seviye aralıkları (ASTM International, 2014)





**Şekil 3.31.** Kusur seviyelendirme süreci

Kusur seviyelendirme sürecinin ilk adımı YOLOv5 dedektörü tarafından tespit edilen kusurun konum bilgileri kullanılarak ana görüntüden kırılmasıdır (Bölüm 3.2.3). Bu işlemin yapılmasının sebebi hem daha küçük boyutlu bir görüntü üzerinden kısa sürede işlem yapmak hem de diğer gereksiz alanların eşikleme aşamasında çıkaracağı sorunları yok etmektir. Kırılan görüntüler k-ortalamlar++ algoritması ile bölütlenir (Denklem (3.17) ve (3.18)). İlgili  $k$  değeri seçim algoritması ile en uygun  $k$  değeri üzerinden bu bölütleme işlemi gerçekleştirilir. Daha sonra bölütlenmiş görüntü üzerinden küme merkez bilgileri kullanılarak en yüksek piksel değerine sahip küme merkezi değeri kullanılarak eşikleme işlemi gerçekleştirilir ve görüntü ikili bir görüntüye dönüştürülür. YOLOv5 dedektöründe girdi olarak kullanılan ilgili kusura ait orijinal görüntü de aynı şekilde eşiklenir. Böylece biri yalnızca kusuru gösteren küçük bir görüntü ile orijinal parçanın tamamına ait görüntünün ikili görüntüleri elde edilmiş olunur. Kusur görüntüsü için kusurlu piksellerin sayısı ve orijinal görüntüdeki parçanın parçaya ait piksellerinin sayısı ikilileştirme sonrasında kolayca hesaplanır. İki değer oranlanarak parçadaki kusurun oranı 2B temelinde bulunmuş olur. Bu oran ilgili kusura ait ASTM referans kusur aralıklarındaki uygun aralığa atanarak kusurun seviyesi belirlenir (Şekil 3.31).

### 3.2.5. Performans metrikleri

Nesne tespit yöntemlerini değerlendirmek için ortalama kesinlik (average precision) ( $AP$ ),  $AP$ 'lerin ortalaması (mean average precision) ( $mAP$ ), kesinlik (Precision) ( $Pr$ ), hatırlama (Recall) ( $Re$ ) ve  $F_1$  skoru gibi bazı metrikler kullanılır. Böylece yöntemin performansı değerlendirilebilir ve analiz edilebilir. Bu metrikler sayesinde nesne tespit yöntemleri arasında karşılaştırmalar yapılır. Bu metrikleri hesaplamak için bazı terimlerin kullanılması gerekir. Nesne tespiti sonucunda elde edilen sınırlayıcı kutunun başarılı bir tahmin olup olmadığını değerlendirilebilmesi için tahmin edilen sınırlayıcı kutu ile temel-gerçek kutusu arasındaki birleşim üzerinden kesişim değerini gösteren  $IoU_p$  değeri ile eşik değeri olarak kullanılan  $IoU_t$  değerinin karşılaştırılması gerekmektedir.  $IoU_t$ , birleşim üzerinden kesişim hesaplandıktan sonra (Şekil 3.15) eşik değeri olarak kullanılır. Bu değer tahmin edilen sınırlayıcı kutunun başarılı bir tahmin olarak değerlendirilebilmesi için gereken en alt seviyedir.  $IoU_p \geq IoU_t$  ise, nesnenin sınırlayıcı kutunun içinde olduğuna karar verilir.  $IoU$  hesabı denklem (3.26)'da verilmiştir. Denklem (3.26)'da yer alan SK terimi sınırlayıcı kutu ve TG terimi temel-gerçek kutusunu temsil etmektedir. (Bölüm 3.2.2.'de detaylı olarak anlatılmıştır).

$$IoU_p = \frac{alan(SK \cap TG)}{alan(SK \cup TG)} \quad (3.26)$$

$IoU_p$  kullanılarak, doğru pozitifler (TP), yanlış pozitifler (FP) ve yanlış negatifler (FN) hesaplanır. TP doğru tespiti ifade ederken ( $IoU_p \geq IoU_t$ ), FP var olmayan bir nesnenin tespit edilmesi veya mevcut bir nesnenin yanlış bir tespittir (lokalizasyon hatası) ( $IoU_p < IoU_t$ ). FN, tespit edilememiş bir temel-gerçek kutusuna karşılık gelir. Bir nesne tespit edilemezse FN olarak ifade edilir. Doğru negatifler (TN) nesne tespitinde kullanılmaz (Nesne içermeyen sonsuz sayıda sınırlayıcı kutu olduğu için). Bu terimler kullanılarak,  $Pr$  ve  $Re$  sırasıyla aşağıdaki Denklem (3.27) ve (3.28)'deki gibi hesaplanabilir:

$$Pr = \frac{TP}{TP + FP} \quad (3.27)$$

$$Re = \frac{TP}{TP + FN} \quad (3.28)$$

Başka bir performans metriği olan  $F_1$  skoru,  $Pr$  ve  $Re$  kullanılarak, (kesinlik ve hatırlamanın harmonik ortalaması) Denklem (3.29)'daki gibi hesaplanabilir:

$$F_1 = 2 \times \frac{Pr \times Re}{Pr + Re} \quad (3.29)$$

$AP$  metriği her sınıf için ayrı ayrı hesaplanır. Bu metrik,  $Pr-Re$  eğrisinin altında kalan alana eşittir ve Denklem (3.30) ile hesaplanabilir:

$$AP = \int_0^1 Pr(Re) \cdot dPr \quad (3.30)$$

Son olarak tüm sınıflar için elde edilen  $AP$  değerlerinin ortalaması hesaplanarak çok sınıflı nesne algılama yöntemlerinde en yaygın kullanılan metrik olan  $mAP$  değeri Denklem (3.31)'deki gibi elde edilir:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3.31)$$

$N$ , veri kümesindeki sınıf sayısıdır (çalışmamızda üç kusur sınıfı vardır: Gaz boşluğu, çekinti ve porozite). Bu metriklerin yanı sıra ikili değerlere dönüştürülen kusur görüntüleri üzerinde eşikleme yöntemlerini karşılaştırmak amacıyla kullanılan bir başka metrik tanımlanması gerekmektedir. Çalışmada ikilileştirme performansını ölçmek için  $IoU$  performans metriği (Jaccard endeksi olarak da bilinir) kullanılmıştır. Tıpkı nesne

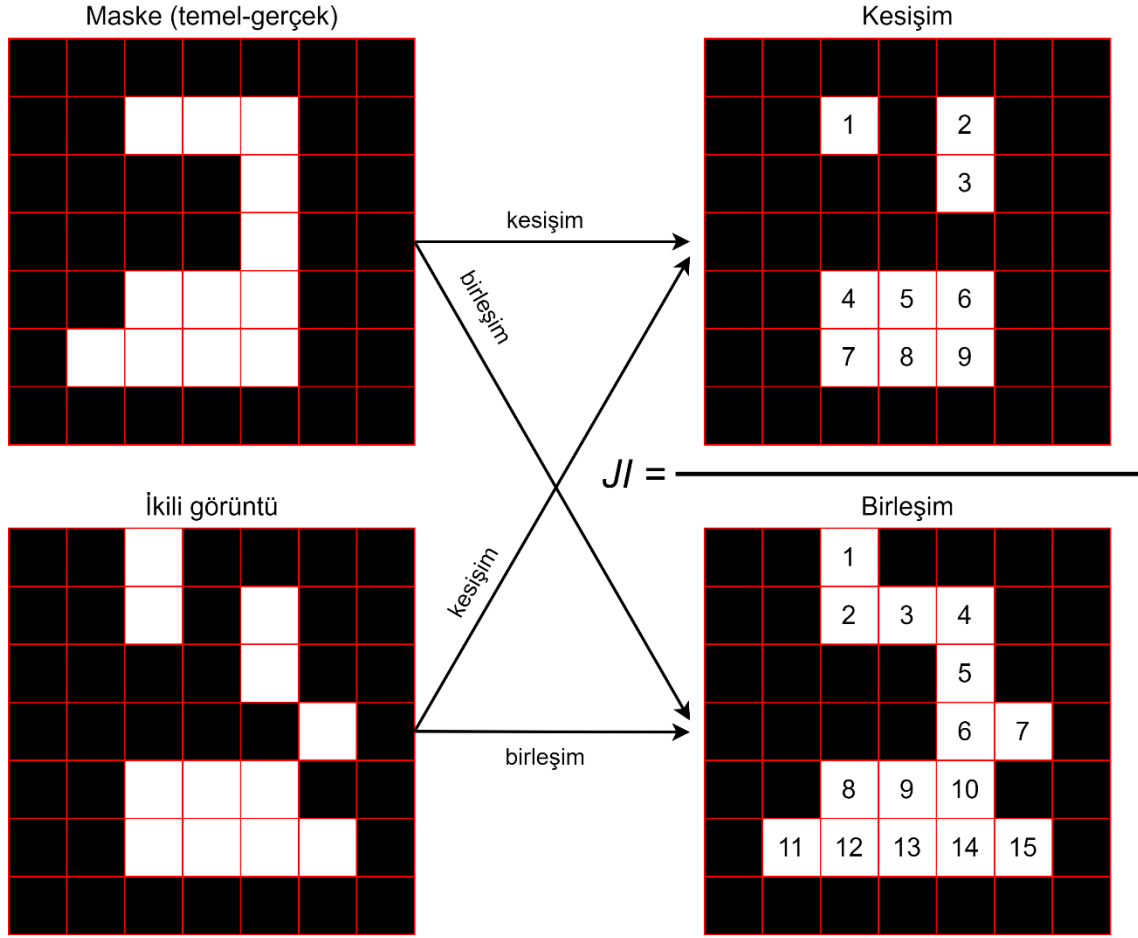
tespitinde olduğu gibi performans, birleşim üzerindeki kesişim değerleriyle ölçülmüştür. Nesne tespitinin aksine, kutular yerine bireysel olarak pikseller kullanılmıştır. Test için hazırlanan parçalı görüntüler yani maskeler (nesne tespitinde bu temel-gerçek olarak isimlendirilmiştir) ile ikilileştirme sonrası elde edilen görüntüler karşılaştırılmıştır. Karışıklığı önlemek için nesne algılama aşamasında *IoU* yerine Jaccard endeksinin kısaltılmış hali olan *Jl* kullanılmıştır.

Şekil 3.32’de örnek bir *Jl* değeri hesabı gösterilmiştir. Sol üst kısımda yer alan ikili görüntü kusur görüntüsüne ait elle oluşturulmuş bir temel-gerçektir (burada maske olarak kullanılmaktadır). Sol alttaki şekil ikilileştirilmiş olan kusur görüntüsüdür. Bu iki şekle ait pikseller kullanılarak iki görüntünün kesişimine ve birleşimine ait şekiller sırasıyla sağ üstte ve sağ altta gösterilmiştir. Beyaz olan pikseller kusuru temsil ederken, siyah pikseller kusuru içermeyen ve arka plan olarak değerlendirilen piksellerdir. Kesişim sonucunda elde edilen beyaz piksel sayısı 9, birleşim sonucunda elde edilen beyaz piksel sayısı ise 15’tir. Bu iki değerden bölünür elde edilen değer ile *Jl* değeri bulunur ( $Jl = \frac{9}{15} = 0,6$ ). Elde edilen bu değer daha önceden belirlenmiş *Jl* eşik değeri yani  $Jl_t$ ’den büyük veya eşit ise ikilileştirme işleminin başarılı olduğu kabul edilir.

### 3.2.6. Döküm parametrelerinin optimizasyonu

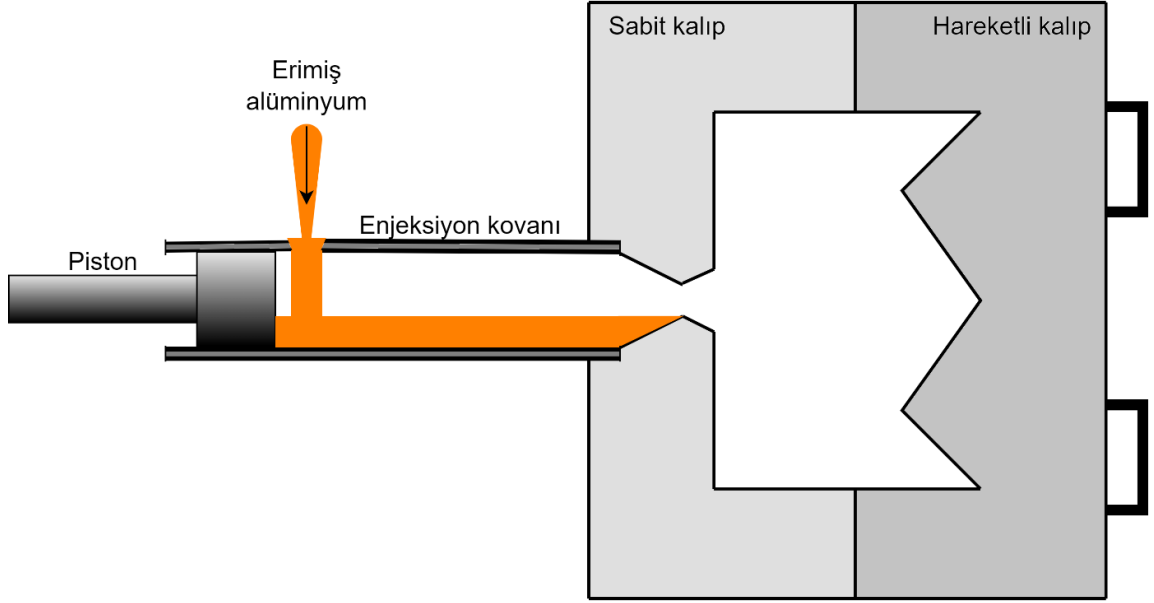
Kusur oluşumu, maliyetli hurda kaybına neden olur ve kritik ve yüksek mukavemet gerektiren uygulamalarda döküm parçaların kullanımını sınırlar. Kusur miktarı, döküm süreç parametreleriyle yakından ilişkilidir (Tsoukalas ve diğerleri, 2004). Basınçlı döküm işlemi, erimiş metalin kalıcı metal kalıplara üç aşamada doldurulmasından oluşur: Birinci veya düşük hız aşaması, ikinci veya yüksek hız aşaması ve kusur oluşumunu engellemek için hidrolik basıncın uygulandığı üçüncü veya yoğunlaştırma aşaması. Yüksek basınçlı döküm kalıbı, sabit yarım ve hareketli yarım olmak üzere iki temel parçadan oluşur. Kalıp açıldığında, döküm, hidrolik veya mekanik olarak etkinleştirilen pimler tarafından hareketli olan yarıda tutulur. Şekil 3.33’te basınçlı döküm cihazıyla döküm parçası üretim sürecinin basit bir gösterimi verilmiştir. Burada ilk olarak erimiş alüminyum, pota aracılığıyla enjeksiyon kovanına dökülür. Belirli bir stabilizasyon süresinden sonra erimiş alüminyum piston tarafından kalıba enjekte edilir ve kalıp içinde soğuyarak katılaşmaya

başlar. Katılma bitene kadar basınç korunur. Katıldıktan sonra kalıbın hareketli kısmı açılır ve ürün yarım kalıptan çıkarılır.

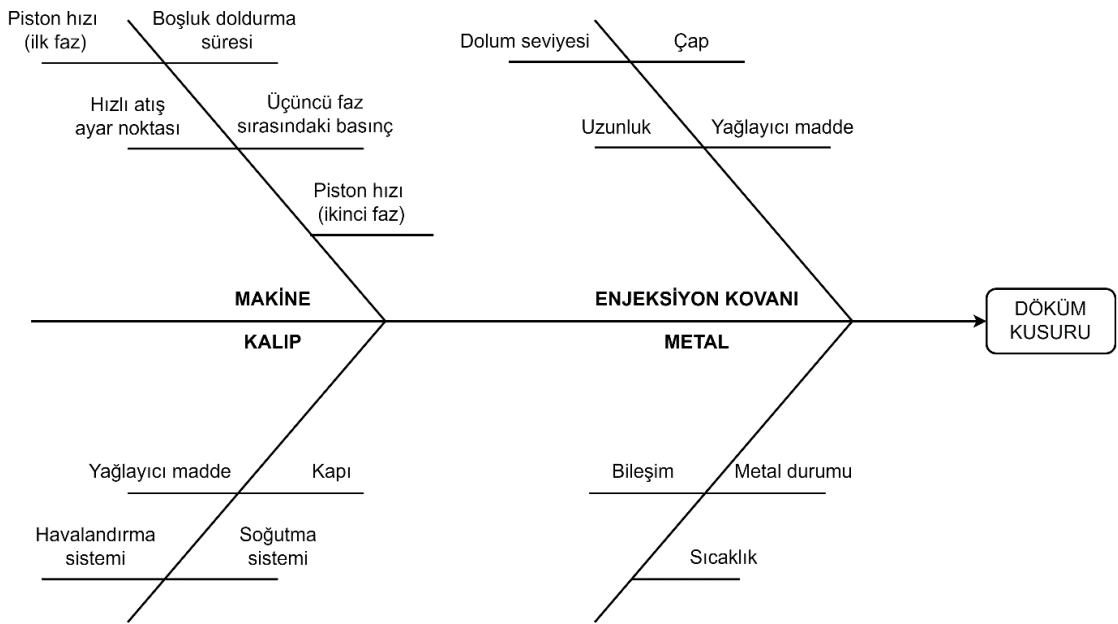


**Şekil 3.32.**  $JI$  değerinin hesaplanmasını gösteren bir örnek

Basınçlı dökümde kusur oluşumu, çok sayıda süreç kontrol parametresinin aldığı değerlere bağlıdır. Şekil 3.34’te kusur oluşumunu etkileyebilecek döküm süreci parametrelerini belirlemek için oluşturulmuş Ishikawa diyagramı yer almaktadır. Literatürde yer alan çalışmalar incelendiğinde genellikle bekletme fırını sıcaklığı, kalıp sıcaklığı, enjeksiyon kovanı dolum noktası, birinci faz hızı, ikinci faz hızı ve üçüncü faz basıncı en önemli parametreler olarak belirtilmiştir (Syracos, 2003; Tsoukalas ve diğerleri, 2004).



