T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME YÖNTEMLERİ İLE METALİK YÜZEYLERDE KUSUR TESPİTİ VE SINIFLANDIRILMASI

DOKTORA TEZİ

Feyza SELAMET

Bilgisayar Mühendisliği Anabilim Dalı

NİSAN 2023

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME YÖNTEMLERİ İLE METALİK YÜZEYLERDE KUSUR TESPİTİ VE SINIFLANDIRILMASI

DOKTORA TEZİ

Feyza SELAMET

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Dr. Öğr. Üyesi Serap ÇAKAR

NİSAN 2023

Feyza SELAMET tarafından hazırlanan "Derin Öğrenme Yöntemleri ile Metalik Yüzeylerde Kusur Tespiti ve Sınıflandırılması" adlı tez çalışması 14.04.2023 tarihinde aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Sakarya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Doktora tezi olarak kabul edilmiştir.

Tez Jürisi

Jüri Başkanı : **Prof. Dr. Abdullah FERİKOĞLU**
Sakarya Uygulamalı Bilimler Üniversitesi

Jüri Üyesi : **Prof. Dr. Cemil ÖZ**
Sakarya Üniversitesi

Jüri Üyesi : **Dr. Öğr. Üyesi Serap ÇAKAR** (Danışman)
Sakarya Üniversitesi

Jüri Üyesi : **Doç. Dr. Halit ÖZTEKİN**
Sakarya Uygulamalı Bilimler Üniversitesi

Jüri Üyesi : **Dr. Öğr. Üyesi Selman HIZAL**
Sakarya Uygulamalı Bilimler Üniversitesi

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Sakarya Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim-Öğretim Yönetmeliğine ve Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesine uygun olarak hazırlamış olduğum "DERİN ÖĞRENME YÖNTEMLERİ İLE METALİK YÜZEYLERDE KUSUR TESPİTİ VE SINIFLANDIRILMASI" başlıklı tezin bana ait, özgün bir çalışma olduğunu; çalışmamın tüm aşamalarında yukarıda belirtilen yönetmelik ve yönergeye uygun davrandığımı, tezin içerdiği yenilik ve sonuçları başka bir yerden almadığımı, tezde kullandığım eserleri usulüne göre kaynak olarak gösterdiğimi, bu tezi başka bir bilim kuruluna akademik amaç ve unvan almak amacıyla vermediğimi ve 20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince Sakarya Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Enstitü tarafından belirlenmiş ölçütlere uygun rapor alındığını, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun ortaya çıkması halinde doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi beyan ederim.

(14/04/2023)

Feyza SELAMET

Çok değerli aileme ve tüm sevdiklerime…

TEŞEKKÜR

Doktora eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Dr. Öğr. Üyesi Serap ÇAKAR'a teşekkürlerimi sunarım.

Tez çalışmamda büyük katkıları olan, bilgi ve deneyimlerinden yararlandığım sayın hocalarım Prof. Dr. Cemil Öz, Prof. Dr. Abdullah Ferikoğlu ve Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK) 1505 projesi kapsamındaki 5160018 proje numaralı proje ekip arkadaşlarıma katkılarından dolayı teşekkür ederim.

Çalışmalarım süresince tüm zorlukları benimle göğüsleyen ve hayatımın her evresinde bana destek olan çok değerli aileme ve sevgili eşim Burak Selamet'e sonsuz teşekkürlerimi sunarım.

Feyza SELAMET

İÇİNDEKİLER

Sayfa

KISALTMALAR

AI	: Yapay Zeka (Artificial Intelligence)
ANN	: Yapay Sinir Ağı (Artificial Neural Network)
CNN	: Evrişimli Sinir Ağı (Convolutional Neural Network)
CPU	: Merkezi İşlem Birimi (Central Processing Unit)
DL	: Derin Öğrenme (Deep Learning)
GPU	: Grafik İşlem Birimi (Graphics Processing Unit)
IoU	: Birleşim Üzerinden Kesişim (Intersection Over Union)
MaP	: Ortalama Hassasiyet (Mean Average Precision)
ML	: Makine Öğrenmesi (Machine Learning)
P	: Hassasiyet (Precision)
R	: Duyarlılık (Recall)
SFS	: Görüntü Tonlarından Şekil Elde Etme (Shape From Shading)
YOLO	: Sadece Bir Kez Bak (You Look Only Once)

TABLO LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

DERİN ÖĞRENME YÖNTEMLERİ İLE METALİK YÜZEYLERDE KUSUR TESPİTİ VE SINIFLANDIRILMASI

ÖZET

Son yıllarda üretim aşamasında otomatik hata tespit sistemlerine olan ihtiyaç artmaktadır. Ürünlerin kusurlarını tespit etmek ve yerlerini belirlemek, önemli ve gerekli bir kalite kontrol sürecidir. Kusur tipinin ve kusurlu alanın kısa sürede tespit edilmesi de kalite kontrol performansı açısından oldukça önemlidir.

Günümüzde, insan iş gücüne dayalı kusurların kontrol edilmesi, üretim sürecinde geleneksel bir yöntem olarak kusurları tespit etmek için hala kullanılmaktadır. İnsana bağlı olan bu yöntemde hız oranı düşük ve hata oranı yüksektir. Üretim aşamasındaki yüzeylerin kalite kontrol çalışmalarında bilgisayarla görme teknikleri sıklıkla kullanılmaktadır. Birçok endüstriyel uygulamada, tekstil, metal ve cam kusur tespiti gibi yüzey kusur tespiti yaygın olarak gerçekleşir. Metal yüzeylerde çeşitli ve karmaşık tipte kusurlar (yamalar, noktalı yüzeyler, çatlak ve çizik, girintili tufal, madde karışımı vb.) vardır. Gerçek zamanlı metalik hata tespit sistemlerinde hız ve yüksek doğruluk üretim aşamasına olumlu etki yapmaktadır.

Yüzeylerde kusur tespiti için geleneksel yöntemler, görüntü işleme veya makine öğrenmesi tekniklerine dayanır, ancak belirli ölçeklerde veya belirli aydınlatma koşullarında düşük gürültü ve güçlü kontrast ile farklı kusurları tespit edebilirler. Günümüzde endüstriyel denetim sistemlerinde bilgisayarlı görme ve derin öğrenme yaklaşımları önemli bir yere sahiptir. Bilgisayarla görme teknolojisi, üretim hattındaki ürünlerin hızlı ve hatasız kontrolü için gereklidir. Bilgisayarla görü kavramının önemi, klasik yöntemlerin sorunları göz önüne alındığında anlaşılır. Metal yüzeyler aydınlatma ve ışık yansıması gibi çevresel faktörlerden kolayca etkilendiğinden, metalik yüzeylerde kusur tespiti zorlu bir problemdir. Karmaşık gerçek dünya problemlerinde geleneksel kusur tespiti algoritmaları verimsiz ve hatalıdır. Sonuç olarak bu çalışmada, çatlak, çizik, vb. metal yüzey kusur tiplerini tespit etmek ve sınıflandırmak için yeni bir yöntem önerilmiştir. Yüzey özelliklerini çıkarabilen Shape From Shading (SFS) algoritması ile kusurlar tespit edilmiştir. Kusurun tipi ve yeri Faster Regional Convolutional Neural Network (Faster R-CNN) tarafından belirlenmiştir. Kusurlu veriler için Northeastern Üniversitesi (NEU) yüzey kusur veritabanı kullanılmıştır. Önerilen algoritma, etiketleme performansını göstermek için etiketlenmemiş bir veri kümesi KolektorSDD2 (KSDD2) üzerinde de test edilmiştir. Sonuçlar hem etiketli hem de etiketsiz veri kümeleri üzerinde, otomatik hata tespiti, sınıflandırma ve etiketlemede en gelişmiş performansı göstermiştir. Önerilen yöntem, metal yüzeydeki kusurların tespiti için başarılı sonuçlara sahiptir ve ortalama hassasiyet 0.83'tür. Çatlaklar, noktalı yüzeyler, yamalar, çizikler, madde karışımı ve girintili tufal ortalama doğruluğu sırasıyla 0.98, 0.81, 0.90, 0.79, 0.88 ve 0.62'dir

.

DEFECT DETECTION AND CLASSIFICATION ON METALLIC SURFACES USING DEEP LEARNING METHODS

SUMMARY

In recent years, the need for automatic defect detection systems has been increasing in the production phase. Identifying and locating product defects is an important and necessary quality control process. Detecting the type of failure and the defective area in a short time is also very important in terms of quality control performance.

In computer vision technology is important for the fast and accurate inspection of products on the production line. Considering the problems with conventional methods, the significance of computer vision becomes apparent.