Şekil 3.33. Enjeksiyonlu döküm bileşenleri



Şekil 3.34. Döküm süreci parametrelerinin Ishikawa diyagramı (Syracos, 2003)

Bu parametrelerin farklı değerleri ile yapılan deneyler sonucu, döküm parçalarında kusur oluşumuna sebep olan etkin parametreleri saptamak mümkündür (Syracos, 2003). Bu parametrelerin kusur oluşumu üzerindeki doğrudan etkisini anlamak, kusurun kaynağını

belirlemek anlamına gelebilir. Kusur oluşumunu etkileyen parametrik faktörleri bulmak için farklı deneysel tasarımlarla ve yöntemlerle anlamlı çıkarımlar yapılabilir. Faktörlerin çıktı üzerindeki etkisini incelemek istatistiksel yöntemlerden faydalanmak gerekir. Varyans analizi (ANOVA) iki veya daha fazla grubun ortalamalarının birbirinden önemli ölçüde farklı olup olmadığını kontrol etmek için kullanılan istatistiksel bir tekniktir. ANOVA, farklı örnekleri karşılaştırarak bir veya daha fazla faktörün etkisini kontrol eder. Çizelge 3.1’de bir ANOVA tablosu yer almaktadır. Tek bir örnek olarak bakıldığında, gözlemler arasındaki toplam değişkenlik miktarı, her bir  $x_{ij}$  ile  $\bar{x}$  arasındaki farkların karelerini toplayarak ölçülebilir ve bu Denklem (3.32)’deki  $SST$  değerini yani kareler toplamını verir:

$$SST = \sum_{i=1}^k \sum_{j=1}^{n_i} (x_{ij} - \bar{x})^2 \quad (3.32)$$

**Çizelge 3.1.** ANOVA tablosu

Kaynak	Kareler toplamı (SS)	Serbestlik derecesi (df)	Kareler ortalaması (MS)	$F$	$p$ -değeri
Gruplar arası	$SSM$	$k - 1$	$MSM = \frac{SSM}{k - 1}$	$\frac{MSM}{MSE}$	$P$
Hata/Grup için	$SSE$	$n - k$	$MSE = \frac{SSE}{n - k}$		
Toplam	$SST$	$n - 1$			

Bu toplam değişkenlik miktarı iki ana kaynaktan gelmektedir. Denklem (3.33) grup ortalamaları arasındaki değişkenliği ( $SSM$ ) ve Denklem (3.34) hatadan ya da grup içinden kaynaklanan değişkenliği ( $SSE$ ) göstermektedir.

$$SSM = \sum_{i=1}^k n_i (\bar{x}_i - \bar{x})^2 \quad (3.33)$$

$$SSE = \sum_{i=1}^k \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2 \quad (3.34)$$

Ayrıca yalnızca  $SSM$  ve  $SSE$ , Denklem (3.35)'teki gibi hesaplanarak da  $SST$  değeri bulunabilir:

$$SST = SSM + SSE \quad (3.35)$$

Burada  $k$ , grup/popülasyon/açıklayıcı değişken/seviye değerlerinin sayısı;  $n$ , toplam örneklem sayısı;  $n_i$ ,  $i$  grubundan alınan örneklem sayısı;  $x_{ij}$ ,  $i$  grubundan örneklenen  $j$  yanıtı;  $\bar{x}_i$ ,  $i$  grubundan alınan yanıtların ortalaması ( $\frac{1}{n_i} \sum_{j=1}^{n_i} x_{ij}$ ) ve son olarak  $\bar{x}$ , tüm yanıtların ortalamasıdır ( $\frac{1}{n} \sum_{ij} x_{ij}$ ).

ANOVA hipotezi aşağıdaki gibi olsun:

$H_0$ : İncelenen tüm grupların ortalamaları eşittir.

$H_1$ : Ortalamaların hepsi eşit değildir.

$H_0$  hipotezi doğruysa,  $F$  istatistiği pay ve paydada sırasıyla  $k - 1$  ve  $n - k$  serbestlik derecesine sahip bir  $F$  dağılımına sahiptir.  $H_1$  hipotezi doğruysa, o zaman  $F$  büyük olma eğilimindedir.  $F$  istatistiği yeterince büyükse  $H_0$ ,  $H_1$  lehine reddedilir.



## 4. BULGULAR ve TARTIŞMA

### 4.1. Bulgular

Bu bölümde sırasıyla nesne tespit yöntemlerinin kusur tespitindeki sonuçları ve performansları ile bölütleme ve ikilileştirme sonuçları verilmiştir. Daha sonra tek tip bir parça üzerinden döküm süreci parametrelerinin bazı farklı değerleri için yapılan deneyler kusur oranı çıktısı üzerinden yorumlanmıştır. Bu deney sonuçları kullanılarak döküm süreci parametreleri için en uygun değerler belirlenmiştir.

#### 4.1.1. Ön çalışmalar (nesne tespiti için en uygun dedektörün belirlenmesi)

Ana çalışmaya geçmeden önce yapılan ön çalışma ile kusurların tespiti için en uygun nesne tespit yöntemine karar verilmiştir (Parlak ve Emel, 2023). Burada yalnızca iki tip kusura ait (gaz boşluğu ve çekinti) veri seti kullanılmıştır. Veri setinde toplam 3466 adet YBD tekniği ile üretilen alüminyum döküm parçalarına ait X-ışını görüntüsü yer almaktadır (Al-Cast veri seti)<sup>4</sup>. Bu görüntülerin yaklaşık %10'u kusur içermemektedir. Çalışmada kullanılan nesne dedektörlerini eğitmek için tüm görüntüler, dökümhanede muayene süreci içinde yer alan uzmanlardan alınan bilgiler ve yardımlarla etiketlenmiştir.

Veri seti, eğitim için %70, validasyon için %20 ve test için %10 olarak ayrılmıştır. Buradaki amaç, testler için en iyi performans gösteren yöntemi seçmektir. YOLOv3 (SPP ve Tiny), Scaled YOLOv4-CSP, YOLOv4, YOLOv5 (“n”, “s”, “m”, “l”, “x”) (bu notasyon ağın derinliğine göre yazılmıştır ve harfler sırasıyla nano, small, medium, large ve xlarge kelimelerini ifade etmektedir), YOLOR, RetinaNet ve EfficientDet nesne tespit yöntemleri ayrı ayrı eğitilmiştir. Makul eğitim süreleri için omurga olarak en az sayıda parametreye sahip olduğundan, bu çalışmada RetinaNet için ResNet50 ve EfficientDet için EfficientNet-D0 kullanılmıştır. Karşılaştırmalar için beş temel performans metriği

---

<sup>4</sup> <https://drive.google.com/drive/folders/1e5kPSc1RuapBtF5zMW5HhoTYVjttjdJ7?usp=sharing>

kullanılmıştır:  $mAP$ ,  $AP_c$  (çekinti sınıfının ortalama kesinliği),  $AP_g$  (gaz boşluğu sınıfının ortalama kesinliği),  $Pr$ ,  $Re$  ve  $F_1$ . Tüm yöntemler için  $IoU_t = 0,25$  olarak alınmıştır.

YOLOv3, YOLOv4, YOLOv5, YOLOR, RetinaNet ve EfficientDet kodları Github'tan sırasıyla ultralytics<sup>5</sup>, AlexeyAB<sup>6</sup>, ultralytics<sup>7</sup>, WongKinYiu<sup>8</sup>, fizyr<sup>9</sup> ve auttml<sup>10</sup>'den alınmış ve bu çalışmaya uyarlanmıştır. Eğitim işlemi CPU'larda çok uzun sürdüğünden tüm işlemler için GPU kullanımı gerekmektedir. Eğitim ve test için GPU'ya (Tesla P100-PCIE-16GB GPU computing processor, sürüm: 460.32.03) ve CUDA'ya, (sürüm: 11.2) erişmek adına Google Colab<sup>11</sup> ortamında 26 GB RAM ve 2.20 GHz'lik iki Intel Xeon CPU donanımlı bilgisayar kullanımı ile Python dilinde tüm modern nesne tespit yöntemleri eğitilmiştir.

Çizelge 4.1'de derin öğrenme tabanlı nesne dedektörlerinin performansları yer almaktadır. Buradaki performans metrikleri Denklem (3.26)-(3.31) kullanılarak elde edilmiştir (3.2.5. numaralı bölüm). Tüm eğitimlerde mini-yığın boyutu 16 ve öğrenme oranı 0,01 olarak ayarlanmış olup, eğitimler 100 epoch boyunca sürdürülmüştür. Çizelgeye bakıldığında en yüksek  $mAP$  değerlerinin YOLOv5 modelleriyle elde edildiği görülmektedir. Tüm YOLOv5 modellerinde  $mAP$ , 0,94'ün üzerinde elde edilmiştir. Daha özel olarak YOLOv5m modeli eğitim süresi dışındaki tüm metriklerde en yüksek performansı göstermiştir. Bunun yanı sıra EfficientNet ve RetinaNet dedektörleri performans açısından diğer tüm yöntemlerin gerisinde kalmıştır. Bunun sebebi büyük nesnelere için yüksek tespit başarısı gösteren bu yöntemlerin, küçük kusurlar gibi görüntü içinde görülmesi zor olan nesnelere tespit etmedeki başarısızlığıdır. Ayrıca eğitim süresi açısından bakıldığında YOLOv5n modeli bir saatin altında eğitim süresi ile en iyi eğitim süresi performansını ortaya koymuştur. Sonraki adımlarda en yüksek  $mAP$  değeri veren YOLOv5 ve en kısa sürede eğitilen YOLOv5n test görüntüleri üzerinden kullanılmış ve karşılaştırılmıştır (Parlak ve Emel, 2023).

---

<sup>5</sup> <https://github.com/ultralytics/yolov3>

<sup>6</sup> <https://github.com/AlexeyAB>

<sup>7</sup> <https://github.com/ultralytics/yolov5>

<sup>8</sup> <https://github.com/WongKinYiu/yolor>

<sup>9</sup> <https://github.com/fizyr>

<sup>10</sup> <https://github.com/google/auttml>

<sup>11</sup> <https://colab.research.google.com>

**Çizelge 4.1.** Nesne dedektörlerinin performansları ( $IoU_t = 0,25$ )

Dedektör	$mAP$	$AP_c$	$AP_g$	$Pr$	$Re$	$F_1$	Eğitim süresi [saat]
YOLOv3-SPP	0,927	0,920	0,933	0,915	0,883	0,900	6,65
YOLOv3-tiny	0,844	0,843	0,846	0,890	0,760	0,820	1,03
YOLOv4	0,918	0,931	0,906	0,900	0,900	0,900	6,85
Scaled-YOLOv4-CSP	0,826	0,887	0,767	0,880	0,800	0,840	5,15
YOLOv5n	0,941	0,953	0,930	0,920	0,891	0,905	<b>0,75</b>
YOLOv5s	0,950	0,955	0,944	0,932	0,915	0,923	1,24
YOLOv5m	<b>0,956</b>	<b>0,962</b>	<b>0,950</b>	<b>0,944</b>	<b>0,922</b>	<b>0,933</b>	2,57
YOLOv5l	0,945	0,945	0,945	0,932	0,904	0,918	4,64
YOLOv5x	0,941	0,948	0,935	0,942	0,898	0,920	8,58
YOLOv5	0,826	0,841	0,814	0,657	0,815	0,728	3,01
EfficientDet	0,614	0,645	0,602	0,564	0,718	0,632	3,72
RetinaNet	0,723	0,746	0,711	0,720	0,712	0,716	2,54

YOLOv5m ve YOLOv5n modelleriyle yapılan testlerde 348 görüntüde yer alan 617 adet kusurun doğru tespit oranı hesaplanmıştır. Burada da YOLOv5m kusurların %95,9'unu tespit ederek, en yüksek performansı gösteren model olmuştur. Bunun yanı sıra YOLOv5n de %93,9'luk doğru tespit yüzdesiyle oldukça başarılı bir şekilde tespit işlemlerini gerçekleştirmiştir. Genel olarak YOLOv5 dedektörlerinin iki tip kusur tespitinde ortaya çıkarılan başarısından dolayı üç tip kusur içeren tespit deneylerinde YOLOv5 modelleri kullanılmıştır.

#### 4.1.2. YOLOv5 dedektörleriyle kusur tespiti

Ön çalışmalarda yapılan birçok modern nesne tespit yönteminin eğitim ve test sonuçlarının ardından kusur tespiti için YOLOv5 modellerinin (“n”, “s”, “m”, “l” ve “x”) kullanılmasına karar verilmiştir. Ön çalışmalardaki görüntülerde iki tip kusur (gaz boşluğu ve çekinti) varken bu kez bu veri setine porozite kusurları da dahil edilerek tespit deneyleri gerçekleştirilmiştir. Kullanılan veri seti (Al-Cast2) yer alan görüntülerin %10'u

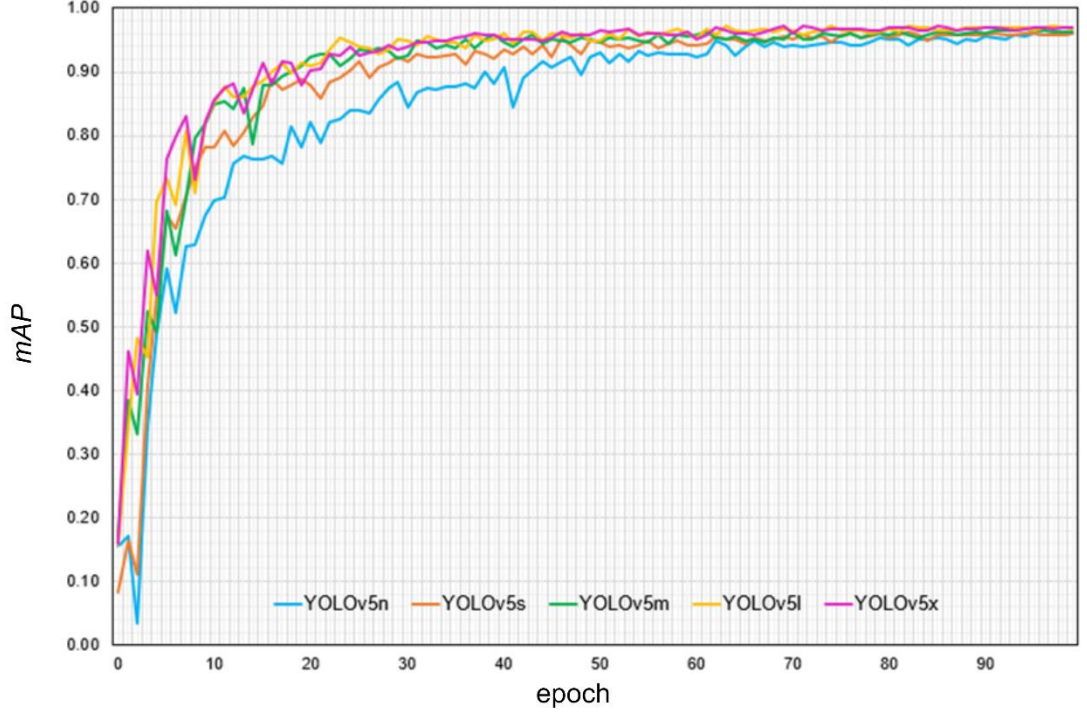
herhangi bir kusur içermemektedir. Tıpkı bir önceki çalışmada olduğu gibi beş model de mini-yığın boyutu 16, öğrenme oranı 0,01 ve epoch sayısı 100 seçilerek eğitilmiştir. Eğitimler yine aynı şekilde Google Colab ortamında aynı donanım ve yazılım şartlarında gerçekleştirilmiştir. Çizelge 4.2’de YOLOv5’in beş farklı modeline ait eğitim performansları verilmiştir. Burada  $AP_c$ ,  $AP_g$  ve  $AP_p$  metrikleri sırasıyla çekinti, gaz boşluğu ve porozite kusur sınıflarının ortalama kesinlik değerleridir ve yine Denklem (3.30) kullanılarak hesaplanmıştır.  $mAP$  değeri açısından en yüksek performansı YOLOv5l göstermiştir ( $mAP = 0,971$ ). En düşük performans YOLOv5n ve YOLOv5s ile ortaya çıkmasına rağmen elde edilen  $mAP$  değeri ( $mAP = 0,962$ ), YOLOv5l modeliyle elde edilen  $mAP$  değerine oldukça yakındır. Bunun yanı sıra üç farklı kusur sınıfında çekinti dışındaki tüm sınıflarda yine aynı şekilde YOLOv5l modeli en yüksek ortalama kesinlik değerini elde ederken çekinti sınıfından en yüksek ortalama kesinlik değeri YOLOv5x tarafından elde edilmiştir. Kesinlik metriğinde en yüksek başarı YOLOv5s ile elde edilirken, hatırlama metriğinde YOLOv5m modeli en yüksek performansı ortaya koymuştur. Ayrıca YOLOv5s ve YOLOv5m  $F_1$  skorunda 0,952’lik bir değerle ilk sırada yer almıştır. Eğitim süresine bakıldığında en küçük model olan YOLOv5n doğal olarak en kısa sürede eğitimi tamamlamıştır.

**Çizelge 4.2.** YOLOv5 yöntemlerinin kusur tespitindeki eğitim performansları

Model	$mAP$	$AP_c$	$AP_g$	$AP_p$	$Pr$	$Re$	$F_1$	Eğitim süresi [saat]
YOLOv5n	0,962	0,960	0,978	0,947	0,925	0,963	0,943	<b>1,52</b>
YOLOv5s	0,962	0,955	0,982	0,949	<b>0,937</b>	0,968	<b>0,952</b>	2,51
YOLOv5m	0,967	0,967	0,985	0,950	0,932	<b>0,973</b>	<b>0,952</b>	3,07
YOLOv5l	<b>0,971</b>	0,966	<b>0,991</b>	<b>0,956</b>	0,933	0,969	0,951	9,19
YOLOv5x	0,969	<b>0,971</b>	0,983	0,953	0,924	0,971	0,947	17,89

Şekil 4.1’de beş farklı YOLOv5 modelinin 100 epoch boyunca değişen  $mAP$  değerleri gösterilmiştir. Grafiğe bakıldığında YOLOv5x modelinin 0,80’lik  $mAP$  değerine oldukça kısa sürede geldiği görülmektedir (yaklaşık on epoch). Eğitim aşamasında  $mAP$  bazında

en iyi performans gösteren YOLOv5l ve süre bakımından en kısa sürede eğitilen YOLOv5n test aşaması için seçilmiştir.



**Şekil 4.1.** YOLOv5 modellerinin 100 epoch boyunca değişen mAP değerleri

Şekil 4.2’de YOLOv5n ve YOLOv5l modellerinin validasyon aşaması sonunda oluşan karışıklık matrisleri verilmiştir. Sütunlar gerçekte sınıfın ne olduğunu gösterirken, satırlar tahmin edilen sınıfı göstermektedir. Şekil 4.2’de *arka plan* olarak yeni bir sınıf tanımının yapıldığı görülmektedir. Normalde veri seti içinde yer almayan *arka plan* sınıfı çıktıda üç sınıfın dışında olan her şey için kullanılmaktadır. Yöntemler karmaşıklık matrisi açısından karşılaştırıldığında kusurların doğru tahmini konusunda her üç kusurda da YOLOv5l yöntemi daha iyidir. Şekil 4.2’de yer alan matrislerdeki değerler Denklem (3.27) kullanılarak elde edilmiştir.

		Gerçek			
		Gaz boşluğu	Çekinti	Porozite	Arka plan
Tahmin edilen	Gaz boşluğu	0,96	0	0	0,24
	Çekinti	0	0,97	0,02	0,51
	Porozite	0	0,01	0,96	0,25
	Arka plan	0,04	0,02	0,02	

YOLOv5n

		Gerçek			
		Gaz boşluğu	Çekinti	Porozite	Arka plan
Tahmin edilen	Gaz boşluğu	0,97	0	0	0,22
	Çekinti	0	0,98	0	0,50
	Porozite	0	0	0,97	0,28
	Arka plan	0,03	0,02	0,03	

YOLOv5l

**Şekil 4.2.** Denklem (3.27)'ye göre karışıklık matrisleri

Çizelge 4.3, her iki modelin test sonuçlarını göstermektedir. Test aşamasında kullanılan modellerin performansları ön çalışmalardan farklı olarak tüm test görüntüleri etiketlenerek temel-gerçek kutuları kullanılarak ölçülmüştür. Testler yaklaşık 500 görüntü üzerinden gerçekleştirilmiştir. Çizelge 4.3'te yer alan sonuçlara göre YOLOv5l modeli, tespit doğruluğu açısından tüm kusur sınıfları için  $AP$  metriklerinde en yüksek performansı göstermiştir. Ayrıca YOLOv5n, gaz boşluğu ve çekinti sınıfları için 0,90'ın üzerinde bir  $AP$  performansı gösterirken, porozite sınıfı için  $AP$  değeri 0,90'ın altında kalmıştır ( $AP_p = 0,892$ ). Her iki model de çekinti kusurlarında en yüksek tespit doğruluğunu elde ederken, en düşük başarı porozite sınıfında ortaya çıkmıştır. Tespit hızı açısından ise YOLOv5l modelinin bir görüntü üzerinde kusur tespit etmesi yaklaşık 29 milisaniye sürerken ( $FPS = 34$ ), YOLOv5n modelinde bu süre 7 milisaniyeye düşürülüyor ki bu da modelin saniyede yaklaşık 143 görüntü tespit edebildiğini göstermektedir ( $FPS = 143$ ). Son olarak Şekil 4.3 ve 4.4'te sırasıyla YOLOv5n ve YOLOv5l modelinin bazı test görüntüleri üzerindeki tespitleri gösterilmiştir.

Kusurların başarılı bir şekilde tespit edilmesinden sonra, tez çalışmasının diğer aşamalarına geçilmiştir. Bu aşamalarda kullanılacak görüntüler YOLOv5l modelinin test görüntülerindeki tespit işlemi sonucunda alınan çıktılarından elde edilmiştir.

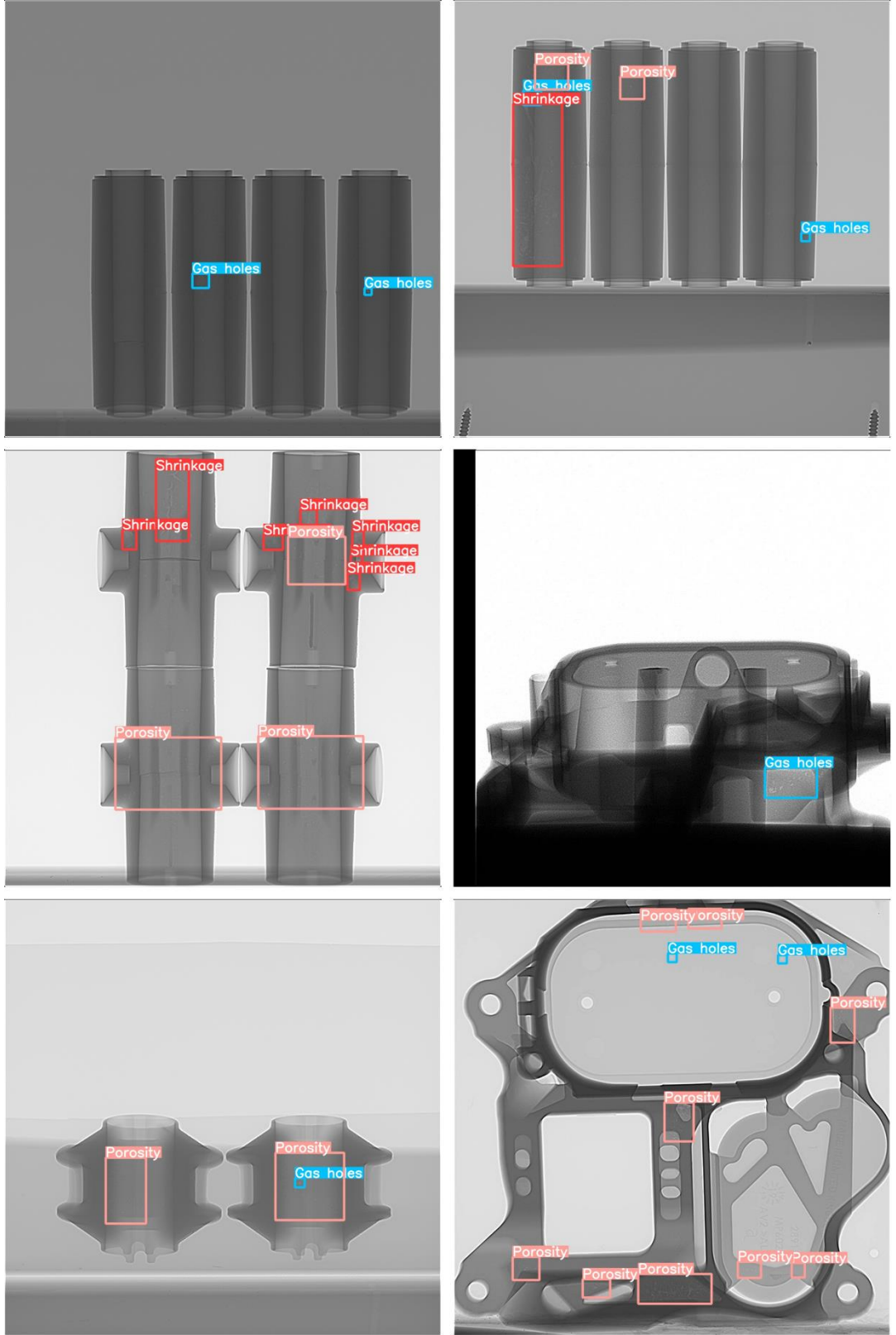
**Çizelge 4.3.** YOLOv5n ve YOLOv5l dedektörlerinin test sonuçları

Model	$mAP$	$AP_s$	$AP_g$	$AP_p$	$Pr$	$Re$	$F_1$	Tespit süresi [ms]
YOLOv5n	0,912	0,923	0,919	0,892	0,912	0,935	0,916	7
YOLOv5l	0,942	0,957	0,943	0,924	0,921	0,966	0,943	29

#### 4.1.3. Kusurların bölütlenmesi ve eşiklenmesi

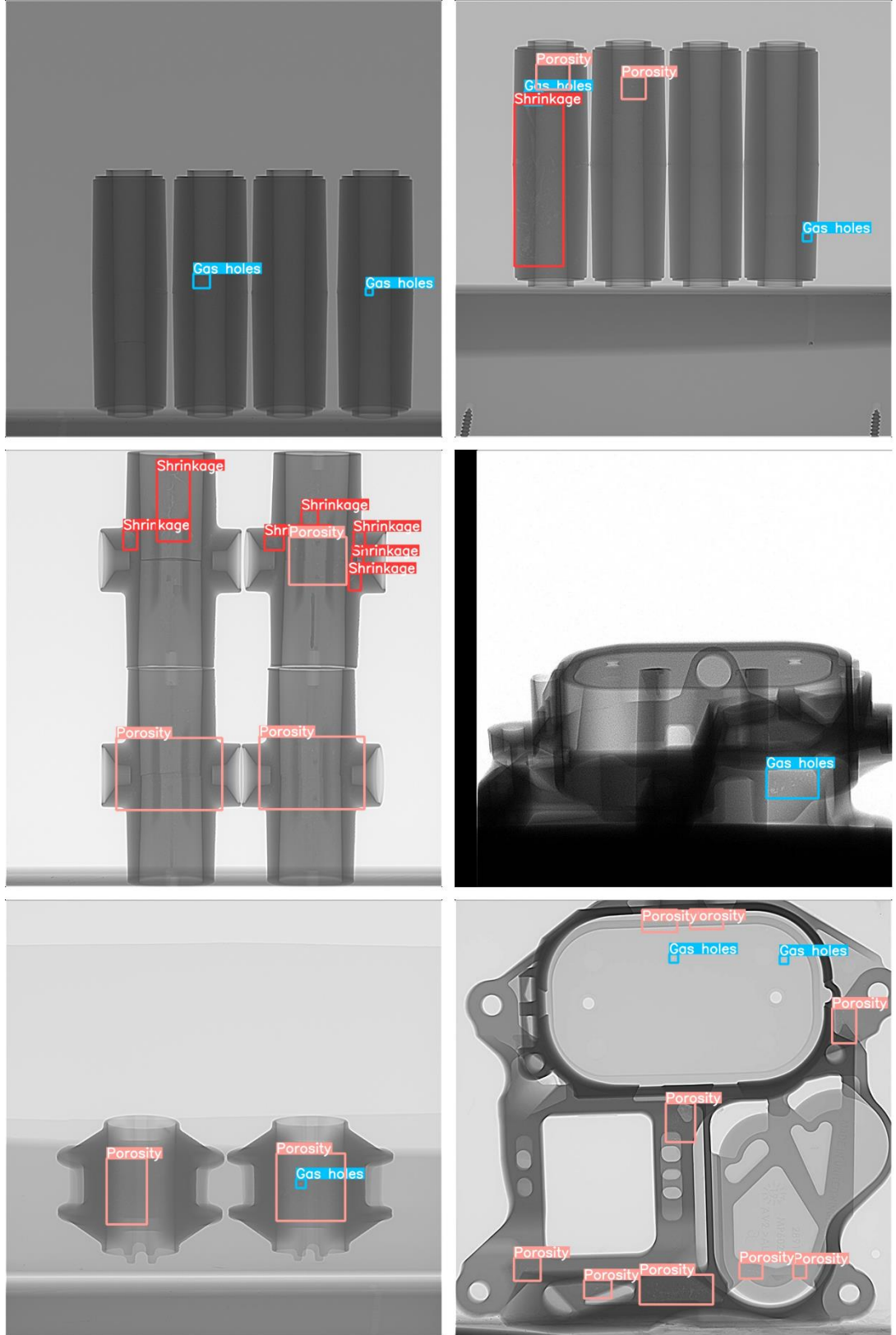
Kusurların bölütlenmesi ve eşiklenmesi için ilk işlem olarak YOLOv5 yöntemi ile yapılan tespit sonucunda elde edilen metin dosyaları kullanılarak kusurları sıkıca kapsayan sınırlayıcı kutular kırılmıştır. Bu görüntülerde kusur içeren daha küçük görüntülerin kırılmasının nedeni, bölütleme görevinin karmaşıklığını azaltmak ve daha küçük bir görüntünün daha kısa sürede işlenmesine olanak sağladığı için işlem süresini kısaltmaktır. Bölütleme için test verilerinden rastgele 108 görüntü seçilmiştir. Kırpma işleminin bir örneği Şekil 4.5'te gösterilmektedir.

YOLOv5, her görüntünün tespitinden sonra bir metin dosyası oluşturur. Bu dosyadaki her satır, X-ışını görüntüsündeki bir kusur hakkında bilgi içerir. Eğer bir metin dosyasında beş satır varsa bu ilgili parça görüntüsünde beş adet kusurun tespit edildiği anlamına gelir. Her satırda toplam beş terim vardır. İlk terim, kusur tipine karşılık gelen indeks numarasıdır (0: Gaz boşluğu, 1: Çekinti ve 2: Porozite). İkinci ve üçüncü terimler, orijinal görüntü üzerindeki koordinatlara göre tespit edilen kusurun sınırlayıcı kutusunun sırasıyla  $x$  ve  $y$  eksenindeki merkezlerdir. Dördüncü ve beşinci terimler, sınırlayıcı kutunun genişliğinin ve yüksekliğinin yarısını temsil eder. Kesilen görüntüler, bölütleme ve eşikleme için kullanılmak üzere ön işlemlerden geçirilmişlerdir. İlk işlem, görüntülerdeki gürültüyü ortadan kaldırmaktır. İkinci işlem, görüntülerdeki kusurları arka plana göre daha görünür hale getirmek için histogram eşitleme işlemidir.

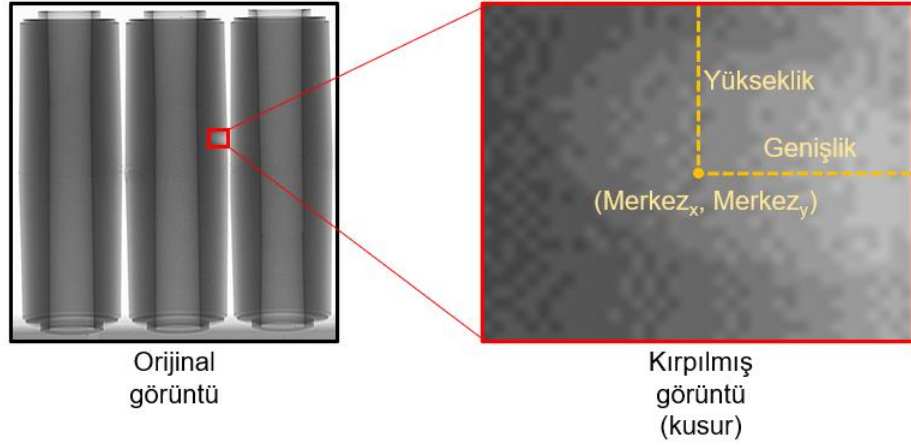


Şekil 4.3. YOLOv5n ile elde edilen çeşitli test çıktıları





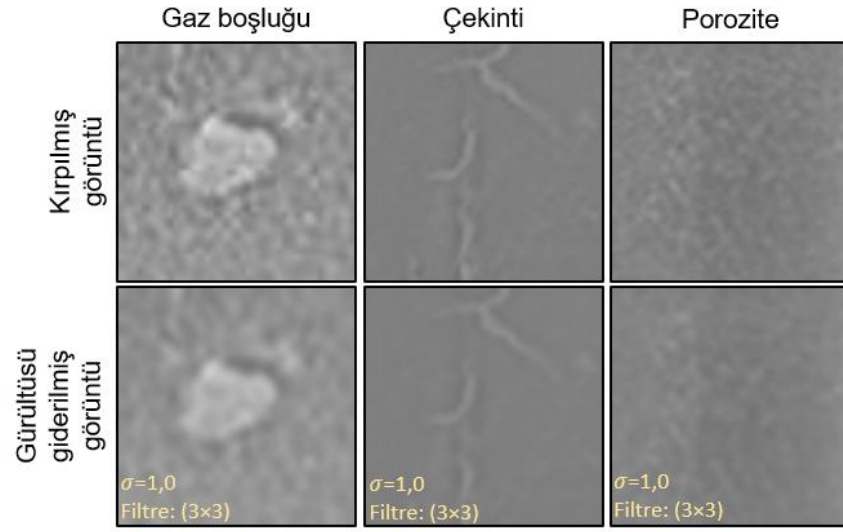
Şekil 4.4. YOLOv51 ile elde edilen çeşitli test çıktıları



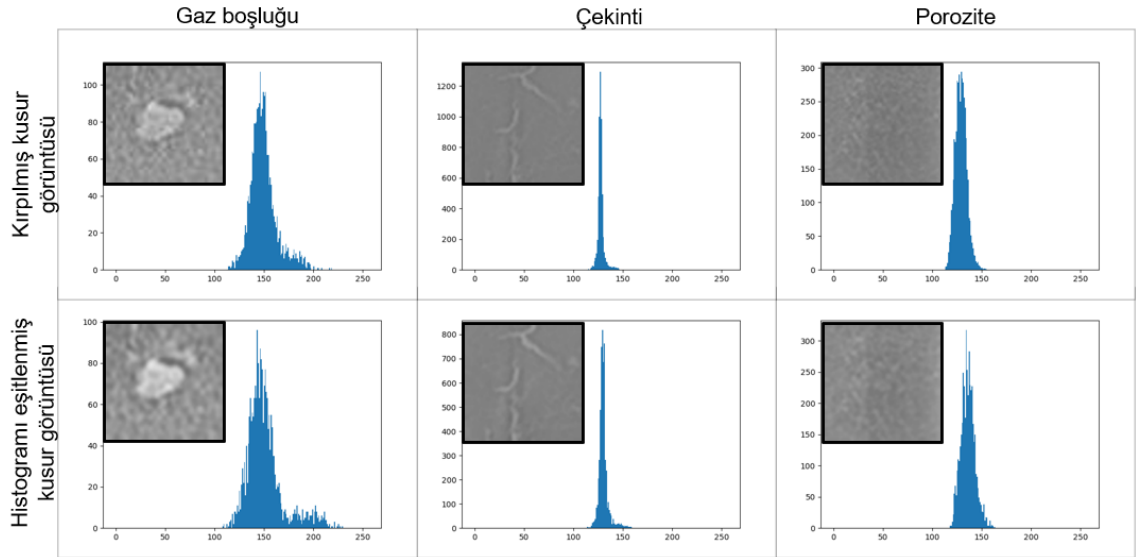
**Şekil 4.5.** Kusurun orijinal görüntüden kırılması

Görüntüleri işlerken karşılaşılan en temel sorunlardan biri olan gürültü, görüntü üzerinde bozulmalara ve istenmeyen detaylara neden olabilir. Bölütlenmek üzere kırılan kusur görüntülerinin her biri, gürültüyü gidermek için Denklem (3.12) kullanılarak Gauss filtresinden geçirilmiştir. Böylece istenmeyen detayların ortadan kaldırılması amaçlanmıştır. Bu işlem gaz boşluğu ve çekinti kusurları içeren görüntülerde çok başarılı sonuçlar verirken, porozite içeren görüntülerde büyük boyutlu Gauss çekirdeklerinin kullanılması bazı görüntülerde porozite kusurunun boyutsal özelliklerinden dolayı kusur detaylarının kaybolmasına neden olmaktadır. Bu durum göz önünde bulundurularak porozite kusurlarının kaybolmaması için her görüntüye aynı boyutta (3×3) Gauss filtresi uygulanmış ve bulanıklaştırma katsayısı  $\sigma = 1,0$  olarak seçilmiştir. Şekil 4.6, üç tip kusurun bazı orijinal ve gürültüsü giderilmiş görüntülerini göstermektedir.