One of the primary application areas of computer vision is industrial automation. Quality control plays a vital role in the production stage. Quality control is important in the production process because the quality of the produced goods is a critical factor for both customer satisfaction and maintaining the company's reputation. The purpose of quality control is to ensure that products comply with defined quality standards and to identify any potential defects that may occur during the production process. This enables the early detection of errors in the production process, allowing for corrections and improvements in product quality. Quality control also helps ensure factors such as the reliability and durability of the company's manufactured products. Detecting product defects and identifying their locations is an essential and necessary part of the quality control process. The prompt identification of the defect type and the defect area is crucial for quality control performance.

Computer vision can be defined as a field that enables a computer to process digital images and generate outputs. By employing image processing, modeling, and analysis techniques, computers can perceive, recognize, and classify objects and scenes. Industrial automation, on the other hand, is a field that facilitates the automation of production processes in an industrial manufacturing environment. Industrial automation applications offer numerous benefits, such as increasing efficiency, ensuring product quality, reducing errors caused by human factors, and enhancing safety.

Industrial automation and computer vision assist businesses in optimizing their production processes, reducing errors, and improving product quality. Advances in these fields bring significant benefits in terms of efficiency, safety, and sustainability.

Surface defect detection is common in many industrial applications, such as textile, metal and glass flaw detection. There are various and complex types of defects (patches, inclusions, scratches, pitted surface, rolled scale, etc.) on metal surfaces. Fastness and high accuracy in real-time metallic defect detection systems have a positive effect on the production stage.

Currently, the labeling process for training data, which is carried out prior to the use of object recognition algorithms, is still done manually. During the training process with Faster R-CNN for classification, each erroneous region needs to be labeled separately. Automating this process is crucial in industrial control systems because individually labeling a dataset can be time-consuming. The aim of this study is to combine the SFS and Deep Learning methods to automate the labeling process and save time. To achieve successful performance in object recognition and classification applications with Faster R-CNN, a large amount of labeled data is required during the training process. The NEU and KolektorSDD2 datasets used in this study contain 1800 and 3335 images, respectively. While manually labeling the images is time-consuming, automating the labeling process with SFS has significantly improved the results in a much shorter time in our study.

Labeling data manually is a time-consuming process. Therefore, using algorithms can accelerate and optimize this process. This also reduces human error in the labeling process. Algorithms can automatically label data based on predefined characteristics, providing a significant advantage over manual labeling by humans, resulting in faster and more accurate results, especially on large data sets. Applications developed with this method enable data scientists to analyze data faster and obtain more accurate results.

In this study, the aim is to detect various types of errors that may occur on the surfaces of metal components by using the Shape From Shading (SFS) method, which extracts surface features, and automatically label the erroneous regions of the images obtained from the NEU and KSDD2 datasets. As a result, instead of manually labeling large datasets, the process is automated, leading to efficiency in terms of time.

Furthermore, by eliminating human errors such as fatigue and inattention during the labeling of error positions, the labeling of error locations has become more precise. The images processed with SFS were classified by Faster R-CNN, a popular and successful method in recent times.

Today, human inspectors are still used to detect defects as a traditional method in the manufacturing process. Computer vision techniques are frequently used in production systems for quality control to increase production speed, reduce error rates and prevent human errors such as fatigue and distraction. Moreover, the labeling process carried out before the training is still done manually. Defect regions must be individually labeled when classifying with Faster R-CNN before the training process. Automating this process is vital in industrial control systems, as it is very time-consuming to label the dataset one by one. This study aims to automate the labeling process and save time by synthesizing and using SFS and Deep Learning methods. In order to achieve successful performance in object recognition and classification applications with Faster R-CNN, a large number of labeled data is needed in the training process. For example, the NEU and KolektorSDD2 datasets used in this study contain 1800 and 3335 images, respectively. While manually labeling the images takes a long time, automating it with SFS yielded considerably better outcomes in musch less period.

Traditional methods for defect detection on surfaces rely on image processing or machine learning techniques, but they can detect different defects with low noise and strong contrast at certain scales or under certain lighting conditions. Today, computer vision and deep learning approaches have an important place in industrial control

systems. Computer vision technology is essential for fast and accurate control of products on the production line. The importance of the concept of computer vision is understandable when considering the problems of classical methods.

Metallic defect detection is a challenging problem as metal surfaces are easily affected by environmental factors such as lighting and light reflection. Since traditional detection algorithms are inefficient in complex problems, we propose a novel method to detect and classify metal surface defects, such as cracks, scratches, inclusion, etc. The type and location of defects were detected by the Faster Regional Convolutional Neural Network (Faster R-CNN), combined with the Shape From Shading (SFS) method, which can extract surface characteristics.

The developed method aims to combine the powerful capabilities of SFS and Faster R-CNN methods. It also aims to automate the training stages by automatically labeling the defect regions. Initially, the SFS method and image processing techniques are used to detect defect areas in metal surface images using depth maps. Using the Faster R-CNN method, a defect detection model is trained on labeled data and automatically classifies and detects the defective regions.

After the training images are inputted into the SFS method, depth maps are obtained using image gradients, and defect regions in the image are detected. The minimum and maximum positions of all objects (defect regions) are determined for each mask.

The bounding boxes (xmin, xmax, ymin, ymax) of the defect regions and their corresponding width and height values are saved in data structures. The image and all the information related to the defect regions are then converted to the Pascal VOC XML format. Additionally, a comparison strategy is applied after creating XML files for each image. This comparison step is added to examine errors detected both outside and inside the ground truth. The original ground truth bounding boxes provided by the datasets and the bounding boxes generated using SFS are compared, and the accuracy of the model is analyzed. By processing the labels of defect regions and input images using Faster R-CNN during the training stages, a model is created for detecting errors.

The Northeastern University (NEU) surface defect database was used for defective samples. The proposed algorithm has also been tested on an unlabeled dataset (KolektorSDD2/KSDD2) to show labeling performance. The results on both labeled and unlabeled datasets have demonstrated state-of-the-art performance in automatic defect detection, classification, and labeling. The proposed method has satisfactory results for the detection of defects on the metal surface, and the mean average precision is 0.83. The average precision of crazing, pitted surface, patches, scratches, inclusion, and rolled-in scale are 0.98, 0.81, 0.90, 0.79, 0.88, and 0.62 respectively.

1. GİRİŞ

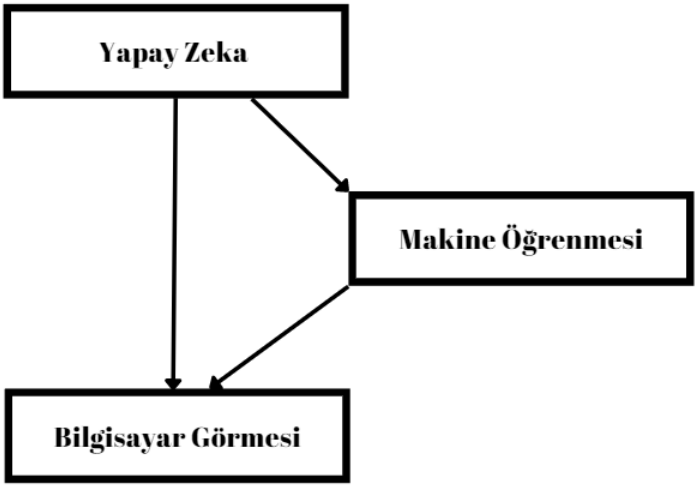
Veri toplamak ve veriye erişim eski zamanlarda zorken, günümüzde teknolojinin yaygınlaşması ile birlikte elektronik postalar, sosyal medya verileri, elektronik alışverişler vs. sayesinde üretilen veri miktarı artmıştır. Bu veri yığınları makine öğrenmesi yöntemleri kullanılarak anlamlı hale getirilirek sınıflandırılmalar yapılabilmektedir. Veri bilimciler, mevcut sistemlerden daha başarılı sonuç veren yapay zeka teknikleri geliştirmeyi hedeflemişlerdir. İlk defa insan sinir sisteminden ilham alınarak beyin fonksiyonlarının işleyişini mantıksal olarak hesaplayan bir model McCulloch-Pitts tarafından oluşturulmuştur [1].

Yapay zeka literatüründe derin öğrenme kullanılarak çok başarılı sonuçlar alınmaya başlanmıştır. Derin öğrenme, insan beyninin işleyişini rol alan ve problem çözümlerine yeni yaklaşımlar oluşturan yapıdır. Derin öğrenmeyi geleneksel sinir ağlarından ayıran en temel özellik ilse birden fazla gizli katmana sahip olmasıdır [2]. Endüstriyel kalite kontrol sistemleri, görüntü, ses analizleri, robotik, otonom sistemler ve tıp alanı gibi birçok alanda yüksek başarı oranı elde etmesi sebebiyle günümüzde derin öğrenme yaygın olarak kullanılmaktadır. Derin öğrenme, görüntü işleme problemlerinde de sıklıkla kullanılmaktadır.