Parlak görünüme sahip bir görüntüdeki piksellerin çoğu yüksek değerlere sahiptir. Aynı şekilde karanlık görüntülerdeki pikseller düşük değerlere sahiptir. Bu, görüntüdeki nesnenin net bir şekilde görülmesini zorlaştırabilir. Bu problemin üstesinden gelmek için görüntü histogramını daha geniş hale getirmek gerekir. Bu işlem, histogram eşitleme yöntemleri ile yapılır ve bu yöntemlerle görüntü kontrastı iyileştirilir. Şekil 4.7, bazı kırılmış kusurların orijinal ve histogramları eşitlenmiş görüntülerini göstermektedir. Gürültü giderme işleminden sonra CLAHE yöntemi ile histogram eşitleme işlemi yapılmıştır. Böylece orijinal parça görüntüsünden kırılan tüm kusur görüntüleri bölütleme ve eşikleme süreci için iyileştirilmişlerdir.



**Şekil 4.6.** Gürültü giderme işlemi uygulanmış görüntü örnekleri



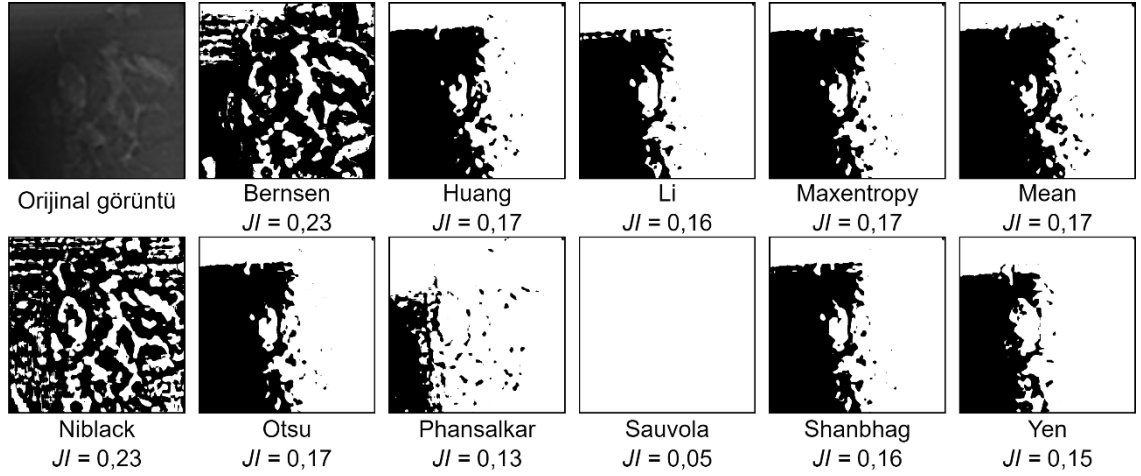
**Şekil 4.7.** Histogramları eşitlenmiş kusur görüntüsü örnekleri

Bir radyografi görüntüsünde anomalilerin (kusurlar) tespit edilmesine ek olarak, bazı durumlarda bu anomalinin boyutu önemli olabilir. Özellikle alüminyum döküm parçalarının muayenesi ve kalite kontrolü gibi işlemlerde kusurun boyutunun belirlenmesi önemlidir. Kusur türlerini ve boyutlarını bilmek, dökümhanelerin müşterilerine ürettikleri parçalar hakkında doğru bilgi vermeleri için çok önemlidir.

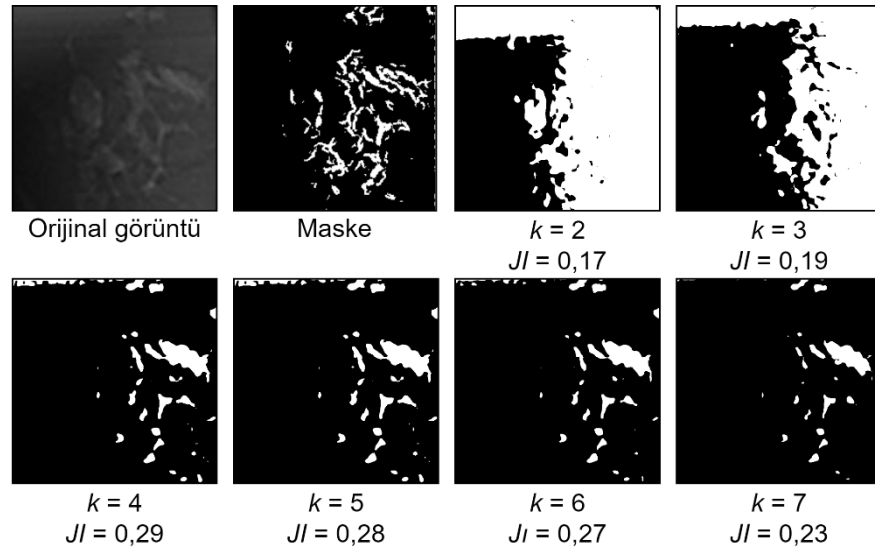
YOLOv5'ten elde edilen çıktı bilgileri kullanılarak, orijinal görüntülerden kırılan kusurlar gürültüleri azaltıldıktan ve histogramları eşitlendikten sonra eşikleme için uygun

bir hale getirilmişlerdir. Literatürde iyi bilinen birçok yöntem, burada yapılan ön çalışmalarda AI-Cast veri setindeki görüntülerde yer alan kusurları eşikleme veya ikilileştirme görevinde başarısız olmuşlardır. Otomatik eşikleme yöntemleri olarak bilinen bu yöntemler, test amacıyla ayrılmış kusur görüntülerinin çoğunu doğru bir şekilde ikili hale getirememiştir. Bir kusur görüntüsü için bazı otomatik eşikleme yöntemlerinin eşikleme sonuçları Şekil 4.8'de gösterilmektedir. Yöntemlerin performansını değerlendirmek için eşik değeri  $JI_t = 0,25$  olarak seçilmiştir. Bu değerin üzerindeki performanslar ( $JI \geq 0,25$ ) başarılı eşikleme olarak kabul edilmiştir. Daha yüksek bir  $JI_t$  değeri seçilmesi daha hassas bir değerlendirme anlamına gelmekle birlikte, eşiklenmesi zor olan bu görüntüler için  $JI_t = 0,25$  olarak seçilmesi ön çalışmalarda uygun bulunmuştur.  $JI$  değerleri Bölüm 3.2.6'da yer alan Şekil 3.32'deki örneğe benzer şekilde yapılmıştır. Burada karşılaştırma için kullanılan temel-gerçekler yani maskeler manuel olarak oluşturulmuştur. Bernsen (1986) tarafından önerilen yöntem eşik değerine en yakın  $JI$  değerinin elde edildiği yöntem olmasına rağmen 0,25'in altında kalmaktadır. En kötü performans Sauvola ve Pietikäinen (2000) tarafından önerilen yöntemle elde edilmiştir. Diğer yöntemler olan (Mean (Glasbey, 1993), Huang (Huang ve Wang, 1995), Otsu (Otsu, 1979) (Denklem (3.14)-(3.16)), Yen (Yen ve diğerleri, 1995), Shanbhag (Shanbhag, 1994), MaxEntropy (Kapur ve diğerleri, 1985), Li (Li ve Lee, 1993; Li ve Tam, 1998), Niblack (Niblack, 1985) ve Phansalkar (Phansalkar ve diğerleri, 2011) otomatik eşikleme yöntemleri ise bu iki yöntem arasında bir performans göstermiş ve eşik değerine ulaşamamışlardır.

Eşikleme işlemi, test için oluşturulan kusur görüntülerine doğrudan uygulandığında birçok otomatik eşikleme yöntemi başarısız olmuştur. Ancak Denklem (3.17) ve (3.18) kullanılarak k-ortalamlar++ algoritması ile yapılan ön çalışmalarda farklı  $k$  değerlerinde kusurun arka plandan başarılı bir şekilde ayrıldığı gözlemlenmiştir. Bu işlem, doğrudan ikilileştirme veya eşikleme olarak değil, önce k-ortalamlar++ algoritması ile görüntünün bölütlenmesi ve ardından en yüksek küme merkezi değerinin eşik değer olarak alınıp, eşikleme işlemi gerçekleştirilerek yapılır (Bölüm 3.2.3). Şekil 4.9 kusur görüntüsü üzerinde k-ortalamlar++ algoritması ile gerçekleştirilen bölütleme ve ardından en yüksek küme merkezi değerine göre eşikleme işleminin çıktılarını göstermektedir. Burada en iyi sonuç  $k = 4$  olduğunda elde edilmiştir ( $JI = 0,29$ ). Bu değer  $JI_t$  değerinden (0,25) büyük olduğu için kusur görüntüsü başarıyla eşiklenmiştir.



Şekil 4.8. Bazı iyi bilinen otomatik eşikleme yöntemlerinin çıktıları



Şekil 4.9. k-ortalamlar++ ile farklı k değerlerinde bölütleme örnekleri

Burada yapılan önemli bir varsayıma değinmek gerekir. Tüm kusur görüntülerinde kusurların en yüksek piksel yoğunluk değerine sahip olması varsayımı altında bu işlemler gerçekleştirilmiştir. Özellikle bölütlenmiş kusur görüntüsü üzerinden en yüksek küme merkezi değerine göre eşik değeri seçimi doğrudan bu kusur ile ilgilidir. Aksi durumda bu yöntem başarısız sonuçlar verecektir. Görüntülerin piksel bazında incelenmesinin ardından, 108 görüntünün yalnızca beşinde bu kusurlardan daha parlak arka plan piksellerinin yer aldığı fark edilmiştir (yani %4.6). Bu oran da bu gibi görüntülerle düşük olasılıkla karşılaşıldığı (Al-Cast2 veri setinde) genellemesini yapmaya olanak

tanılmaktadır. Başka bir deyişle, orijinal görüntüden kırılan çerçevenin bu özel duruma sahip olma olasılığı çok düşüktür.

Ön çalışmalarda ortaya çıkarılan  $k$ -ortalamalar++ algoritmasının bu başarısından sonra algoritma, 108 kusur görüntüsünün tamamı üzerinde test edilmiştir. Çizelge 4.4, 108 görüntü için elde edilen sonuçları göstermektedir. Bu testlerde  $k$  küme sayısı iki ile sekiz arasında alınmış ve en yüksek  $Jl$  değerine sahip  $k$  küme sayısı en başarılı sonucun elde edildiği küme sayısı olarak raporlanmıştır.  $k = 3$  olduğunda, toplam 39 kusur görüntüsü başarıyla eşiklenmiştir ( $\overline{Jl} = 0,81$ ). Tüm kusurların %93,52'si  $k < 6$  durumunda başarıyla eşiklenmiştir.  $k > 6$  olduğunda, tüm görüntülerde  $Jl$  değerleri en düşük seviyededir. Bunun nedeni, küme sayısının çok fazla olmasıdır ve bu da kusuru kendi içinde ayırmakta ve bilgi kaybına neden olmaktadır.

**Çizelge 4.4.**  $k$ -ortalamalar++ kümeleme algoritmasının performansı ( $k$  sayısı)

	En yüksek $Jl$ değerinin elde edildiği $k$ sayısı							$Jl < 0,25$ (tüm $k$ 'ler için)	Toplam görüntü sayısı
	2	3	4	5	6	7	8		
Görüntü sayısı	21	39	32	9	2	0	0	5	108
Toplam görüntülerin #Yüzdesi	19,44	36,11	29,63	8,33	1,85	0,00	0,00	4,63	
Kümülatif toplam	19,44	55,56	85,19	93,52	95,37	95,37	95,37	100,00	
$\overline{Jl}$	0,81	0,70	0,70	0,74	0,85				

Çizelge 4.4'e göre, 108 görüntünün sadece beşi başarılı bölütlenememiştir (tüm  $k$  sayıları için  $Jl < 0,25$ ). Bu beş görüntü analiz edildiğinde bu görüntülerin, daha önceden belirtilen özel duruma sahip görüntüler olduğu fark edilmiştir. Bu beş görüntünün bir ortak özelliği en yüksek yoğunluk değerine sahip piksellerin sadece kusur piksellerinde olmamasıdır. Ayrıca bu beş görüntünün başka bir ortak özelliği parça kalınlığının kesikli bir şekilde değil sürekli bir şekilde değişmesidir (Bu durum teknik resimler analiz edilerek de doğrulanmıştır). Başarılı bir şekilde ikilileştirilmiş çerçevelerde kalınlık tüm arka plan için sabit iken (veya kesikli bir şekilde değişiyorken), kusurların ayrılamadığı çerçevelerde kalınlık kademeli olarak değişmektedir. Arka plandaki pikseller aslında

parçanın kusursuz bölgeleridir. Kalınlık ise, bu arka planda yer alan parça bölgesinin kalınlığını temsil eder (arka plan rengi ne kadar koyu olursa parçanın kalınlığı o kadar fazladır). Bu özel durumla karşılaşılmadığında, k-ortalamar++ algoritması ile önce bölütleme daha sonra bölütlenmiş görüntü üzerinde eşikleme başarılı bir şekilde gerçekleştirilir.

Kusur tiplerine dayalı eşikleme performansı Çizelge 4.5'te verilmiştir. 108 kusur görüntüsünde en yaygın olarak rastlanan çekinti kusurlarının çoğu  $k = 3$  olduğunda başarılı bir şekilde eşiklenmiştir. Ancak diğer kusurlara göre en düşük ortalama  $Jl$  değeri ( $Jl = 0,65$ ) elde edilmiştir. 45 çekinti kusurundan sadece biri eşiklenememiştir. Test görüntülerinde ikinci en yaygın olarak bulunan kusur türü porozitedir. Porozite kusurlarının çoğu,  $k = 4$  olduğunda en yüksek  $Jl$  değeri ile eşiklenmektedir (Ayrıca ortalama  $Jl$  değeri 0,78'dir). Ancak, dört porozite kusuru arka plandan başarılı bir şekilde ayrıştırılamamıştır. Veri setinde en az görülen kusur tipi olan gaz boşluğu ise en yüksek ortalama  $Jl$  değerine sahiptir ( $\bar{Jl} = 0,80$ ). Gaz boşluğu kusurlarının çoğu  $k = 2$  olduğunda en yüksek  $Jl$  değeri ile eşiklenmiştir. Tüm gaz boşluğu kusurları arka plandan başarıyla ayrıştırılmıştır. Gaz boşluğu kusurlarının eşiklenmesi, diğer kusur türlerine göre daha kolay görünmektedir.

Buraya kadar yapılan deneylerin tamamında  $k$  sayısı elle ayarlanmıştır. Ancak alüminyum döküm parçalarının kusur tespiti ve seviyelerinin belirlenmesi sürecinin otomatikleştirilmesi için bu  $k$  sayısının otomatik olarak seçilmesi gerekmektedir. Literatürde bu problemi çözmek için birçok  $k$  sayısı seçim yöntemi bulunmaktadır. Bu yöntemlerden üçünün performansı çalışmada test edilmiştir: Dirsek yöntemi (Thorndike, 1953) (Denklemler (3.19)), Silhouette (Rousseeuw, 1987) (Denklemler (3.20)) ve Davies-Bouldin (Davies ve Bouldin, 1979) (Denklemler (3.21)-(3.25)). Bu yöntemlerin hiçbirinin en iyi sonuçları garanti etmediğini belirtmek gerekir.

Çizelge 4.5'te bu üç yöntemle elde edilen en iyi  $k$  sayısı ile daha önce  $Jl$  değeri hesaplanarak elde edilen en iyi  $k$  sayıları karşılaştırılmıştır.  $Jl$  kullanılarak elde edilen  $k$  sayıları ile aynı değere sahip  $k$  sayıları çoğunlukla Davies-Bouldin yöntemi (%80,56) ile elde edilmiştir. Silhouette kullanıldığında bu oran %70,37 olarak bulunmuştur. Ancak Dirsek yöntemi her iki yöntemle göre oldukça düşük bir performans (%40,74) göstermiştir. Tüm yöntemler için daha önce bulunan  $k$  sayılarından farklı değerlerin elde

edildiği görüntüler incelenmiştir. Yöntemlerin önerdiği  $k$  değeri için  $Jl$  değerleri hesaplanmıştır. Bu hesaplama sonucunda elde edilen  $k$  sayısından farklı değerlere sahip tüm görüntüler için Silhouette ve Davies-Bouldin ile elde edilen  $k$  sayılarındaki  $Jl$  değerleri hepsinde 0,25'in üzerinde çıkmıştır. Ancak aynı durum dirsek yöntemi için geçerli değildir. Dirsek yönteminde farklı bulunan  $k$  değerlerinin yaklaşık yarısında  $Jl$  değeri 0,25'in altında kalmıştır. Bu, Silhouette ve Davies-Bouldin'in, dirsek yöntemine göre otomatik  $k$  değer seçiminde daha tercih edilebilir olduklarını ortaya koymuştur. En iyi  $k$  değerinin otomatik hesaplanması için Davies-Bouldin indeksi, Al-Cast2 veri seti için özellikle uygundur.