1.1. Yapay Zeka

İnsan beyni yaklaşık yüz milyar nöronu birbirine bağlayan yüz trilyon sinaps ile en karmaşık organımızdır. Beynin düşünme, akıl yürütme, kavrama, nesneleri algılama ve sonuç çıkarma gibi yeteneklerinin dijitalleştirme isteği, bilimsel çalışmaları teşvik etmiştir. 1956 yılında Yapay Zeka'nın öncülerinden olan M. Minsky, A. Newell ve H. Simon'un katılmış olduğu konferans, yapay zekanın doğuşu olarak kabul edilmektedir [3]. Yapay zeka, insan zekasına ait özellikleri sergileyerek bilgisayar sistemlerinin düşünme, öğrenme ve sonuç çıkarma gibi işlemleri gerçekleştirmesini sağlayan bir teknolojidir. Bu teknoloji, insanların düşüncelerini taklit ederek ve geçmiş deneyimlerden elde edilen sonuçları kullanarak öğrenme sürecini gerçekleştirir. Genel olarak AI (Artificial Intelligence) olarak belirtilen yapay zeka son yılların en popüler konusudur. 1997 yılında Deep Blue isimli bilgisayarın, Dünya Satranç Şampiyonu

Kasporov'u yenmesiyle birlikte yapay zeka adından bahsettirmeye başlamıştır. 2014 yılında Google sürücüsüz otomobil prototipini test sürüşüne çıkarmıştır. 2011 yılında yapay zeka kullanan ilk kişisel asistanı olan Siri, Apple tarafından ios işletim sistemli akıllı telefonlarda kullanımına sunulmuştur. 2016 yılında ise Google'ın yapay zeka firmalarından olan DeepMind'in geliştirdiği AlphaGo adlı yapay zekanın, dünya şampiyonunu yenmesi yapay zekayı popüler hale getirmiştir. Yapay zeka ve bilgisayarlı görme ilişkisine genel bakış Şekil 1.1'de gösterilmiştir.



Şekil 1.1. Yapay zeka ve bilgisayarlı görme ilişkisine genel bakış

Bilgisayarların artan kapasitesi ile birlikte, doğal dil oluşturma, konuşma algılama, sanal aracılar, karar yönetimi, derin öğrenme platformları, biyometrik, robotik süreç otomasyonu, metin analizi ve NLP gibi alanlarda kullanılmaktadır.

1.2. Makine Öğrenmesi

Yapay zeka ve istatiğe dayanan, veri kümelerinden bilgi üretmemizi sağlayan araştırma dalına makine öğrenmesi (ML) denir. ML, matematiksel ve istatistiksel yaklaşımları kullanarak, mevcut verilerden sonuçlar çıkarmak ve bilinmeyen veriler hakkında tahminlerde bulunmak için modelleme ve algoritmaları kullanır. Bu yapay zeka teknolojisinin temel amacı, doğru tahminler yapmaktır.

Günümüzde bilgisayarların donanımsal yönü çok daha güçlü özellilklere sahiptir. Bilgisayar donanım özelliklerinin gelişmesi ve veri sayısının çoğalmasıyla veri analizi için daha başarılı modeller oluşturulmaktadır. Veri toplamak ve veriye erişim eski

zamanlarda zorken günümüzde, teknolojinin yaygınlaşması ile birlikte üretilen veri miktarı artmıştır. Toplanan bu veriler, analiz edildiğinde, önemli sonuçlar elde edilmektedir. Veriler arasında ilgili veya ilgisiz bazı özellikler vardır bu verileri kullanarak anlamlı sonuç çıkarma makine öğrenmesi sayesinde yapılmaktadır.

Yazılımda, bir görevi yapmak için açıkça programlamak yerine, görevi yapmayı öğrenebilen programlar geliştirme yaklaşımı kullanılmaktadır. Bir sorunu çözmek için bir algoritmaya ihtiyaç duyarız ve algoritma, giriş verisini kullanarak çıkış verisine dönüştürmeyi sağlayan talimatlar sırasıdır. Geleneksel programlama ve makine öğrenmesi arasındaki fark, temelde bir programın nasıl yazıldığı ve çalıştırıldığıdır. Geleneksel programlama, insanların belirli bir problemi çözmek için yazdığı belirli bir dizi adımdan oluşan bir programlama yöntemidir. Bu adımlar, belirli bir girdiye dayanarak belirli bir çıktı üretmek için tasarlanmıştır ve genellikle bu adımlar, programcı tarafından belirlenir. Örneğin, bir kredi kartı işlemini onaylamak için yazılan bir programda, işlem adımları (örneğin, kredi kartı numarasını kontrol etme, hesap bakiyesini kontrol etme vb.) önceden belirlenir ve programcı tarafından kodlanır.

Makine öğrenmesi ise, bir programın verilerden öğrenerek kendisini geliştirmesi ve sonuçları tahmin etmek için kullanılmasıdır. Makine öğrenmesi algoritmaları, belirli bir veri kümesi üzerinde çalışarak öğrenme yapar ve sonunda bir tahmin modeli oluşturur. Bu tahmin modeli, yeni girdilere dayanarak doğru sonuçlar üretmek için kullanılabilir. Örneğin, bir makine öğrenmesi modeli kullanılarak bir fotoğraftaki nesnelerin tanınması mümkündür. Makine öğrenmesi, programcıların her olası senaryoyu önceden tahmin etmesi yerine, verileri kullanarak doğru sonuçları tahmin etmek için bir model oluşturarak daha esnek bir yaklaşım sunar.

Özetle, geleneksel programlama belirli bir problemi çözmek için belirli adımların programcı tarafından kodlandığı bir yöntemdir. Makine öğrenmesi ise verilerden öğrenerek kendisini geliştirir ve sonuçları tahmin etmek için bir model oluşturur.

Optik karakter tanıma, yüz tanıma, istenmeyen e-posta filtreleme, konuşulan dili anlama, tıbbi teşhis, tüketici segmentasyonu, dolandırıcılık tespiti ve hava tahmini dahil olmak üzere birçok çeşitli sorun ML ile çözülür. Sonraki bölüm ML yöntemlerini kısaca açıklar. Tablo 1.1'de gösterildiği gibi bilinen dört makine öğrenmesi yöntemi vardır.

Tablo 1.1. Makine öğrenmesi çeşitleri [4].

Makine Öğrenmesi	Örnek
Denetimli Öğrenme	Ev fiyatı tahmini
Denetimsiz Öğrenme	Müşteri–Ürün segmentasyonu
Takviyeli Öğrenme	Otomatik sürüşlü araba

1.3. Bilgisayar Görmesi

Bilgisayar bilimlerinde gelişen teknolojilerle birlikte bilgisayar görmesi adı verilen bir alan ortaya çıkmıştır. Bilgisayar görmesi (Computer Vision - CV), görüntü ve video gibi görsel sayısal girdilerden insan görüsü yeteneklerinin taklit edilerek sistemlere anlamlı bilgiler kazandırılmasına denir [5]. Sistemlerin kazandığı bilgileri kullanarak önerilerde bulunan ve eylemler gerçekleştirebilen bir bilgisayar bilim alanıdır.

Bilgisayar görmesi, kalite kontrol sistemlerinde yaygın bir şekilde kullanılan bir teknolojidir. Bu sistemler, ürünlerin kalitesini kontrol etmek için bilgisayarın görüş yeteneğinden yararlanarak üretim hattındaki hataları tespit etmek için kullanılır. Örneğin, bir fabrika üretim hattındaki ürünlerin kalitesini kontrol etmek için bilgisayar görüş sistemlerini kullanabilir. Bu sistemler, üretim hattındaki ürünlerin görüntülerini alarak, belirli kalite kriterlerine göre kontrol edebilirler. Sistem, önceden belirlenmiş bir kalite standardına uygun olmayan ürünleri otomatik olarak reddedebilir veya uygun olmayan ürünlerin üretimini durdurabilir.

Bilgisayar görmesi teknolojisi, kalite kontrol sistemlerinde insan hatalarını azaltmak ve üretim hatlarında tutarlılığı sağlamak için çok önemli bir rol oynamaktadır. Sayısal görüntülerden faydalı bilgiler çıkarmaya dayanan bilgisayar görmesi, endüstriyel kalite kontrolü, ulaşım, jeoloji, otonom araçlar, robotik ve sanal gerçeklik gibi birçok alanda kullanılmaktadır. Bilgisayar görmesi uygulamalarına örnek Tablo 1.2'de gösterilmiştir.

Tablo 1.2. Bazı güncel bilgisayar görmesi uygulamaları [6].

Uygulama	Örnek kullanım
Optik karakter tanıma	Otomatik plaka tanıma
Makine muayenesi	Metal yüzeylerde kusur arama
Satış	Otomatik ödeme şeritleri için nesne tanıma
3B model oluşturma	Havadan çekilen fotoğraflardan 3B model oluşturma
Tıbbi görüntüleme	Ameliyat öncesi ve sonrası görüntülerin kaydedilmesi
Otomotiv güvenliği	Beklenmedik engelleri tespit etme
Hareket eşleştirme	Bilgisayar tarafından oluşturulan görüntüleri canlı çekim görüntüleri ile birleştirme
Parmak izi tanıma, biyometri	Adli uygulamaların yanı sıra otomatik erişim yetkisi verme

Yapay zeka ve makine öğreniminin bir alt alanı olan bilgisayar görmesi, farklı algoritmalardan yararlanabilen çok disiplinli bir alandır. Görüntü üzerindeki bilgisayar görmesi uygulamaları:

1. Nesne Sınıflandırma
2. Nesne Tanımlama
3. Nesne Doğrulama
4. Nesne Tespiti
5. Nesne Yer Tespiti
6. Nesne Segmentasyonu

Endüstriyel kontrol sistemlerinde oldukça önemli bir yer tutan bilgisayarlı görme ve görüntü işleme yaklaşımları özellikle kalite kontrol sistemlerinde üretim hattında üretilen ürünlerin hatasız ve hızlı bir şekilde kontrolünde sıklıkla kullanılmaktadır. Bilgisayar Görmesinin kalite kontrol sistemlerinde kullanıldığı yerler:

1. Öngörücü bakım: Yapay zeka, sensörler ve diğer kaynaklardan gelen verilere dayanarak ekipmanın ne zaman arıza yapacağını veya bakıma ihtiyaç duyacağını öngörmek için kullanılabilir. Bu, süre kaybını azaltmaya ve verimliliği artırmaya yardımcı olabilir.
2. Süreç optimizasyonu: Yapay zeka, sensörler ve diğer kaynaklardan gelen verileri analiz ederek üretim süreçlerini optimize etmek için kullanılabilir. Bu, verimliliği artırmaya yardımcı olur.
3. Kalite öngörüsü: Yapay zeka, önceki üretim çalışmalarından ve diğer kaynaklardan gelen verilere dayanarak ürünlerin kalitesini üretmeden önce öngörebilir. Bu, yalnızca yüksek kaliteli ürünlerin üretildiğinden emin olmak için yardımcı olabilir.
4. Görüntü analizi: Yapay zeka, ürünlerin görüntülerini analiz ederek kusurları ve diğer kalite sorunlarını tespit etmek için kullanılabilir. Bu görsel incelemenin önemli olduğu gıda işleme veya ilaç endüstrilerinde kullanışlı olabilir.

Klasik metotlarla yapılan kalite kontrol çalışmalarında, yorgunluk, dalgınlık vb. insandan kaynaklanan hatalar sıklıkla olmaktadır. Bilgisayarlı görü ile bu hataların önüne geçilmesini sağladığı için endüstriyel kontrol sistemlerinde önemli bir yer tutmaktadır.

Bu çalışmada metal parçalarının üzerindeki kusur Shape From Shading algoritması ile etiketlenip daha sonra derin öğrenme algoritmalarından Faster-R-CNN kullanılarak hatalı bölgelerin yeri tespit edilmiştir. Precision, mAP değerleri ile sonuçların performans karşılaştırılması yapılmıştır.

Tez içeriğinin geri kalanı ise şu şekilde özetlenebilir: 2. Bölüm'de metaller üzerindeki kusur tespiti ile ilgili literatür çalışması yapılmıştır. 3. Bölüm'de derin öğrenme, 4. Bölüm'de nesne tanıma algoritmaları açıklanmıştır. 5. Bölüm'de derin öğrenme –SFS kullanılarak geliştirilen uygulamadan bahsedilmiştir. 6. Bölüm olan son bölümde ise yapılan çalışmaların sonuçları tartışılmış ve önerilerde bulunulmuştur.

2. LİTERATÜR ÖZETİ

2.1. Metal Üzerindeki Kusur Tespiti ile İlgili Yapılan Çalışmalar

Son yıllarda üretim aşamasında otomatik hata tespit sistemlerine olan ihtiyaç artmaktadır. Ürünlerin kusurlarını tespit etmek ve yerlerini belirlemek, önemli ve gerekli bir kalite kontrol sürecidir. Arıza tipinin ve arızalı alanın kısa sürede tespit edilmesi de kalite kontrol performansı açısından oldukça önemlidir.

Metal yüzeylerdeki kusurlar, üretim sürecinde veya sonrasında oluşabilir. Örneğin, kaynak hatası, çatlaklar, delikler, erozyon, paslanma, korozyon, malzeme yorgunluğu gibi birçok nedenle metal yüzeylerde kusurlar oluşabilir.

Kusur tespit çalışmaları, metal yüzeylerdeki kusurları belirlemek, büyüklüklerini ölçmek ve bunların nedenlerini araştırmak için yapılır. Bu çalışmalar, metal yüzeylerin kullanım ömrünü artırmak, güvenilirliğini sağlamak ve maliyetleri azaltmak için önemlidir.

Bugün, insana dayalı kalite kontrol, üretim sürecinde geleneksel bir yöntem olarak kusurları tespit etmek için hala kullanılmaktadır. İnsana bağlı olan bu yöntemde hız oranı düşük ve hata oranı yüksektir. Üretim aşamasındaki yüzeylerin kalite kontrol çalışmalarında bilgisayarla görme teknikleri sıklıkla kullanılmaktadır.

Birçok endüstriyel uygulamada, tekstil, metal ve cam kusur tespiti gibi yüzey kusur tespiti yaygın olarak gerçekleşir. Metal yüzeyler aydınlatma ve ışık yansıması gibi birçok çevresel faktörden kolayca etkilendiğinden, metalik kusur tespiti zor bir problemdir. Metal yüzeylerde çeşitli ve karmaşık tipte kusurlar (çatlaklar, yamalar, çizikler, noktalı yüzeyler vb.) vardır. Gerçek zamanlı metalik hata tespit sistemlerinde hız ve yüksek doğruluk üretim aşamasına olumlu etki yapmaktadır.

Yüzeylerde kusur tespiti için geleneksel yöntemler, görüntü işleme veya makine öğrenmesi tekniklerine dayanır, ancak belirli ölçeklerde veya belirli aydınlatma koşullarında düşük gürültü ve güçlü kontrast ile farklı kusurları tespit edebilirler [7].

Birçok endüstriyel uygulamada, tekstil, metalik ve cam gibi yüzeylerde yaygın olarak kusurlar meydana gelir. Metal yüzeyler, aydınlatma ve ışığın yansıması gibi çevresel

faktörlerden kolayca etkilendiğinden, metalik kusur tespiti karmaşık bir problemdir. Gerçek zamanlı metalik hata tespit sistemlerinde hızlılık ve yüksek doğruluk, üretim aşamasında oldukça önemlidir.

Yüzey hatası tespiti için çeşitli görüntü işleme yöntemleri kullanılmaktadır. Doku özelliklerini tanımlamak için kullanılan oto korelasyon yöntemi [8], gri seviyeli birlikte oluşum matrisi yöntemi [9], morfoloji yöntemi [10] ve histogram öznitelik istatistikleri [11], görüntünün dokusal yapısını bulan Fourier, Gabor ve dalgacık öznitelik yöntemi [12-14], diğer özellikleri belirli modellerle modelleyerek doku modellerini tanımlamak için kullanılan fraktal gövde modeli [15], geri saçılma modeli [16], ve rastgele alan modeli [17], yüzey kusur tespiti için kullanılan görüntü işleme yöntemleridir. Bu geleneksel yöntemler, karmaşık dokuları veya herhangi bir kusur türünü içeren doku kusurlarını tespit edemez. Bayes ağ sınıflandırıcıları [18], Temel Bileşen Analizi (PCA) [19], Destek Vektör Makineleri gibi çeşitli makine öğrenme teknikleri (SVM) [20], Rastgele orman [21], ve Kendi Kendini Düzenleyen Haritalar (SOM) [22], da yüzey kusurlarını tespit etmek ve sınıflandırmak için kullanılmıştır.