Bu nedenle, kusurların  $k$ -ortalamalar++ algoritması kullanılarak bölütlenmesi ve küme sayısı  $k$ 'nin seçiminde Davies-Bouldin yöntemi önerilmektedir.

**Çizelge 4.5.** Üç farklı yöntemin küme sayısı  $k$ 'nin seçimindeki performansı

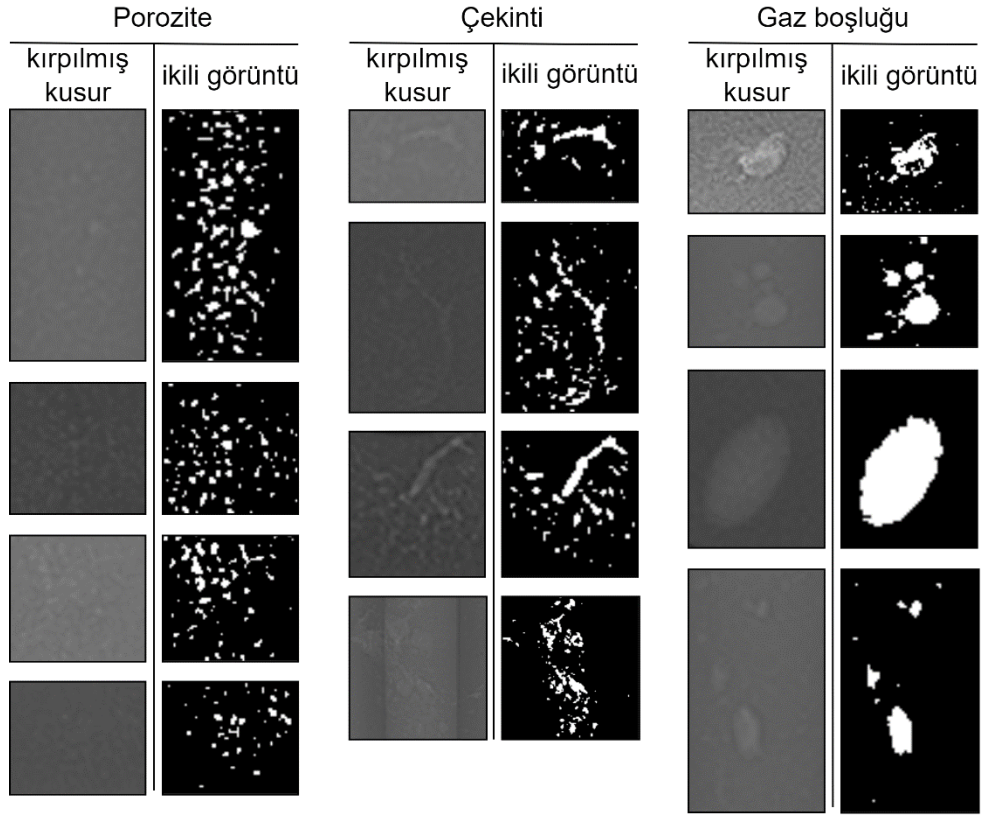
Yöntem	Dirsek	Silhouette	Davies-Bouldin
Aynı $k$ sayısının elde edildiği toplam görüntü sayısı	44	76	87
Tüm görüntüler arasındaki yüzdesi	40,74	70,37	80,56

Son olarak Şekil 4.10'da Davies-Bouldin yöntemi ile seçilen  $k$  değerine göre  $k$ -ortalamalar++ algoritması uygulanarak elde edilen ve daha sonra en yüksek küme merkezi değerine göre eşiklenen bazı görüntü örnekleri bulunmaktadır.

#### 4.1.4. Kusurların seviyelerinin belirlenmesi

Başarıyla ikilileştirilmiş kusur görüntüleri, kusuru nihai olarak seviyelendirmek için gereken son görüntü işleme adıdır. Kusur seviyelendirmesinde kullanılacak referans ölçümler için ASTM standartları kullanılmıştır. Üç tip ASTM kusur referansının görüntülerinden elde edilen ölçüm bilgileri kullanılarak bu işlem gerçekleştirilmiştir (Bkz: Bölüm 3.2.4 ve Şekil 3.30).





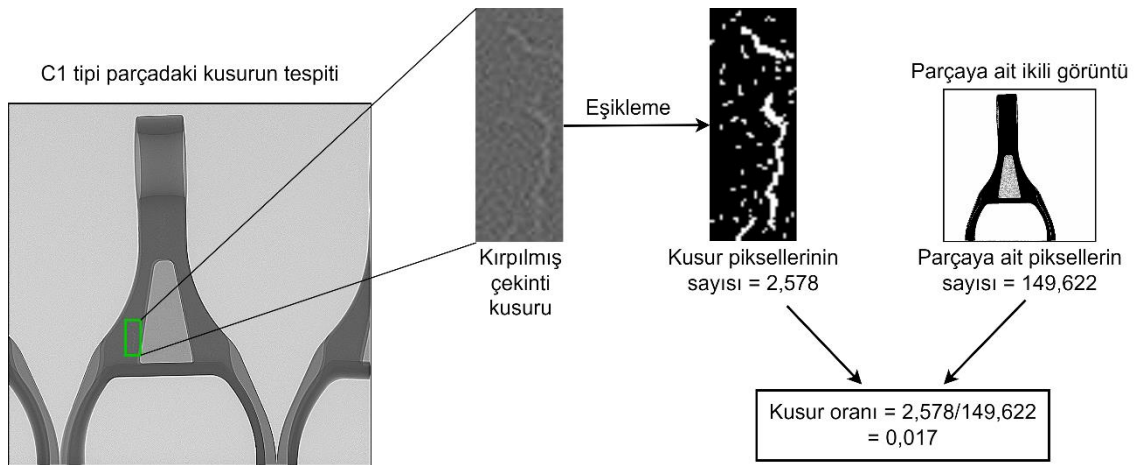
**Şekil 4.10.** Önerilen yöntem grubuyla elde edilen bazı siyah-beyaz görüntü örnekleri

Çizelge 4.6'da her biri 1000×1000 olan görüntüde yedi farklı parça tipinin görüntü üzerindeki toplam piksel sayıları verilmiştir. Parça tipleri A'dan G'ye kodlanmıştır. Beş parça türü (A, B, D, F, G) için tek açıdan görüntüler alınırken, iki parça türü için (C, E) iki farklı açıdan görüntü alınmıştır. Kusur tespitinin yapıldığı parça görüntüsünden kusur çerçeveleri kırıldıktan sonra, Çizelge 4.6'da verilen ve daha önce hesaplanan parçalara ait piksel sayısı orantılanır. Ortaya çıkan kusur oranı, ilgili ASTM kusur referanslarındaki kusur derecelerine (Şekil 3.30) bakılarak derecelendirilir. Bu işlemin örnek bir görüntüsü Şekil 4.11'de verilmiştir. Şekil 4.11'deki adımlarla elde edilen çekinti kusurunun piksel sayısının oranı parçanın toplam piksel sayısına oranı 0,0017 olarak hesaplanmıştır. Bu oran ASTM standartlarında çekinti için verilen derecelerden birinci seviyeye karşılık gelmektedir. Böylece parça üzerinde tespit edilen kusur, tipi, konumu ve seviyesi ile birlikte belirlenmiştir. Buradaki piksel hesaplamaları aslında alan hesaplamaları gibi kullanılmaktadır. Bu hem pratik bir hesaplama sağlamıştır hem de teknik resim bilgisi ihtiyacını ortadan kaldırmıştır.

**Çizelge 4.6.** Parça tiplerine ait piksel sayıları

Parça tipi	1000×1000'lik görüntüde yer alan parçaya ait toplam piksel sayısı
A	94 397
B	111 544
C1	149 622
C2	266 982
D	111 750
E1	364 806
E2	704 673
F	101 620
G	49 455

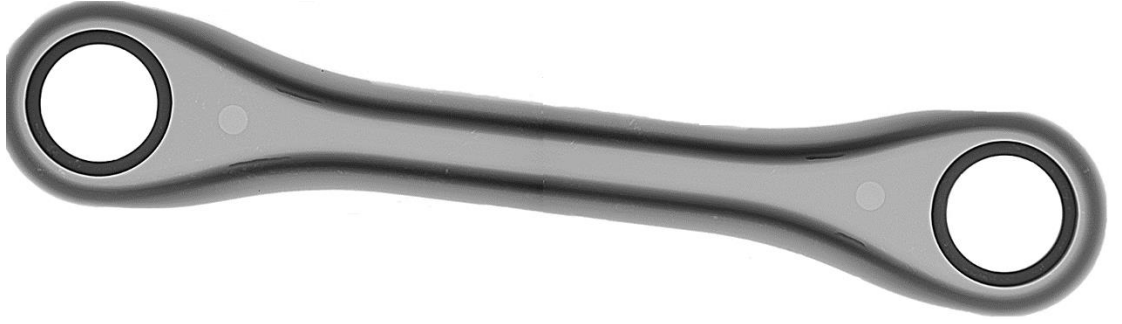
Ancak burada önemli bir varsayım söz konusudur. Kusur seviyelerinin belirlenmesi için yapılan deneylerde kullanılmak üzere seçilen parça tiplerinin hepsi kendi içinde (yani tip bazında) 1000×1000'lik görüntüde aynı sayıda piksel içermektedir. Ancak gerçek bir uygulamada bu durum farklılık gösterebilir. Önerilen yöntemin gerçek bir muayene ortamında uygulanabilmesi için bazı standartların yerine getirilmesi gerekmektedir. X-ışını cihazına konulan bir parçanın her bir radyografi görüntüsü elde edilme işleminde kaynak-parça ve parça-levha arası uzaklıklar aynı olmalıdır. Bu durumda her bir parça tipi için standart bir piksel sayısı kullanılabilir. Aksi halde her seferinde parçanın tamamının bölütlenip eşitlenerek ek işlemler yapılması gerekir.



**Şekil 4.11.** Kusur oranının hesaplanması

#### 4.1.5. Döküm süreci parametreleri

Dökümhanede üretilen alüminyum döküm parçalarındaki iç kusurların kaynağını anlayabilmek için veri setinde yer almayan bir parça tipi üzerinden farklı parametre ayarlarında dökümler yapılmıştır. Buradaki döküm süreci parametreleri yüksek basınçlı enjeksiyon döküm makinesinden (soğuk kamara) alınmıştır. Amaç, önceki aşamalardaki yöntemlerle elde edilen kusur bilgilerini kullanarak farklı döküm süreci parametreleriyle bu kusur bilgileri arasındaki ilişkileri kurmaktır. Kullanılan parçaya ait bir X-ışını görüntüsü Şekil 4.12’de verilmiştir.



**Şekil 4.12.** Parametre analizi için deneylerde kullanılan parçanın X-ışını görüntüsü

Toplam on bir farklı deney, yedi farklı döküm süreci parametresi üzerinden yapılmıştır. Çizelge 4.7’de, deneylerde göz önünde bulundurulmuş döküm süreci parametrelerinin değer aralıkları verilmiştir.

**Çizelge 4.7.** Döküm süreci parametreleri ve değer aralıkları

Döküm süreci parametresi/faktör (birim)	Değer aralığı
İkinci faz azot turu	0,50 – 2,50
Üçüncü faz azot turu	0,50 – 2,50
Enjeksiyon dolu noktası (mm)	250 – 330
Birinci faz hızı (m/s)	0,08 – 0,34
İkinci faz hızı (m/s)	1,15 – 3,01
Üçüncü faz basıncı (mbar)	160 – 390
Enjeksiyon ileri hızı	0,28 – 0,40

Üç fazdan oluşan enjeksiyon sürecinin ilk fazında piston yavaş bir şekilde hareket ederken ikinci fazda çok daha yüksek hızda hareket ederek erimiş metale kalıba doğru enjeksiyon kovani boyunca kuvvet uygular. Üçüncü ve son fazda piston hareketsiz kalarak kalıpta oluşan basıncın korunmasını sağlar ve bu faz erimiş metal, kalıp içerisinde soğuyup katılaşıncaya kadar devam eder. Çizelge 4.7’de verilen parametrelerle yapılan on bir deneye ait veriler Çizelge 4.8’de gösterilmektedir. Çizelge 4.8’in en sağında yer alan sütun bağımlı değişken olarak kullanılan kusur oranıdır. Bu oranlar, bölütleme ve eşikleme aşamasındaki yöntemler kullanılarak elde edilmiştir. Her bir deney sonucunda elde edilen alüminyum parçasına ait X-ışını görüntüleri kullanılarak yapılan bu işlem sonucunda parçada yer alan kusur pikselleri parçaya ait toplam piksel sayısına oranlanmıştır. Üretilen tüm parçalarda yalnızca porozite kusuru bulunmaktadır. Bu nedenden dolayı kusurun tipi kapsam dışı bırakılarak yalnızca kusurun seviyesine odaklanılmıştır. Bununla birlikte Çizelge 4.8’de de görülen kusur oranları ASTM referanslarında birinci ve üçüncü kusur seviyeleri arasına düşmektedir. Tüm seviyeleri kapsamamasından dolayı, kusur oranları doğrudan bağımlı değişken olarak kullanılmıştır.

**Çizelge 4.8.** Deneylere ait veriler

Deney	F1 İkinci faz azot turu	F2 Üçüncü faz azot turu	F3 Enjeksiyon dolu noktası	F4 Birinci faz hızı	F5 İkinci faz hızı	F6 Üçüncü faz basıncı	F7 Enjeksiyon ileri hızı	Kusur oranı
1	2,5	1,5	300	0,30	1,30	320	0,40	0,0016
2	0,5	1,5	300	0,32	1,22	310	0,40	0,0060
3	2,5	1,5	300	0,08	1,30	320	0,28	0,0041
4	2,5	1,5	300	0,15	1,28	308	0,33	0,0057
5	2,5	1,5	330	0,29	2,23	314	0,40	0,0157
6	2,5	1,5	315	0,31	1,15	312	0,40	0,0065
7	2,5	1,5	280	0,29	2,23	312	0,40	0,0059
8	2,5	1,5	250	0,29	3,01	312	0,40	0,0071
9	2,5	1,5	300	0,33	1,61	160	0,40	0,0124
10	2,5	0,5	300	0,33	1,61	160	0,40	0,0115
11	2,5	2,5	300	0,34	1,43	390	0,40	0,0016

Çizelge 4.8’deki parametre değerlerine bakıldığında, buradaki deneylerin bir faktöriyel tasarıma uygun olarak yapılmadığı görülmektedir. Şekil 4.12’deki söz konusu deneysel

amaçlı parçanın üretimi için enjeksiyon pres makinasında yedi adet kontrol edilebilen parametrenin (bağımsız değişken) sınırlı değerleri ile yapılabilecek bir faktör analizinde bağımlı değişken olarak *Kusur oranı* alınmıştır. ANOVA yöntemi ile yapılan faktör analizi, etkileşimler dahil edilmeden yalnızca ana etkileri bulmak için tanımlanmıştır. Elde edilen ANOVA sonuçları Çizelge 4.9'da verilmiştir. Tüm analizler Statistica paket programı (sürüm 12) kullanılarak gerçekleştirilmiştir. Çizelge 4.9'da deneyde dikkate alınan faktörler; ikinci ve üçüncü faz azot turu, enjeksiyon dolu noktası, birinci ve ikinci faz hızı, üçüncü faz basıncı ve enjeksiyon ileri hızı sırasıyla F1-F7 olarak kodlanmıştır.

Düşük bir  $p$  değeri, etkinin büyük olduğunu veya sonucun teorik veya pratik öneme sahip olduğunu gösterir. Çizelge 4.9'daki  $p$  değerlerine bakıldığında %5 anlamlılık düzeyi için enjeksiyon dolu noktası (F3), ikinci faz hızı (F5) ve üçüncü faz basıncı (F6) faktörlerinin etkilerinin önemli olduğu görülmektedir (bu faktörler için  $p < 0,05$ 'tir). Bu faktörlerin dışındaki faktörlere ait  $p$  değeri 0,05'in oldukça üzerindedir. Bu durum ilgili faktörlerin kusur oranına anlamlı bir etkilerinin olmadığını çıkarımını yapmaya olanak tanımaktadır. Döküm süreci göz önünde bulundurulduğunda azot turu ve enjeksiyon ileri hızı gibi faktörler kusur oluşumunda anlamlı bir etkiye sahip değildir. Bunun yanı sıra birinci faz hızı faktörü, döküm parçası oluşumunda kayda değer bir katkısı olmadığı için beklenildiği gibi anlamlı bir etkiye sahip değildir. ANOVA analizi sonucunda elde edilen regresyon katsayıları Çizelge 4.10'da verilmiştir. Bu katsayılar kullanılarak belirlenen parametre değerleri üzerinden kusur oranı hesaplanabilir. Regresyon sonucunda  $R^2 = 0,968$  olarak elde edilirken, düzeltilmiş  $R^2 = 0,892$  olarak elde edilmiştir.

Bu faktör analizinde kusur oranının en düşük olduğu parametre değerleri için, istenirlik en yüksek seviyededir. Yani kusur oranı ve istenirlik arasında ters orantılı bir ilişki söz konusudur. Şekil 4.13'te yedi farklı faktörün ve istenirliğin, kusur oranına göre oluşturulan grafikleri verilmiştir. Şekil 4.13'te kusur oranı üzerinde birinci ve ikinci faz azot turu sayısı, birinci faz hızı ve enjeksiyon ileri hızının etkisinin az olduğu görülmektedir. Bununla birlikte ikinci faz hızı, üçüncü faz basıncı ve enjeksiyon dolu noktasının kusur oranı üzerinde etkisi büyüktür.