Geleneksel görüntü işleme yöntemleri, genellikle arka plan renklerine ve aydınlatma koşullarına çok duyarlı olan algoritmalardaki çeşitli kusurları hedefleyen birden çok eşik gerektirir ve her bir problem için bu eşiklerin yeniden ayarlanması gerekir. Bu yüzden, günümüzde yapay zekanın gelişmesiyle birlikte CNN (Convolutional Neural Network), yüzey kusurlarının etiketlenmiş görüntülerinden doğrudan öğrenebilen ve yüksek bir tanıma oranı elde edebilen kusur tespiti için yaygın olarak kullanılmaktadır. Lin ve ark. [23], kusurları tespit etmek için MobileNet-v2-dense adlı çok ölçekli bir ardışık CNN önerdi. Yi ve ark. [24], CNN'ye dayalı çelik şerit yüzeyleri için bir kusur tespit sistemi önerdi. Kusurlu alanı tespit etmek için belirginlik haritası ve görüntü segmentasyonu kullanıldı. Kim ve ark. [25], az sayıda görüntü ile kusurları tespit edebilen CNN yapısını kullandı. Zhou ve ark. [26] CNN kullanarak sıcak haddelenmiş çelik levhalar üzerindeki yüzey kusurlarının sınıflandırılması üzerine bir çalışma yürütmüştür. Farklı SNR değerlerine sahip görüntülere Gauss gürültüsü eklenerek sistemin sağlamlığı test edilmiştir. Soukup ve Huber-Mork [27], çelik ray yüzey kusur tespiti için fotometrik stereo görüntüleri kullanarak bir CNN sınıflandırması gerçekleştirdi. Ray yüzeyindeki karanlık alanlardaki boşluklar farklı renkli ışık kaynakları ile görünür hale getirildi. Amin ve Akhter [28] çelik yüzeylerdeki kusurları tespit etmek için U-NET ve Derin Kalıntı U-NET olmak üzere iki derin öğrenme

yöntemini kullanmış ve görüntüleri beş farklı sınıfa ayırmıştır. Lv ve ark. [29], metalik yüzey kusur tespiti için on kusur tipi içeren GC10-DET adlı bir veri seti oluşturdu. Ayrıca single shot multibox detector'a dayalı bir kusur tespit sistemi önerdiler. Önerilen yöntemi, NEU-DET ve GC10-DET veri kümelerini kullanarak klasik makine öğrenme yöntemleri ve derin öğrenme yöntemleriyle karşılaştırdılar. Li ve ark. [30], YOLO ağını kullanarak gerçek zamanlı çelik şerit yüzey kusur tespiti üzerine bir çalışma yürütmüştür. YOLO ağını geliştirdiler ve hepsini evrimsel hale getirdiler. Geliştirdikleri ağ 27 evrişim katmanı içeriyor. Fu ve ark. [31], transfer öğrenmeyi kullanarak çelik şerit yüzey kusur tespiti üzerine bir çalışma yürütmüştür.

Öznitelikleri çıkarmak için önceden eğitilmiş VGG16'yı ve sınıflandırma için CNN'yi kullandılar. Ayrıca görüntülere farklı SNR seviyelerinde Gauss gürültüsü ekleyerek doğruluk analizi yaptılar. Lee ve ark. [32] için bir yaklaşım önerdi sınıf aktivasyon haritaları ile bir CNN kullanarak çelik kusurlarını tespit edilerek, basit bir sınıflandırma görevi yerine gerçek zamanlı bir görsel süreci desteklemek için CNN hata tespit modelini genişlettiler.

Derin öğrenmeye dayalı çelik yüzey kusur tespit yöntemi, çelik üretiminde yaygın olarak kullanılmaktadır ve bu konuda daha iyi sonuçlar elde edilmek için çalışmalar devam etmektedir [33-34]. Fu ve ark. [35], çelik yüzey kusurlarının hızlı ve doğru sınıflandırmasını gerçekleştiren bir evrişimli sinir ağı modeli önermiştir. Lv ve ark. [36], farklı tipteki çelik yüzey kusurları için bir uçtan uca kusur tespit ağı (EDDN) önerdi. Marco ve ark. [37], çelik kusur sınıflandırmasında geleneksel makine öğrenmesi modeli ve derin öğrenme modelini karşılaştırarak geliştirilmiş çelik yüzey kusur tespiti ve sınıflandırma yöntemlerini tartıştı. Song ve ark. [38], kodlayıcı-kod çözücü artık ağına (EDR-Net) dayalı bir belirginlik algılama yöntemi önermiştir. Kodlama aşamasında, kusur özelliklerini çıkarmak için tam evrişimli sinir ağı kullanılır ve modelin yakınsamasını hızlandırmak için bir dikkat mekanizması entegre edilerek, ana hedef tespit çerçevesi olarak, Faster R-CNN, çelik ve metalin yüzey kusuru tespitinde yaygın olarak kullanılmaktadır. Wang ve ark. [39], metal plaka ve şerit yüzeyindeki çeşitli ve rastgele kusurların tespiti problemini çözen, çok seviyeli özellikleri entegre eden Faster R-CNN algoritması önerdi. Dai ve ark. [40], geleneksel yöntemlerle karşılaştırıldığında kusurların tespit performansını artıran iş parçası yüzey kusur tespitinin sınırlamaları ve düşük hassasiyet sorunlarını çözmek için geliştirilmiş Faster R-CNN'ye dayalı bir kusur tespit algoritması tasarlamıştır.

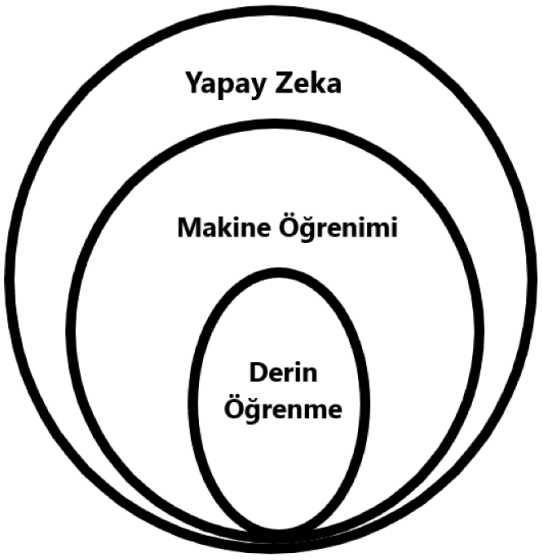
Cheng ve Wang [41], drenaj borularının hasar tespiti için Faster R-CNN'yi uygulayarak %83 mAP elde etti. Li ve ark. [42], ZF-Net'i Faster-RCNN kullanarakve kusur tespiti için ağın başına bir max-pooling katmanı ekleyerek %80,7 mAP'ye ulaştı. Zhang ve ark. [43], iyi bir performans elde eden YOLOV3'ü köprü yüzey hasarını tespit etmek için kullandı. Yin ve ark. [44] kanalizasyon boru hattı hasar kusurlarını tespit etmek için YOLOV3'ü kullandı ve %85.37 mAP elde etti. Deng ve ark. [45], karmaşık arka plan altında beton bir yüzeydeki çatlakları ve kusurları tespit etmek için YOLOV2'yi kullandı.

Bu çalışmada, metal bileşenlerin yüzeylerinde oluşabilecek çeşitli kusur türlerinin tespit edilmesi için uçtan uca otonom bir sistem amaçlanmıştır. Ayrıca, NEU veri setinden alınan görüntülerin kusurlu alanlarının otomatik olarak etiketlenmesi, yüzey özelliklerini çıkarabilen SFS algoritması yöntemiyle yapılmıştır. Böylece hata konumlarının etiketlenmesi aşamalarında yorgunluk, devamsızlık gibi insan hatalarının önüne geçilmesi amaçlanmış ve hata konumlarının daha hassas etiketlenmesi sağlanmıştır. SFS'den alınan görüntüler, son yıllarda popüler olan ve oldukça başarılı sonuçlar veren Faster R-CNN ile sınıflandırılmıştır.

3. DERİN ÖĞRENME

Genellikle AI (Artificial Intelligence) olarak adlandırılan yapay zeka, makinelere insan zekası verme sürecidir. Yapay Zekanın temel amacı, insanlar gibi düşünebilen ve hareket edebilen kendine güvenen makineler geliştirmektir. Bu makineler, insan davranışını taklit edip, öğrenerek ve problem çözerek görevleri gerçekleştirebilir. Yapay zeka sistemlerinin çoğu, karmaşık sorunları çözmek için doğal zekayı simüle eder. Yapay zeka, bir bilgisayar programı veya algoritma kullanarak verileri analiz etmek, tahmin yapmak, öğrenmek ve karar vermek gibi insan zekasının belirli yönlerini taklit etmeye çalışır. Büyük miktarda veri kullanarak öğrenirler ve daha sonra bu öğrenilen bilgileri kullanarak kararlar verirler. Yapay zeka teknolojileri, otomasyon, tahmin, öneri ve problem çözme gibi birçok alanda kullanılmaktadır. Örneğin, konuşma tanıma sistemleri, sesli asistanlar, görüntü tanıma sistemleri, akıllı ev sistemleri ve finansal tahmin modelleri gibi birçok uygulama alanı vardır.

Makine öğrenmesi, iş sorunlarını çözebilecek tahmine dayalı modeller oluşturmak için bilgisayar algoritmalarını ve analitiğini kullanan bir bilgisayar bilimi disiplinidir. Yapay zeka, makine öğrenimi ve derin öğrenme arasındaki ilişki Şekil 3.1'de gösterilmiştir.



Şekil 3.1. Yapay zeka, makine öğrenimi ve derin öğrenme arasındaki ilişki

Derin öğrenme ise, insan beyninin yapısı ve işlevinden ilham alan algoritmalarla ilgilenen bir makine öğrenmesi alt kümesidir. Derin öğrenme algoritmaları hem yapılandırılmış hem de yapılandırılmamış çok büyük miktarda veriyle çalışabilir. Derin öğrenmenin temel konsepti, makinelerin karar vermesini sağlayan yapay sinir ağlarında (YSA, Artificial Neural Network, ANN) yatmaktadır. Derin öğrenme, yapay zeka alanında kullanılan bir makine öğrenmesi yöntemidir. Bu yöntem, çok katmanlı yapay sinir ağları kullanarak verileri otomatik olarak öğrenir ve daha sonra bu öğrenilen bilgileri kullanarak yeni verileri analiz etmek ve kararlar vermek için kullanır. Derin öğrenme, verilerin analiz edilmesi ve öğrenilmesi için birçok katmanlı sinir ağı kullanarak özellikleri ve örüntüleri otomatik olarak tanımlamayı öğrenir. Öğrenme süreci, büyük miktarda veri kullanılarak gerçekleştirilir ve bu veriler, genellikle yapay sinir ağlarının eğitimi sırasında kullanılır.

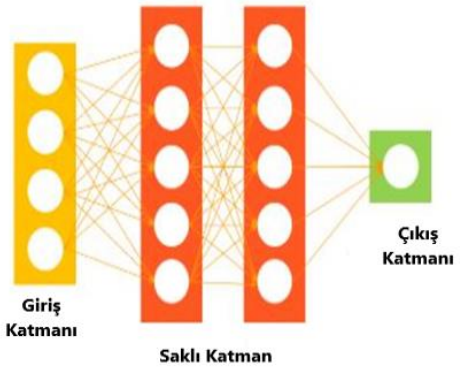
Derin öğrenme, birçok uygulama alanında kullanılır. Örneğin, görüntü tanıma, konuşma tanıma, doğal dil işleme, robotik ve oyunlar gibi alanlarda kullanılır. Derin öğrenme algoritmaları, verilerin analiz edilmesi ve öğrenilmesi için insan müdahalesine ihtiyaç duymazlar ve bu nedenle çok sayıda veri seti kullanarak öğrenme yapabilirler.

Makine öğrenmesinin bir alt kümesi olan derin öğrenme, birden çok katmanlı yapay sinir ağlarını kullanarak verileri öğrenmek ve analiz etmek için kullanılanılır. Bu sinir ağları, verilerin özelliklerini otomatik olarak keşfedip temsil edebildikleri için büyük ve karmaşık veri kümelerinden öğrenme ve tahmin yapmayı mümkün kılarlar. Geleneksel makine öğrenmesi algoritmalarının aksine, derin öğrenme algoritmaları, ön işlem yapmaya ihtiyaç duymadan bu özellikleri otomatik olarak ham verilerden öğrenebilirler.

Derin öğrenme algoritmaları, insan beyninin yapısından esinlenen yapay sinir ağlarına dayanır. Bu ağlar, birbirine bağlı düğümlerden veya nöronlardan oluşur ve veri girdisi ağdan geçerken işlenip dönüştürülür. Ağın her katmanı, verilerin farklı özelliklerini tanımayı öğrenir ve bir katmanın çıktısı bir sonraki katmanın girdisi olarak kullanılır, böylece ağ verilerin giderek daha karmaşık temsillerini öğrenir.

Son yıllarda, geriye yayılım gibi derin sinir ağlarının eğitiminde tekniklerin geliştirilmesi nedeniyle derin öğrenme yaklaşımı giderek popüler hale gelmiştir.

Görüntü ve ses tanıma, doğal dil işleme, otonom sürüş ve daha birçok uygulama için kullanılmıştır. Basit bir sinir ağı Şekil 3.2'de gösterilmiştir.



Şekil 3.2. Sinir ağı yapısı

Derin öğrenme, doğrudan veri öğrenme tekniği kullanarak ses, görüntü veya metin gibi farklı veri tipleri üzerinde makine öğrenmesi yapar. Derin öğrenme, yapay zeka ve makine öğrenmesinin en altındaki sistemli yaklaşımdır ve en popüler uygulamalarından biridir [46].

Öğrenme işlemi sırasında, etiketlenmiş bir örüntüdeki tüm pikseller giriş verisi olarak kullanılır. Görüntü üzerindeki özellik haritası elde etmek için konvolüsyon, relu ve havuzlama katmanları uygulanır. Daha sonra, tam bağlantılı katman kullanılarak her pikselin her sınıf için ihtimal değerleri hesaplanır ve ilgili pikselin hangi sınıfa ait olduğu belirlenir [47]. Yapay Sinir Ağlarından, derin öğrenme algoritmalarını ayıran en önemli fark, özellik haritasını, kendi içerisinde yer alan konvolüsyonel sinir ağları ile yapmasıdır [48]. Farklı sayıda farklı filtreler uygulanarak, konvolüsyonel sinir ağları ile özellik haritaları oluşturulur ve bir test görüntüsü için uygulandığında, hangi görüntüye ait olduğu tahmin edilmektedir [49].

Öznitelik çıkarımı geleneksel makine öğrenmesi yöntemlerinde zor bir problemdir. Özniteliklerin belirlenmesi için alanında uzman kişilere problemin çözümünde ihtiyaç duyulmaktadır ve bu işlem oldukça zaman almaktadır [50]. Derin öğrenmeyi geleneksel makine öğrenmesi yöntemlerinden ayıran özelliklerden bir tanesi öznitelik çıkarımını ağın kendi içerisinde yapılmasıdır ve böylece ön işlem olan öznitelik çıkarılma problemleri ortadan kaldırılmıştır. Eğitim süresi boyunca derin öğrenme, girdi ve çıktı verileri arasındaki ilişkilerin etkin öğrenimini sağlar [51].

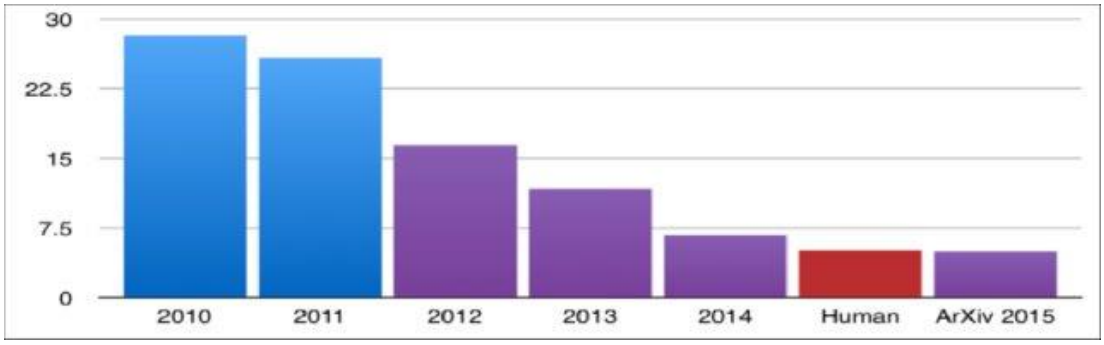
Veri sayısının fazla olmaması ve gerekli donanımlara sahip olunmamasından dolayı, derin öğrenmenin temelleri aslında geçmişe dayanmasına rağmen popülerliği son yıllarda artmıştır [50]. Büyük veri (big data) kavramı, veri kaynaklarının artmasıyla birlikte günümüzde bir alan oluşturmuştur. Paralel işlemcilerin kullanımı, çok fazla gizli katmanlı ağların eğitimindeki çözülmesi zor hesaplamaları çok daha kolay hale gelmesini sağlamıştır [50].

Yapay zekanın önemli konularından birisi olan insan karar verme yetisini modelleyen sistemler üretmektir. McCulloch ve ark. [52], yapay sinir ağlarının temeli olarak kabul edilen ve insan sinir sisteminden esinlerek beyin fonksiyonlarının işleyişini hesaplayan bir model üretilmiştir.

Adaptive linear element (ADALINE) [53] ve Perceptron [54] modelleri ise daha sonraki çalışmalarda üretilmiştir. Fakat XOR gibi doğrusal olmayan problemlere çözüm üretememesinden [55], ve sadece doğrusal problemlere çözüm üretilmesinden dolayı yapay sinir ağlarına olan ilgi zamanla azalmıştır [50].

Paralel dağıtık işlem yaklaşımları [56], [57] ile yapay Sinir ağları , 1980'lerde yeniden popülerlik kazanmıştır.Derin sinir ağları ile başarılı bir öğrenme sağlanacağını 2006 yılında Hinton ve ark. [58] göstermiştir. Devamında yapılan çalışmalar ile derin öğrenme kavramı daha da yagınlaşmış ve başarılı sonuçlar elde edilmesi için daha derin ağların oluşturulması gerektiği belirtilmiştir [59- 60].

Nesne tanıma yarışmalarından olan ImageNet , derin öğrenmenin temel mimarisi CNN ile 2012 yılında kazanılmıştır ve bu yarışma ile birlikte çok başarılı sonuçlar verdiği için CNN kullanımı yaygınlaşmıştır [61]. Şekil 3.3'te ImageNet yarışmasının yıllara göre hata oranı gösterilmiştir.