Şekil 4.14'te tüm faktörlerin istenirlik seviyesine göre grafikleri verilmiştir. Şekil 4.13'teki benzer yorumlar burada da yapılabilmektedir. Grafiklere göre, enjeksiyon dolu seviyesi ve ikinci faz hızı arttığında kusur oranı artmaktayken yani istenirlik seviyesi

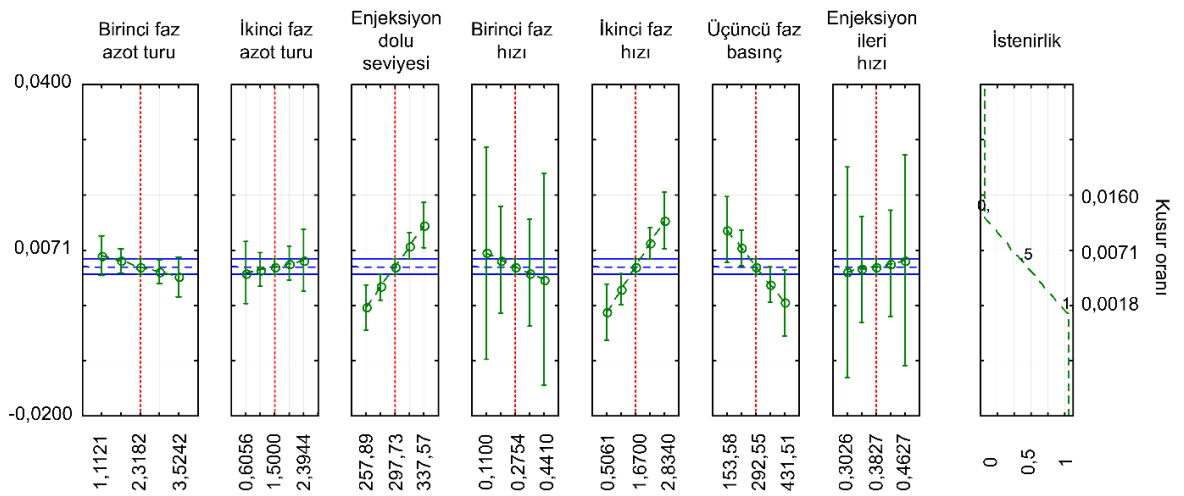
azalmaktayken, üçüncü faz hızı arttığında kusur oranı azalmaktadır, yani istenirlik seviyesi artmaktadır.

**Çizelge 4.9.** Ana etkilere ait ANOVA tablosu

Kaynak	SS	df	MS	F	p
Sabit	0,000014	1	0,000014	6,522	0,084
F1	0,000007	1	0,000007	3,436	0,160
F2	0,000001	1	0,000001	0,439	0,555
<b>F3</b>	<b>0,000081</b>	<b>1</b>	<b>0,000081</b>	<b>38,204</b>	<b>0,008</b>
F4	0,000000	1	0,000000	0,151	0,723
<b>F5</b>	<b>0,000062</b>	<b>1</b>	<b>0,000062</b>	<b>29,171</b>	<b>0,012</b>
<b>F6</b>	<b>0,000028</b>	<b>1</b>	<b>0,000028</b>	<b>13,372</b>	<b>0,035</b>
F7	0,000000	1	0,000000	0,0332	0,866
Hata	0,000006	3	0,000002		

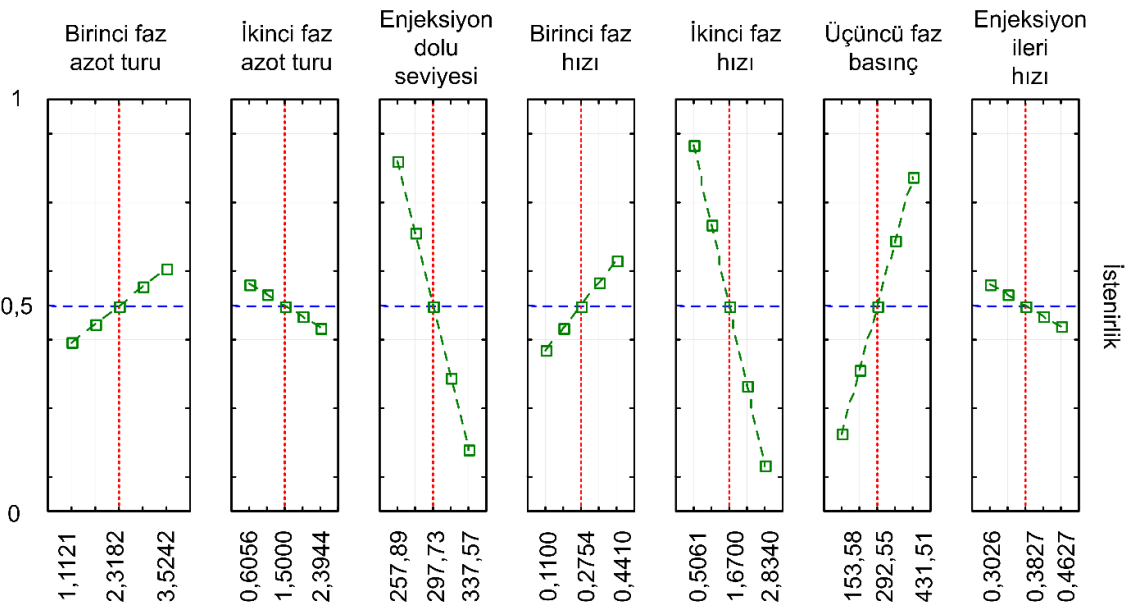
**Çizelge 4.10.** Regresyon modeli katsayıları

Faktör	Katsayı
F1	-0,001601
F2	0,001263
F3	0,000187
F4	-0,014116
F5	0,007131
F6	-0,000048
F7	0,013612
Sabit	-0,046126



**Şekil 4.13.** Kusur oranı ve faktörlerin değişimini gösteren grafikler

Yapılan bu analizlerin sonucu olarak, dikkate alınan parametrelere göre kusur oranını kontrol etmek için, üç faktöre dikkat edilmelidir. Kusur oranının düşük olması hedeflendiğinde ikinci fazda piston hızının ve enjeksiyon kovanının dolu noktasının düşük olması beklenirken, üçüncü fazda kalıp içerisindeki basıncın yüksek olması beklenmektedir. Çalışma faktöriyel deneysel tasarımlara göre kontrollü deneylerle yapıldığında daha anlamlı sonuçlar elde edilebilir. Ayrıca daha güvenilir sonuçlar için tekrarlı deneyleri içeren daha büyük bir veri seti kullanılması gerekir. Bunun yanı sıra kalıp sıcaklığı gibi faktörler de dikkate alınarak, daha kapsamlı analizler yapmak da mümkündür.



Şekil 4.14. İstenirlik ve faktörler grafiği

## 4.2. Tartışma

Al-Cast2 veri setindeki tüm görüntüler gri tonlu görüntüler olduğundan, tüm veriler tek bir kanalda işlenir; yani RGB gibi üç kanalı yoktur. Bu hem kullanılan yöntemlerin işlem süresini kısaltır hem de siyah beyaz piksel değerleri ile nihai görüntüye ulaşmayı kolaylaştırır. İşlem süresini azaltmak için atılan başka bir adım, orijinal görüntülerdeki kusurları kırpmaktır. Kırpılan kusurlu görüntülerin boyutu orijinal görüntüden daha küçük olduğundan, görüntü üzerindeki her işlem çok daha kısa sürecektir. Özellikle bu

durum gürültü giderme, histogram eşitleme ve k-ortalamlar++ algoritması ile bölütleme söz konusu olduğunda önemli bir fark yaratacaktır. Bununla birlikte görüntünün bütünü kullanılarak (yani parçanın tamamının yer aldığı görüntü) eşikleme işlemi yapmak kusurun ayrıştırılmasını zorlaştıracaktır. Bunun sebebi parçaların tamamının yer aldığı görüntülerde kusur ile aynı veya kusurdan daha yüksek piksel yoğunluğu değerlerinin olduğu piksellerin bulunacak olmasıdır.

Al-Cast2 veri seti üzerinde eğitilen YOLOv5 nesne algılama yöntemi ile yapılan kusur tespitleri gerçek zamanlı ve yüksek doğrulukta yapılmıştır. İlk aşama, verileri hazırlama ve etiketleme ve sonrasında modellerin eğitimi ve test edilmesi olarak tamamlanmıştır. İkinci aşama, birçok adımdan oluşan kusur seviyelerinin belirlenmesi sürecini içermektedir. İkinci aşamadaki adımlar sırasıyla, kusur görüntüsünün kırılması, gürültü giderme, histogram eşitleme, k-ortalamlar++ kümeleme algoritması kullanarak bölütleme, eşikleme, piksel hesaplama ve seviyelerin uluslararası bir standarda uygun olarak belirlenmesidir.

Bu tespit ve seviyelendirme süreçlerinde kusur özellikleri ve tipleri ele alınırken ASTM standartları kullanılmıştır. Böylece yapılan birçok işlem bu standardı desteklemektedir. Dökümhane ile müşteri arasında yapılan sözleşmelere göre standartlardaki kusurlara ilişkin farklı değerlendirmelerin olabileceği unutulmamalıdır. ASTM kusur referans radyografları, belirli uygulamaya nasıl uygulandıklarına bağlı olarak tamamen veya kısmen kullanılabilir. Her bir süreksizlik biçimi için, çeşitli dereceler veya kabul sınırlamaları tanımlanabilir. Ek olarak, bir alüminyum parçasının farklı alanları veya bölgeleri için farklı kalite seviyeleri belirtilebilir. Aynı bölgede birden fazla tip kusur varsa özel değerlendirmeler gerekebilir ve bu müşteri ve dökümhane arasında alınacak karar ile belirlenebilir.

Buradaki görüntülerin bölütlenmesi ve eşiklenmesi belirli bir zorluk derecesine sahiptir. Bu durum birçok iyi bilinen otomatik görüntü eşikleme yönteminin başarısızlığıyla ortaya konulmuştur. Yapılan ön analizlerde, otomatik eşikleme yöntemi, test için ayrılan 108 görüntünün önemli bir kısmını eşiklemede başarısız olmuştur. Doğrudan eşikleme yerine ilk olarak çok etkili bir kümeleme algoritması olan k-ortalamlar++ ile bölütleme yapılmıştır. Bölütlenmiş görüntü üzerinde gerçekleştirilen eşikleme yani siyah/beyaz ikilileştirme işlemi ile otomatik eşikleme yöntemlerine göre çok daha yüksek başarı elde



edilmiştir. Bununla birlikte, bazı özel parça kalınlığı durumlarında bu çalışma kapsamında önerilen yöntem de başarısız olmuştur. Ancak, bu özel durumları içeren görüntüler veri setinde çok yaygın değildir.

Seviyelendirme süreci, teknik detay gerektirmeden önceki eşikleme süreci nedeniyle basitleştirilmiştir. Ancak, seviyelendirme için piksel hesaplamalarında bazı varsayımlar vardır. İlk varsayım, parçadan kırılan kusurun doğru eşiklenebilmesidir. İkincisi, kaynak-parça ve parça-film mesafeleri eşit olacak şekilde X-ışını cihazında test veri setindeki farklı parça türlerinin her birinin görüntülerinin oluşturulmasıdır (YOLOv5 dedektörünün eğitimi için kullanılan eğitim verisi setinde yer alan görüntülerde parça ve X-ışını kaynağı veya parça ve algılayıcı panel arasındaki mesafeler farklıdır). Böylece yedi farklı parçanın önceden bölütlenmiş ve ardından eşiklenmiş görüntüleri kullanılarak parça boyutu kolaylıkla hesaplanabilmekte ve bu bilgi aynı parça için diğer tüm görüntülerde kullanılabilir. Bu işlem sayesinde alan hesabı piksel hesabına dönüşmüştür ve teknik resim gerektiren karmaşık işlemlerden kaçınılmıştır.

## 5. SONUÇ

YBD tekniđi ile üretilen alüminyum parçalarda döküm sürecinden kaynaklanan birçok faktör döküm parçasında süreksizliklere yani kusurlara neden olabilmektedir. Bu kusurlar parçaların kalitesini ve dayanıklılıđını olumsuz etkilemekte ve bu durum özellikle otomobil gibi araçlarda yüksek bir riske neden olabilmektedir. Ek olarak, dökümhanelerin müşterilerine ürettikleri parçalar hakkında doğru bilgi verebilmeleri için kusur türlerini, konumlarını ve seviyelerini bilmek kalite açısından büyük önem taşımaktadır. Bu kusurları tespit etmek için kullanılan yöntemlerden biri de X-ışını teknolojisidir. Böylece tüm kusurlar parçaya zarar vermeden anında ekrana aktarılabilmektedir. Ancak, bu denetim ve muayene süreci için iyi eğitilmiş insan denetçiler gerekmektedir. İnsan doğasından kaynaklanan olumsuzlukların giderilmesi ve sürecin istikrarlı ve hızlı bir şekilde sürdürülebilmesi için denetim ve muayene sürecinin otomatik hale getirilmesi amaçlanmıştır.

YBD tekniđi ile üretilen alüminyum döküm parçalarının X-ışını DR görüntüleri kullanılarak gerçekleştirilen bu çalışmanın ilk aşamasında kusur tespiti yüksek başarı ile gerçekleştirilmiştir. Döküm parçalarında yaygın olarak karşılaşılan üç farklı iç kusur tipi için yapılan bu tespit işleminde YOLOv5 derin öğrenme tabanlı nesne tespit yöntemi ile çok yüksek *mAP* değerleri elde edilmiştir (*mAP* = 0,97). Yüksek doğruluđa ek olarak, gerçek zamanlı tespit başka bir başarıdır. Kusur tespiti aşamasından elde edilen çıktılar ile nihayetinde kusurların seviyelendirilmesi süreci başlar. Birçok otomatik eşikleme yöntemi kullanılmış olmasına rağmen biri dışında hiçbir istenilen başarıyı gösterememiştir. O yöntem de k-ortalamlar++ kümeleme algoritması olup, bölütlenen görüntüler en yüksek küme merkezi değerine göre eşiklendiğinde çok daha iyi sonuçlar elde edilmiştir. Bu başarılı eşikleme işlemi ile ASTM referansları bir araya getirilerek gerekli piksel hesaplamaları yapılmış ve kusurlar seviyelendirilmiştir. Böylece YBD tekniđi ile üretilmiş bir alüminyum döküm parçasındaki kusurun tipi ve konumu birlikte tespit edilmiş ve uluslararası bir standarda göre seviyesi belirlenmiştir.

Çalışmada bölütleme işlemi için tıpkı tespit aşamasında olduğu gibi derin öğrenme yöntemi kullanmak mümkündür. Ancak, bu işlem etiketleme süreci nedeniyle çok zaman gerektirmektedir. Nesne tespiti için etiketlemede, nesne sadece bir kutu ile etiketlenirken,

bölütleme söz konusu olduğunda bu işlem, nesnenin şekline göre nesneyi kapsayacak şekilde farklı sayıda çizgilerden oluşacaktır. Özellikle, düz kenarları olmayan veya karmaşık şekillere sahip nesnelere etiketlemek için çok zaman gerekir. Bu işlem pratik olmadığı için çalışmada önerilen yöntem tercih edilmiştir.

Çalışmada uluslararası standartlar dikkate alınarak gerçekleştirilen kusur tespiti ve seviyelendirme işlemlerinin özgün yönleri bulunmaktadır. İlk kez YBD tekniği ile üretilen alüminyum parçalarda porozite kusurlarının tespiti, derin nesne tespit yöntemi ile gerçekleştirilmiştir. Ayrıca buradan elde edilen çıktı ile uluslararası standartlara göre pratik bir şekilde bu kusurun seviyesi belirlenmiştir. Ayrıca veri seti olarak kullanılan Al-Cast2 veri seti, NDT topluluğu tarafından açık erişimli olarak eğitim ve araştırma amacıyla kullanılabilir.

Çalışmada önerilen yöntem farklı standartlara göre yeniden şekillendirilebilir. Ayrıca yine ASTM standartları kullanılarak farklı döküm malzemelerine (magnezyum gibi) uyarlanabilmektedir. Çalışmanın ikinci aşamasında gerçekleştirilen seviyelendirme süreci, üretim koşullarına ve müşterilere göre uyarlanabilir. Örneğin, parçanın tamamına bakmak yerine müşteri tarafından kritik olarak belirtilen bölgelere bakılarak kusurlar giderilebilir. Parçanın farklı bölgeleri için farklı kusur seviyesi varsayımları da olabilir. Bu tür durumlar tamamen müşterilerin isteklerine veya üretim standartlarına bağlıdır. Çalışmada vaat edilen sistem denetim sürecini, mümkün olduğu kadar otomatik, kararlı ve hızlı gerçekleştirmektir. Bu sistem, muayene ve denetim uzmanları için bir karar destek sistemi olarak da kullanılabilir.