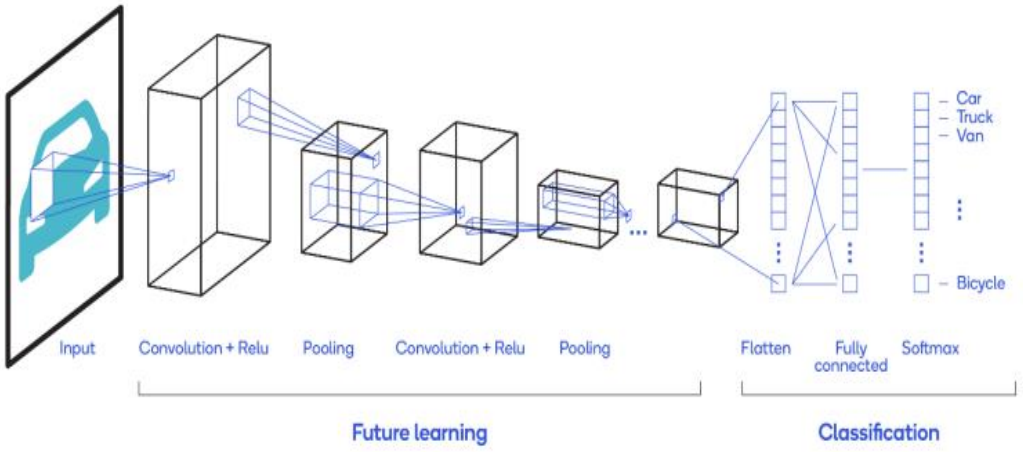
Şekil 3.3. ImageNet yarışmasının yıllara göre hata oranları [62].

Derin öğrenme modellerinden CNN, literatürde görüntü analizinde sıklıkla kullanılan ve özellikle büyük boyutlu verilerde başarılı sonuçlar üreten popüler bir yaklaşımdır [63], [64]. CNN'de 2 boyutlu çekirdek çerçevelerin kullanımı ile hiyerarşik olarak öğrenme gerçekleşir. CNN ile genel olarak, ardışık yerleştirilmiş eğitim bölümlerinin sonuna bir sınıflandırıcı yerleştirilmesi ile sınıflandırma işlemi gerçekleştirilebilir. CNN sınıflandırma amacıyla kullanılabildiği gibi, regresyon işlemlerini de başarıyla gerçekleştirir.

3.1. CNN Oluşturan Katmanlar

Evrişim işlemi, giriş olarak verilen görüntü, sesten ya da metinden bilgilerin veya özelliklerin çıkarılmasını hedefler. Bu giriş verileri, bir matris olarak düşünülebilir ve evrişim işlemi, belirli filtreler kullanarak belli adımlarda bu matrisi tarayarak istenilen giriş nesnesi için belirleyici özellikler çıkarmayı amaçlar. Evrişim işleminde kullanılacak olan filtrenin x ve y eksenlerine göre simetrisi alınır ve filtrenin giriş görüntüsü üzerinde adım uzunluğuna bağlı olarak gezdirilmesiyle, her adımda çakışan değerler eleman eleman çarpılarak toplanır ve bu değerler, çıkış matrisinin değerleri olarak kaydedilir. Giriş verisinin kanal sayısı ve formatına bağlı olarak, işlemin basitliği değişebilir.

CNN giriş katmanı, konvolüsyon katmanı, havuzlama katmanı, aktivasyon katmanı, dropout katmanı, tam bağlantılı katman ve sınıflandırma gibi katmanlardan oluşur. Bu katmanlar Şekil 3.4'te gösterilmiştir.



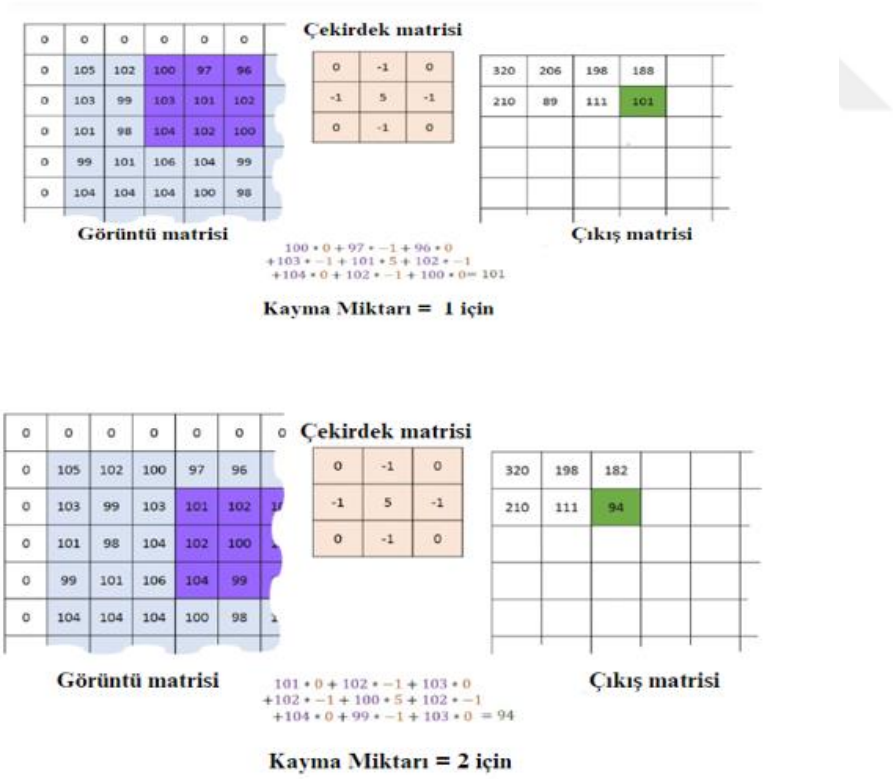
Şekil 3.4. Evrişimsel sinir ağlarını oluşturan katmanlar [65].

3.1.1. Giriş katmanı (Input layer)

CNN'nin ilk katmanıdır very giriş katmanı olarak da bilinmektedir. Sistemin başarısını, CNN'e verilecek verinin boyutu etki etmektedir. Giriş katmanına verilecek olan görüntünün boyutunun büyük olması eğitim süresini uzatabilir ve fazla bellek ihtiyacı gerektirse de eğitim sonuçlarını olumlu olarak etkileyebilir. Görüntünün küçük boyutta olması ise eğitim süresini azaltsa da eğitim sonuçlarını olumsuz olarak etkiler [50].

3.1.2. Konvolusyon katmanı (Convolution Layer)

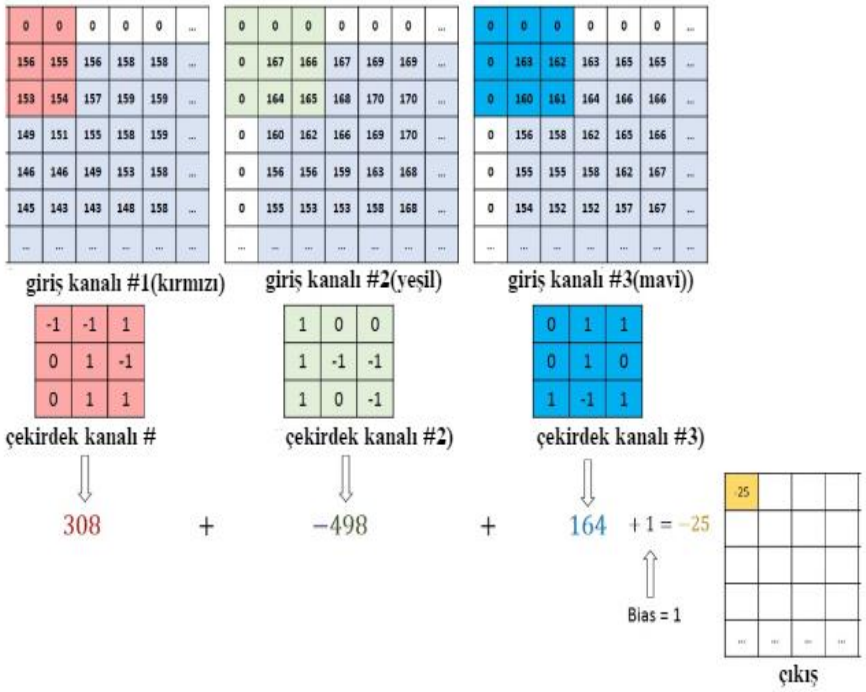
CNN temelini oluşturan katmandır ve bazı filtreler görüntü üzerinde dolaşarak konvolüsyon işlemini gerçekler. Bu katman sayesinde özellik haritaları oluşturulur ve her bir filtrenin oluşturduğu özellikler haritalanır. Farklı boyutta filtrelemeler kullanılarak konvolüsyon, derin öğrenme algoritmalarında yapılır. AlexNette 11x11 boyutunda matrisler yer alırken, Google Net'de 5x5, VggNet 'de 3x3, ResNet derin öğrenme mimarisinde ise 1x1 filtrelemler kullanılmştır.