## KAYNAKLAR

- Akçay, S., & Breckon, T. P. (2018). An evaluation of region based object detection strategies within X-ray baggage security imagery. *Proceedings - International Conference on Image Processing, ICIP, 2017-Septe*, 1337–1341. <https://doi.org/10.1109/ICIP.2017.8296499>
- Anastasiou, K. (2002). Optimization of the aluminium die casting process based on the Taguchi method. *Proceedings of The Institution of Mechanical Engineers Part B- Journal of Engineering Manufacture - PROC INST MECH ENG B-J ENG MA*, 216, 969–977. <https://doi.org/10.1243/09544050260174175>
- Anijdan, S. H. M., Bahrami, A., Hosseini, H. R. M., & Shafyei, A. (2006). Using genetic algorithm and artificial neural network analyses to design an Al-Si casting alloy of minimum porosity. *Materials and Design*, 27(7), 605–609. <https://doi.org/10.1016/j.matdes.2004.11.027>
- Araujo, P., Fontinele, J., & Oliveira, L. (2020). Multi-Perspective Object Detection for Remote Criminal Analysis Using Drones. *IEEE Geoscience and Remote Sensing Letters*, 17(7), 1283–1286. <https://doi.org/10.1109/LGRS.2019.2940546>
- Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, 07-09-Janu*, 1027–1035.
- ASTM. (2001). Standard Reference Radiographs for Inspection of Aluminum and Magnesium Die Castings 1. *Order A Journal On The Theory Of Ordered Sets And Its Applications*, 03(August), 1–3. <https://doi.org/10.1520/E0155-20>.
- Standard Reference Radiographs for Inspection of Aluminum and Magnesium Die Castings, 03 ASTM E155-20 1 (2014). <https://doi.org/10.1520/E0155-20>.
- Bandara, A., Kan, K., Morii, H., Koike, A., & Aoki, T. (2020). X-ray computed tomography to investigate industrial cast Al-alloys. *Production Engineering*, 14(2), 147–156. <https://doi.org/10.1007/s11740-019-00946-8>
- Berkhin, P. (2006). A survey of clustering data mining techniques. *Grouping Multidimensional Data: Recent Advances in Clustering*, 25–71. [https://doi.org/10.1007/3-540-28349-8\\_2](https://doi.org/10.1007/3-540-28349-8_2)
- Bernsen, J. (1986). Dynamic Thresholding of Gray Level Image. *ICPR '86: Proceedings of International Conference on Pattern Recognition, Berlin*, 1251–1255.
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020a). Yolov4: Optimal speed and accuracy of object detection. *ArXiv Preprint ArXiv:2004.10934*.
- Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020b). YOLOv4: Optimal Speed and Accuracy of Object Detection. *ArXiv*.
- Boerner, H., & Strecker, H. (1988). Automated X-Ray Inspection of Aluminum Castings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(1), 79–91. <https://doi.org/10.1109/34.3869>

- Bonollo, F., Gramegna, N., & Timelli, G. (2015). High-pressure die-casting: Contradictions and challenges. *Jom*, 67(5), 901–908. <https://doi.org/10.1007/s11837-015-1333-8>
- Carrasco, M., & Mery, D. (2011). Automatic multiple view inspection using geometrical tracking and feature analysis in aluminum wheels. *Mach. Vis. Appl.*, 22, 157–170. <https://doi.org/10.1007/s00138-010-0255-2>
- Chiang, K. T., Liu, N. M., & Chou, C. C. (2008). Machining parameters optimization on the die casting process of magnesium alloy using the grey-based fuzzy algorithm. *International Journal of Advanced Manufacturing Technology*, 38(3–4), 229–237. <https://doi.org/10.1007/s00170-007-1103-z>
- Cica, D., & Kramar, D. (2018). Intelligent Process Modeling and Optimization of Porosity Formation in High-Pressure Die Casting. *International Journal of Metalcasting*, 12(4), 814–824. <https://doi.org/10.1007/s40962-018-0213-8>
- Cogranne, R., & Retraint, F. (2014). Statistical detection of defects in radiographic images using an adaptive parametric model. *Signal Processing*, 96(PART B), 173–189. <https://doi.org/10.1016/j.sigpro.2013.09.016>
- Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. *Advances in Neural Information Processing Systems*, 379–387.
- Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 1, 886–893. <https://doi.org/10.1109/CVPR.2005.177>
- Das, S. K., & Yin, W. (2007). Global perspectives: Trends in the global aluminum fabrication industry. *Jom*, 59(2), 83–87. <https://doi.org/10.1007/s11837-007-0027-2>
- Davies, D. L., & Bouldin, D. W. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1*(2), 224–227. <https://doi.org/10.1109/TPAMI.1979.4766909>
- Deng, L., & Yu, D. (2014). Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing*, 7(3–4), 197–387. <https://doi.org/10.1561/20000000039>
- Dhanachandra, N., Mangle, K., & Chanu, Y. J. (2015). Image Segmentation Using K-means Clustering Algorithm and Subtractive Clustering Algorithm. *Procedia Computer Science*, 54, 764–771. <https://doi.org/10.1016/j.procs.2015.06.090>
- Dhillon, A., & Verma, G. K. (2020). Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence*, 9(2), 85–112. <https://doi.org/10.1007/s13748-019-00203-0>
- Ding, M., Song, J., & Honghui, L. (2014). Effect of pouring temperature on typical structure of thin-walled ZL105A alloy casting. *Materials and Manufacturing Processes*, 29(7), 853–863. <https://doi.org/10.1080/10426914.2014.880472>
- Drury, C. G. (2001). Human Factors and Automation in Test and Inspection. In *Handbook*

of *Industrial Engineering: Technology and Operations Management, Third Edition* (pp. 1887–1920).

- Drury, C. G., & Sinclair, M. A. (1983). Human and machine performance in an inspection task. *Human Factors*, 25(4), 391–399. <https://doi.org/10.1177/001872088302500404>
- Du, W., Shen, H., Fu, J., Zhang, G., & He, Q. (2019). Approaches for improvement of the X-ray image defect detection of automobile casting aluminum parts based on deep learning. *NDT and E International*, 107(May), 102144. <https://doi.org/10.1016/j.ndteint.2019.102144>
- Du, W., Shen, H., Zhang, G., Yao, X., & Fu, J. (2022). Interactive defect segmentation in X-Ray images based on deep learning. *Expert Systems with Applications*, 198(March), 116692. <https://doi.org/10.1016/j.eswa.2022.116692>
- Duan, L., Yang, K., & Ruan, L. (2021). Research on Automatic Recognition of Casting Defects Based on Deep Learning. *IEEE Access*, 9, 12209–12216. <https://doi.org/10.1109/ACCESS.2020.3048432>
- Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014). Scalable object detection using deep neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2147–2154.
- Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2015). The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, 111(1), 98–136. <https://doi.org/10.1007/s11263-014-0733-5>
- Feng, D., Harakeh, A., Waslander, S. L., & Dietmayer, K. (2022). A Review and Comparative Study on Probabilistic Object Detection in Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(8), 9961–9980. <https://doi.org/10.1109/TITS.2021.3096854>
- Feng, X., Jiang, Y., Yang, X., Du, M., & Li, X. (2019). Computer vision algorithms and hardware implementations: A survey. *Integration*, 69, 309–320. <https://doi.org/https://doi.org/10.1016/j.vlsi.2019.07.005>
- Ferguson, M., Ak, R., Lee, Y. T. T., & Law, K. H. (2017). Automatic localization of casting defects with convolutional neural networks. *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, 1726–1735. <https://doi.org/10.1109/BigData.2017.8258115>
- Ferguson, M., Ak, R., Lee, Y. T. T., & Law, K. H. (2018). Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning. *Smart and Sustainable Manufacturing Systems*, 2(1), 137–164. <https://doi.org/10.1520/SSMS20180033>
- Filbert, D., Klatte, R., Heinrich, W., & Purschke, M. (1987). Computer aided inspection of castings. *IEEE-IAS Annual Meeting*, 1087–1095.
- Friedman, H. P., & Rubin, J. (1967). On some invariant criteria for grouping data. *Journal of the American Statistical Association*, 62(320), 1159–1178.

- García Pérez, A., Gómez Silva, M. J., & de la Escalera Hueso, A. (2022). Automated Defect Recognition of Castings Defects Using Neural Networks. *Journal of Nondestructive Evaluation*, 41(1). <https://doi.org/10.1007/s10921-021-00842-1>
- Ghiasi, G., Lin, T.-Y., & Le, Q. V. (2019). Nas-fpn: Learning scalable feature pyramid architecture for object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7036–7045.
- Gholizadeh, S. (2016). A review of non-destructive testing methods of composite materials. *Procedia Structural Integrity*, 1, 50–57. <https://doi.org/10.1016/j.prostr.2016.02.008>
- Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision, 2015 Inter*, 1440–1448. <https://doi.org/10.1109/ICCV.2015.169>
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 580–587. <https://doi.org/10.1109/CVPR.2014.81>
- Glasbey, C. A. (1993). An Analysis of Histogram-Based Thresholding Algorithms. In *CVGIP: Graphical Models and Image Processing* (Vol. 55, Issue 6, pp. 532–537). <https://doi.org/10.1006/cgip.1993.1040>
- Gupta, M., Khan, M. A., Butola, R., & Singari, R. M. (2021). Advances in applications of Non-Destructive Testing (NDT): A review. *Advances in Materials and Processing Technologies*, 00(00), 1–22. <https://doi.org/10.1080/2374068X.2021.1909332>
- HajiRassouliha, A., Taberner, A. J., Nash, M. P., & Nielsen, P. M. F. (2018). Suitability of recent hardware accelerators (DSPs, FPGAs, and GPUs) for computer vision and image processing algorithms. *Signal Processing: Image Communication*, 68, 101–119. <https://doi.org/https://doi.org/10.1016/j.image.2018.07.007>
- Hangai, Y., Kuwazuru, O., Yano, T., Utsunomiya, T., Murata, Y., Kitahara, S., Bidhar, S., & Yoshikawa, N. (2010). Clustered Shrinkage Pores in Ill-Conditioned Aluminum Alloy Die Castings. *MATERIALS TRANSACTIONS*, 51(9), 1574–1580. <https://doi.org/10.2320/matertrans.MAW201032>
- Hanke, R., Fuchs, T., & Uhlmann, N. (2008). X-ray based methods for non-destructive testing and material characterization. *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 591(1), 14–18. <https://doi.org/10.1016/j.nima.2008.03.016>
- Hartigan, J A, & Wong, M. A. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1), 100–108. <https://doi.org/10.2307/2346830>
- Hartigan, John A. (1975). Clustering Algorithms. *John Wiley & Sons*, 99th. ed.
- He, K, Gkioxari, G., Dollár, P., & Girshick, R. (2020). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2), 386–397. <https://doi.org/10.1109/TPAMI.2018.2844175>

- He, Kaiming, Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 1904–1916.
- He, Kaiming, Zhang, X., Ren, S., & Sun, J. (2016a). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- He, Kaiming, Zhang, X., Ren, S., & Sun, J. (2016b). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.
- Hernández, S., Sáez, D., & Mery, D. (2004). Neuro-fuzzy method for automated defect detection in aluminium castings. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3212(June 2014), 826–833. [https://doi.org/10.1007/978-3-540-30126-4\\_100](https://doi.org/10.1007/978-3-540-30126-4_100)
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. <http://arxiv.org/abs/1704.04861>
- Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., & Vasudevan, V. (2019). Searching for mobilenetv3. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1314–1324.
- Hu, C., & Wang, Y. (2020). An Efficient Convolutional Neural Network Model Based on Object-Level Attention Mechanism for Casting Defect Detection on Radiography Images. *IEEE Transactions on Industrial Electronics*, 67(12), 10922–10930. <https://doi.org/10.1109/TIE.2019.2962437>
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017a). Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-Janua*, 2261–2269. <https://doi.org/10.1109/CVPR.2017.243>
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017b). Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4700–4708.
- Huang, L. K., & Wang, M. J. J. (1995). Image thresholding by minimizing the measures of fuzziness. *Pattern Recognition*, 28(1), 41–51. [https://doi.org/10.1016/0031-3203\(94\)E0043-K](https://doi.org/10.1016/0031-3203(94)E0043-K)
- Huang, Z., Wang, J., Fu, X., Yu, T., Guo, Y., & Wang, R. (2020). DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection. *Information Sciences*, 522, 241–258.
- Hummel, R. (1977). Image Enhancement By Histogram Transformation. *Comput Graphics Image Process*, 6(2), 184–185. [https://doi.org/10.1016/s0146-664x\(77\)80011-7](https://doi.org/10.1016/s0146-664x(77)80011-7)



- Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size*. 1–13. <http://arxiv.org/abs/1602.07360>
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning*, 448–456.
- Jiang, L., Wang, Y., Tang, Z., Miao, Y., & Chen, S. (2021). Casting defect detection in X-ray images using convolutional neural networks and attention-guided data augmentation. *Measurement: Journal of the International Measurement Confederation*, 170(October 2020), 108736. <https://doi.org/10.1016/j.measurement.2020.108736>
- Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A survey of deep learning-based object detection. *IEEE Access*, 7, 128837–128868. <https://doi.org/10.1109/ACCESS.2019.2939201>
- Jin, C., Kong, X., Chang, J., Cheng, H., & Liu, X. (2020). Internal crack detection of castings: a study based on relief algorithm and Adaboost-SVM. *International Journal of Advanced Manufacturing Technology*, 108(9–10), 3313–3322. <https://doi.org/10.1007/s00170-020-05368-w>
- Jocher, G. (2020). *YOLOv5*. <https://github.com/ultralytics/yolov5>
- Kamalakaran, A., & Rajamanickam, G. (2017). Spatial smoothing based segmentation method for internal defect detection in X-ray images of casting components. *Proceedings - TIMA 2017: 9th International Conference on Trends in Industrial Measurement and Automation*, 0–5. <https://doi.org/10.1109/TIMA.2017.8064796>
- Kapur, J. N., Sahoo, P. K., & Wong, A. K. C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 29(3), 273–285. [https://doi.org/https://doi.org/10.1016/0734-189X\(85\)90125-2](https://doi.org/https://doi.org/10.1016/0734-189X(85)90125-2)
- Kathuria, A. (2018). *What's new in YOLO v3? Towards Data Science*. <https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b>
- Keste, A. A., Gawande, S. H., & Sarkar, C. (2016). Design optimization of precision casting for residual stress reduction. *Journal of Computational Design and Engineering*, 3(2), 140–150. <https://doi.org/10.1016/j.jcde.2015.10.003>
- Ketcham, D. J. (1976). Real-Time Image Enhancement Techniques. *Proc.SPIE*, 0074, 120–125. <https://doi.org/10.1117/12.954708>
- Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. In *Artificial Intelligence Review* (Vol. 53, Issue 8). Springer Netherlands. <https://doi.org/10.1007/s10462-020-09825-6>
- Khan, W. (2014). Image Segmentation Techniques: A Survey. *Journal of Image and Graphics*, 1(4), 166–170. <https://doi.org/10.12720/joig.1.4.166-170>
- Kim, T.-H., Kim, H.-R., & Cho, Y.-J. (2021). Product Inspection Methodology via Deep

- Learning: An Overview. In *Sensors* (Vol. 21, Issue 15). <https://doi.org/10.3390/s21155039>
- Kittur, J. K., Choudhari, M. N., & Parappagoudar, M. B. (2015). Modeling and multi-response optimization of pressure die casting process using response surface methodology. *International Journal of Advanced Manufacturing Technology*, 77(1–4), 211–224. <https://doi.org/10.1007/s00170-014-6451-x>
- Kmieć, M., & Glowacz, A. (2015). Object detection in security applications using dominant edge directions. *Pattern Recognition Letters*, 52, 72–79. <https://doi.org/10.1016/j.patrec.2014.09.018>
- Krimpenis, A., Benardos, P. G., Vosniakos, G. C., & Koukouvitaki, A. (2006). Simulation-based selection of optimum pressure die-casting process parameters using neural nets and genetic algorithms. *International Journal of Advanced Manufacturing Technology*, 27(5–6), 509–517. <https://doi.org/10.1007/s00170-004-2218-0>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90.
- Kujawińska, A., & Vogt, K. (2015). Human factors in visual quality control. *Management and Production Engineering Review*, 6(2), 25–31. <https://doi.org/10.1515/mper-2015-0013>
- Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Mallocci, M., & Kolesnikov, A. (2020). The open images dataset v4. *International Journal of Computer Vision*, 128(7), 1956–1981.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- LeCun, Y., Jackel, L. D., Bottou, L., Cortes, C., Denker, J. S., Drucker, H., Guyon, I., Muller, U. A., Sackinger, E., & Simard, P. (1995). Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural Networks: The Statistical Mechanics Perspective*, 261(276), 2.
- Li, C. H., & Lee, C. K. (1993). Minimum cross entropy thresholding. *Pattern Recognition*, 26(4), 617–625. [https://doi.org/https://doi.org/10.1016/0031-3203\(93\)90115-D](https://doi.org/https://doi.org/10.1016/0031-3203(93)90115-D)
- Li, C. H., & Tam, P. K. S. (1998). An iterative algorithm for minimum cross entropy thresholding. *Pattern Recognition Letters*, 19(8), 771–776. [https://doi.org/https://doi.org/10.1016/S0167-8655\(98\)00057-9](https://doi.org/https://doi.org/10.1016/S0167-8655(98)00057-9)
- Li, H., Li, J., Guan, X., Liang, B., Lai, Y., & Luo, X. (2019). Research on Overfitting of Deep Learning. *Proceedings - 2019 15th International Conference on Computational Intelligence and Security, CIS 2019*, 78–81. <https://doi.org/10.1109/CIS.2019.00025>
- Li, J., Oberdorfer, B., & Schumacher, P. (2019). *Determining Casting Defects in Thixomolding Mg Casting Part by Computed Tomography BT - Shape Casting* (M. Tiryakioğlu, W. Griffiths, & M. Jolly (eds.); pp. 99–103). Springer International

Publishing.

- Li, W., Li, K., Huang, Y., & Deng, X. (2016). *A New Trend Peak Algorithm with X-ray Image for Wheel Hubs Detection and Recognition* (Vol. 575). [https://doi.org/10.1007/978-981-10-0356-1\\_3](https://doi.org/10.1007/978-981-10-0356-1_3)
- Li, X., Tso, S. K., Guan, X. P., & Huang, Q. (2006). Improving automatic detection of defects in castings by applying wavelet technique. *IEEE Transactions on Industrial Electronics*, 53(6), 1927–1933. <https://doi.org/10.1109/TIE.2006.885448>
- Lin, J., Yao, Y., Ma, L., & Wang, Y. (2018). Detection of a casting defect tracked by deep convolution neural network. *International Journal of Advanced Manufacturing Technology*, 97(1–4), 573–581. <https://doi.org/10.1007/s00170-018-1894-0>
- Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *ArXiv Preprint ArXiv:1312.4400*.
- Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2117–2125.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2980–2988.
- Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2020). Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2), 318–327. <https://doi.org/10.1109/TPAMI.2018.2858826>
- Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS(PART 5), 740–755. [https://doi.org/10.1007/978-3-319-10602-1\\_48](https://doi.org/10.1007/978-3-319-10602-1_48)
- Liu, J., & Kim, J. H. (2022). A Variable Attention Nested UNet++ Network-Based NDT X-ray Image Defect Segmentation Method. *Coatings*, 12(5). <https://doi.org/10.3390/coatings12050634>
- Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path Aggregation Network for Instance Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 8759–8768. <https://doi.org/10.1109/CVPR.2018.00913>
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016a). Ssd: Single shot multibox detector. *European Conference on Computer Vision*, 21–37.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016b). SSD: Single shot multibox detector. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS, 21–37. [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
- Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). *A ConvNet*

for the 2020s. 11966–11976. <https://doi.org/10.1109/cvpr52688.2022.01167>

- Lockard, C. D. (2015). Anomaly detection in radiographic images of composite materials via crosshatch regression [Mills College PP - United States -- California]. In *ProQuest Dissertations and Theses*. <https://www.proquest.com/dissertations-theses/anomaly-detection-radiographic-images-composite/docview/1679278047/se-2?accountid=148336>
- Luo, A. A. (2021). Recent advances in light metals and manufacturing for automotive applications. *CIM Journal*, 12(3), 79–87. <https://doi.org/10.1080/19236026.2021.1947088>
- Ma, N., Zhang, X., Zheng, H.-T., & Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. *Proceedings of the European Conference on Computer Vision (ECCV)*, 116–131.
- Manjunath Patel, G. C., Krishna, P., & Parappagoudar, M. B. (2016). Modelling and multi-objective optimisation of squeeze casting process using regression analysis and genetic algorithm. *Australian Journal of Mechanical Engineering*, 14(3), 182–198. <https://doi.org/10.1080/14484846.2015.1093231>
- Martínez-Pastor, J., & Hernández-Ortega, J. J. (2022). A Decision Support System ( DSS ) for the Prediction and Selection of Optimum Operational Parameters in Pressure. *Materials*, 15(15), 5309.
- Mery, D. (2015). Inspection of Complex Objects Using Multiple-X-Ray Views. *IEEE/ASME Transactions on Mechatronics*, 20(1), 338–347. <https://doi.org/10.1109/TMECH.2014.2311032>
- Mery, D., & Filbert, D. (2002). Automated flaw detection in aluminum castings based on the tracking of potential defects in a radiosopic image sequence. *IEEE Transactions on Robotics and Automation*, 18(6), 890–901. <https://doi.org/10.1109/TRA.2002.805646>
- Mery, D., Riffo, V., Zuccar, I., & Pieringer, C. (2013). Automated X-Ray Object Recognition Using an Efficient Search Algorithm in Multiple Views. *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 368–374. <https://doi.org/10.1109/CVPRW.2013.62>
- Mery, Domingo. (2003). Crossing line profile: A new approach to detecting defects in aluminium castings. *Proceedings of the Scandinavian Conference on Image Analysis*, 2749, 725–732.
- Mery, Domingo. (2006). Automated Radioscopic Inspection of Aluminium Die Castings. *Materials Evaluation*, 65(6), 643–647.
- Mery, Domingo. (2020). Aluminum Casting Inspection Using Deep Learning: A Method Based on Convolutional Neural Networks. *Journal of Nondestructive Evaluation*, 39(1), 1–12. <https://doi.org/10.1007/s10921-020-0655-9>
- Mery, Domingo. (2021). Aluminum Casting Inspection using Deep Object Detection Methods and Simulated Ellipsoidal Defects. *Machine Vision and Applications*, 32(3), 1–16. <https://doi.org/10.1007/s00138-021-01195-5>

- Mery, Domingo, & Arteta, C. (2017). Automatic defect recognition in x-ray testing using computer vision. *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, 1026–1035. <https://doi.org/10.1109/WACV.2017.119>
- Mery, Domingo, Riffo, V., Zscherpel, U., Mondragón, G., Lillo, I., Zuccar, I., Lobel, H., & Carrasco, M. (2015). GDxray: The Database of X-ray Images for Nondestructive Testing. *Journal of Nondestructive Evaluation*, 34(4), 1–12. <https://doi.org/10.1007/s10921-015-0315-7>
- Mhalla, A., Chateau, T., Gazzah, S., & Amara, N. E. Ben. (2019). An Embedded Computer-Vision System for Multi-Object Detection in Traffic Surveillance. *IEEE Transactions on Intelligent Transportation Systems*, 20(11), 4006–4018. <https://doi.org/10.1109/TITS.2018.2876614>
- Muthukrishnan. (2020). *Otsu's method for image thresholding explained and implemented*. AI, Computer Vision and Mathematics. <https://muthu.co/otsus-method-for-image-thresholding-explained-and-implemented/>
- Niblack, W. (1985). *An introduction to digital image processing*. Strandberg Publishing Company.
- O’Keeffe, S., & Villing, R. (2018). Evaluating pruned object detection networks for real-time robot vision. *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 91–96. <https://doi.org/10.1109/ICARSC.2018.8374166>
- Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, C(1), 62–66.
- Pang, G., Shen, C., Cao, L., & Hengel, A. Van Den. (2021). Deep Learning for Anomaly Detection: A Review. *ACM Computing Surveys*, 54(2). <https://doi.org/10.1145/3439950>
- Parlak, İ. E., & Emel, E. (2023). Deep learning-based detection of aluminum casting defects and their types. *Engineering Applications of Artificial Intelligence*, 118, 105636, <https://doi.org/10.1016/j.engappai.2022.105636>.
- Phansalkar, N., More, S., Sabale, A., & Joshi, M. (2011). Adaptive Local Thresholding for Detection of. *International Conference on Communications and Signal Processing (ICCSP)*, 218–220.
- Pieringer, C., & Mery, D. (2010). Flaw detection in aluminium die castings using simultaneous combination of multiple views. *Insight - Non-Destructive Testing and Condition Monitoring*, 52, 548–552. <https://doi.org/10.1784/insi.2010.52.10.548>
- Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A., Greer, T., Romeny, B. T. H., Zimmerman, J. B., & Zuiderveld, K. (1987). Adaptive histogram equalization and its variations. In *Computer Vision, Graphics, and Image Processing* (Vol. 38, Issue 1, p. 99). [https://doi.org/10.1016/s0734-189x\(87\)80156-1](https://doi.org/10.1016/s0734-189x(87)80156-1)
- Precedence Research. (2021). *Aluminum Market Size, Growth Report, Trends, 2022-2030*. Precedence Research. <https://www.precedenceresearch.com/aluminum->

market

- Rai, J. K., Lajimi, A. M., & Xirouchakis, P. (2008). An intelligent system for predicting HPDC process variables in interactive environment. *Journal of Materials Processing Technology*, 203(1–3), 72–79. <https://doi.org/10.1016/j.jmatprotec.2007.10.011>
- Rai, P., & Singh, S. (2010). A Survey of Clustering Techniques. *International Journal of Computer Applications*, 7(12), 1–5. <https://doi.org/10.5120/1326-1808>
- Ramírez, F., & Allende, H. (2013). Detection of flaws in aluminium castings: A comparative study between generative and discriminant approaches. *Insight: Non-Destructive Testing and Condition Monitoring*, 55(7), 366–371. <https://doi.org/10.1784/insi.2012.55.7.366>
- Rao, R. V., Kalyankar, V. D., & Waghmare, G. (2014). Parameters optimization of selected casting processes using teaching-learning-based optimization algorithm. *Applied Mathematical Modelling*, 38(23), 5592–5608. <https://doi.org/10.1016/j.apm.2014.04.036>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788.
- Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-Janua*, 6517–6525. <https://doi.org/10.1109/CVPR.2017.690>
- Redmon, J., & Farhadi, A. (2018a). Yolov3: An incremental improvement. *ArXiv Preprint ArXiv:1804.02767*.
- Redmon, J., & Farhadi, A. (2018b). YOLOv3: An incremental improvement. *ArXiv*.
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
- Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(C), 53–65. [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7)
- Salonitis, K., Jolly, M., Pagone, E., & Papanikolaou, M. (2019). Life-cycle and energy assessment of automotive component manufacturing: The dilemma between aluminum and cast iron. *Energies*, 12(13). <https://doi.org/10.3390/en12132557>
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4510–4520.
- Sauvola, J., & Pietikäinen, M. (2000). Adaptive document image binarization. *Pattern Recognition*, 33(2), 225–236. [https://doi.org/https://doi.org/10.1016/S0031-3203\(99\)00055-2](https://doi.org/https://doi.org/10.1016/S0031-3203(99)00055-2)

- Sezgin, M., & Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, *13*(1), 146–165.
- Shafyei, A., Anijdan, S. H. M., & Bahrami, A. (2006). Prediction of porosity percent in Al-Si casting alloys using ANN. *Materials Science and Engineering A*, *431*(1–2), 206–210. <https://doi.org/10.1016/j.msea.2006.05.150>
- Shanbhag, A. G. (1994). Utilization of Information Measure as a Means of Image Thresholding. *CVGIP: Graphical Models and Image Processing*, *56*(5), 414–419. <https://doi.org/https://doi.org/10.1006/cgip.1994.1037>
- Sharma, S., Sharma, S., & Athaiya, A. (2020). Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*, *04*(12), 310–316. <https://doi.org/10.33564/ijeast.2020.v04i12.054>
- Shrivastava, A., Sukthankar, R., Malik, J., & Gupta, A. (2016). Beyond skip connections: Top-down modulation for object detection. *ArXiv Preprint ArXiv:1612.06851*.
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.
- Skalski, P. (2019). *Make-sense*. <https://www.makesense.ai/>
- Solawetz, J. (2020). *YOLOv5 New Version: Improvements And Evaluation*. Roboflow. <https://blog.roboflow.com/yolov5-improvements-and-evaluation/>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.
- Sylla, C. (2002). Experimental investigation of human and machine-vision arrangements in inspection tasks. *Control Engineering Practice*, *10*(3), 347–361. [https://doi.org/10.1016/S0967-0661\(01\)00151-4](https://doi.org/10.1016/S0967-0661(01)00151-4)
- Syrcos, G. P. (2003). Die casting process optimization using Taguchi methods. *Journal of Materials Processing Technology*, *135*(1), 68–74. [https://doi.org/10.1016/S0924-0136\(02\)01036-1](https://doi.org/10.1016/S0924-0136(02)01036-1)
- Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. *Thirty-First AAAI Conference on Artificial Intelligence*.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2818–2826.
- Taguchi, G. (1987). *System of experimental design; engineering methods to optimize*

quality and minimize costs.

- Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., & Le, Q. V. (2019). Mnasnet: Platform-aware neural architecture search for mobile. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2820–2828.
- Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *36th International Conference on Machine Learning, ICML 2019, 2019-June*, 10691–10700.
- Tan, M., Pang, R., & Le, Q. V. (2020a). EfficientDet: Scalable and efficient object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 10778–10787. <https://doi.org/10.1109/CVPR42600.2020.01079>
- Tan, M., Pang, R., & Le, Q. V. (2020b). Efficientdet: Scalable and efficient object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10781–10790.
- Tang, Y., Zhang, X., Li, X., & Guan, X. (2009). Application of a new image segmentation method to detection of defects in castings. *International Journal of Advanced Manufacturing Technology*, 43(5–6), 431–439. <https://doi.org/10.1007/s00170-008-1720-1>
- Tang, Z., Tian, E., Wang, Y., Wang, L., & Yang, T. (2021). Nondestructive Defect Detection in Castings by Using Spatial Attention Bilinear Convolutional Neural Network. *IEEE Transactions on Industrial Informatics*, 17(1), 82–89. <https://doi.org/10.1109/TII.2020.2985159>
- Thorndike, R. L. (1953). Who belongs in the family? *Psychometrika*, 18(4), 267–276. <https://doi.org/10.1007/BF02289263>
- Thuan, D. (2021). *Evolution of Yolo Algorithm and Yolov5: the State-of-the-Art Object Detection Algorithm*. 61.
- Tsoukalas, V. D. (2008). Optimization of porosity formation in AlSi9Cu3 pressure die castings using genetic algorithm analysis. *Materials and Design*, 29(10), 2027–2033. <https://doi.org/10.1016/j.matdes.2008.04.016>
- Tsoukalas, V. D., Mavrommatis, S. A., Orfanoudakis, N. G., & Baldoukas, A. K. (2004). A study of porosity formation in pressure die casting using the Taguchi approach. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 218(1), 77–86. <https://doi.org/10.1243/095440504772830228>
- Wan, S., & Goudos, S. (2020). Faster R-CNN for multi-class fruit detection using a robotic vision system. *Computer Networks*, 168, 107036. <https://doi.org/10.1016/j.comnet.2019.107036>
- Wang, J., Fu, P., & Gao, R. X. (2019). Machine vision intelligence for product defect inspection based on deep learning and Hough transform. *Journal of Manufacturing Systems*, 51(March), 52–60. <https://doi.org/10.1016/j.jmsy.2019.03.002>
- Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart



- manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48, 144–156. <https://doi.org/10.1016/j.jmsy.2018.01.003>
- Wang, Y., Wu, S., Niu, L., Xue, X., Zhang, J., & Wenfeng, X. (2014). Optimization of low-pressure die casting process parameters for reduction of shrinkage porosity in ZL205A alloy casting using Taguchi method. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 228(11), 1508–1514. <https://doi.org/10.1177/0954405414521065>
- Woldstad, J., & Pesante-Santana, J. (2006). Quality Inspection Task in Modern Manufacturing. *International Encyclopedia of Ergonomics and Human Factors, Second Edition - 3 Volume Set*. <https://doi.org/10.1201/9780849375477.ch445>
- Wu, B., Iandola, F., Jin, P. H., & Keutzer, K. (2017). Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 129–137.
- Wu, J. (2017). Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5(23), 495.
- Xu, R., Lin, H., Lu, K., Cao, L., & Liu, Y. (2021). A forest fire detection system based on ensemble learning. *Forests*, 12(2), 1–17. <https://doi.org/10.3390/f12020217>
- Yagüe, F. J., Diez-Pastor, J. F., Latorre-Carmona, P., & Osorio, C. I. G. (2022). Defect detection and segmentation in X-Ray images of magnesium alloy castings using the Detectron2 framework. *ArXiv Preprint ArXiv:2202.13945*.
- Yarlagadda, P. K. D. V., & Chiang, E. C. W. (1999). Neural network system for the prediction of process parameters in pressure die casting. *Journal of Materials Processing Technology*, 89–90, 583–590. [https://doi.org/10.1016/S0924-0136\(99\)00071-0](https://doi.org/10.1016/S0924-0136(99)00071-0)
- Yourui, T., Shuyong, D., & Xujing, Y. (2016). Reliability modeling and optimization of die-casting existing epistemic uncertainty. *International Journal on Interactive Design and Manufacturing*, 10(1), 51–57. <https://doi.org/10.1007/s12008-014-0239-y>
- Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., & Lee, B. (2022). A survey of modern deep learning based object detection models. *Digital Signal Processing: A Review Journal*, 126, 103514. <https://doi.org/10.1016/j.dsp.2022.103514>
- Zaitoun, N. M., & Aqel, M. J. (2015). Survey on Image Segmentation Techniques. *Procedia Computer Science*, 65(Iccmit), 797–806. <https://doi.org/10.1016/j.procs.2015.09.027>
- Zhang, B., Cockcroft, S. L., Maijer, D. M., Zhu, J. D., & Phillion, A. B. (2005). Casting defects in low-pressure die-cast aluminum alloy wheels. *Jom*, 57(11), 36–43. <https://doi.org/10.1007/s11837-005-0025-1>
- Zhang, J., Guo, Z., Jiao, T., & Wang, M. (2018). Defect Detection of Aluminum Alloy Wheels in Radiography Images Using Adaptive Threshold and Morphological

- Reconstruction. In *Applied Sciences* (Vol. 8, Issue 12). <https://doi.org/10.3390/app8122365>
- Zhang, L., & Wang, R. (2013). An intelligent system for low-pressure die-cast process parameters optimization. *International Journal of Advanced Manufacturing Technology*, 65(1–4), 517–524. <https://doi.org/10.1007/s00170-012-4190-4>
- Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6848–6856.
- Zhao, X., He, Z., & Zhang, S. (2014). Defect detection of castings in radiography images using a robust statistical feature. *Journal of the Optical Society of America. A, Optics, Image Science, and Vision*, 31(1), 196–205. <https://doi.org/10.1364/JOSAA.31.000196>
- Zhao, X., He, Z., Zhang, S., & Liang, D. (2015). A sparse-representation-based robust inspection system for hidden defects classification in casting components. *Neurocomputing*, 153. <https://doi.org/10.1016/j.neucom.2014.11.057>
- Zheng, X., Zheng, S., Kong, Y., & Chen, J. (2021). Recent advances in surface defect inspection of industrial products using deep learning techniques. *The International Journal of Advanced Manufacturing Technology*, 113(1), 35–58. <https://doi.org/10.1007/s00170-021-06592-8>
- Zhu, P., Wen, L., Bian, X., Ling, H., & Hu, Q. (2018). *Vision Meets Drones: A Challenge*. 1–11. <http://arxiv.org/abs/1804.07437>
- Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). *Object Detection in 20 Years: A Survey*. 1–39. <http://arxiv.org/abs/1905.05055>
- Zuiderveld, K. (1994). Contrast Limited Adaptive Histogram Equalization. In *Graphics Gems*. Academic Press, Inc. <https://doi.org/10.1016/b978-0-12-336156-1.50061-6>

## ÖZGEÇMİŞ

Adı Soyadı : İsmail Enes Parlak  
Doğum Yeri ve Tarihi : Sakarya/16.11.1990  
Yabancı Dil : İngilizce

### Eğitim Durumu

- Lise : Şükrü Şankaya Anadolu Lisesi (2008)
- Lisans : Bursa Uludağ Üniversitesi Endüstri Mühendisliği (2015)
- Yüksek Lisans : Bursa Uludağ Üniversitesi Endüstri Mühendisliği (2018)

Çalıştığı Kurum : Bursa Teknik Üniversitesi (2017-devam)

İletişim (e-posta) : enes.parlak@btu.edu.tr

### Akademik çalışmalar

: Parlak, İ.E., & Emel, E. (2023). Deep learning-based detection of aluminum casting defects and their types. *Engineering Applications of Artificial Intelligence*, 118, 105636, <https://doi.org/10.1016/j.engappai.2022.105636>.

Parlak, İ.E., Emel, E. (2022, Ekim 15-16). Detection And Grading Of Gas Holes In Aluminum Casting Parts Using Deep Learning-Based Methods. Anadolu 10th International Conference on Applied Sciences, Diyarbakır, Türkiye.