Şekil 3.5. Konvolüsyon işleminin uygulanması (kayma miktarı: 1 ve kayma miktarı:2 için) [66].

Şekil 3.5 de görülen örnekte, bir giriş görüntüsünün üzerinde dolaşarak, görüntünün piksel değerleri ile çarpılması ve toplam değerlerin ilgili piksel kısmına yazılması işlemi görülmektedir. Matris sınırına gelindiğinde, filtrenin aşağıya 1 birim kaydırılarak işleme devam edilmesi ve kayma miktarı 1 ve 2 olarak elde edilen sonuçlar gösterilmektedir.

Şekil 3.6'da görüldüğü üzere, konvolüsyon işlemi renkli görüntülere uygulanırken, farklı renk kanallarına filtreler ayrı ayrı uygulanıp, bu değerlerin toplamı alınır ve aktivasyon haritasını oluşturur [50]. Stride (kayma miktarı) adı verilen parameter ile görüntü üzerinde sağa veya sola doğru ne kadar kaydırılması gerektiği bilgisi verilir.



Şekil 3.6. Üç kanallı bir görüntünün, üç kanallı bir filtre ile konvolüsyona tabi tutulması [67].

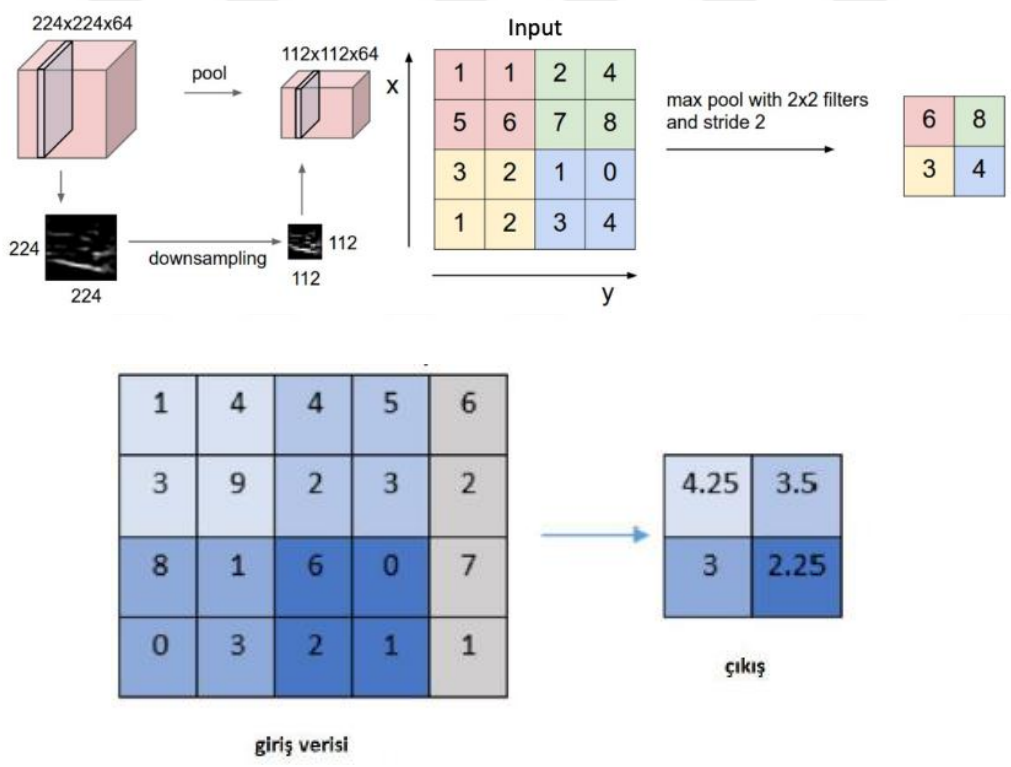
3.1.3. Aktivasyon katmanı

Konvolüsyon katmanından sonra aktivasyon katmanı gelir ve hiperbolik tanjant, sinüs, sigmoid vb. fonksiyonlar, aktivasyon fonksiyonu olarak kullanılmaktadır. Doğrusal olmayan relu ve sigmoid aktivasyon fonksiyonlarını kullanan konvolüsyonel sinir ağları daha hızlı bir şekilde eğitildiği bilinmektedir [68].

3.1.4. Havuzlama katmanı (Pooling Layer)

Giriş boyutunu azaltmak havuzlama katmanın temel amacıdır. Genellikle aktivasyon katmanından sonra yer alan havuzlama katmanında, giriş görüntüsünde yükseklik ve genişlik azaltılır fakat derinlik boyutunda herhangi bir değişiklik gözlenmez ve bu katman zorunlu bir katman değildir. Derin öğrenme mimarisinde, tasarıma bağlı olarak oluşturulabilir. Kendinden sonraki diğer katmanlar için hesaplama yükünün azaltılmasını sağlar [69].

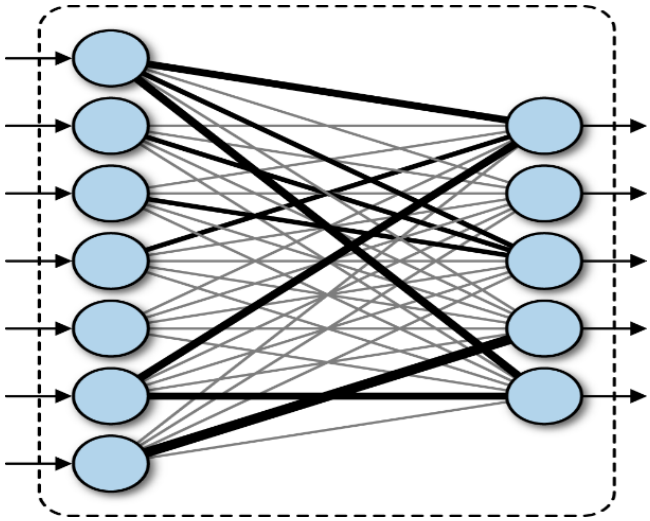
Maksimum havuzlama (max-pooling) ve Ortalama havuzlama (average pooling) olarak Şekil 3.7'deki gibi 2 şekilde gerçekletirilebilir. Ortalama havuzlamada filtre içerisindeki piksellerin ortalaması alınırken, maksimum havuzlamada ise filtredeki görüntüde bulunan piksellerin maksimum değeri çıkış pikseli olarak belirlenir.



Şekil 3.7. Maksimum havuzlama ve ortalama havuzlama işlemine dair bir örnek [70].

3.1.5. Tam bağlantılı katman (Fully connected layer)

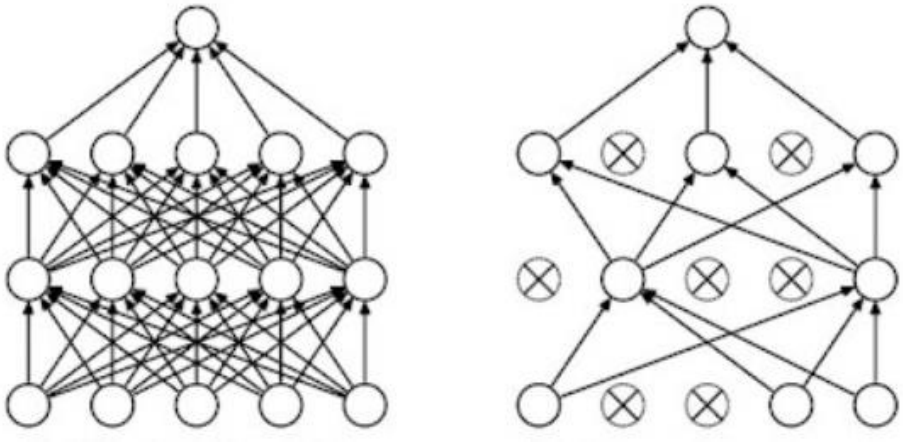
Tüm nöronlar bu katmanda bir dizi şeklinde görülür ve katmandaki tüm nöronlar kendisinden önceki katmanın alanlarına bağlıdır. Sınıflandırmadan önceki katmandır. Bu katmanda nesneye ait olan özelliklerin hangi sınıfla ilişkili olduğu belirlenir. Tam bağlantılı (full connected) katman Şekil 3.8'de gösterilmiştir.



Şekil 3.8. Tam bağlantılı (full connected) katman

3.1.6. Dropout

Konvolüsyonel sinir ağlarında, ağın ezberlemesini önlemeyi sağlayan katmandır. Bu katman kullanılarak ağın içerisindeki bazı düğümler, belirli bir nörona bağımlılığı önlemek için rastgele kaldırılır. Şekil 3.9'da ağın ilk hali ve dropput katmanı uygulandıktan sonraki hali gösterilmiştir.

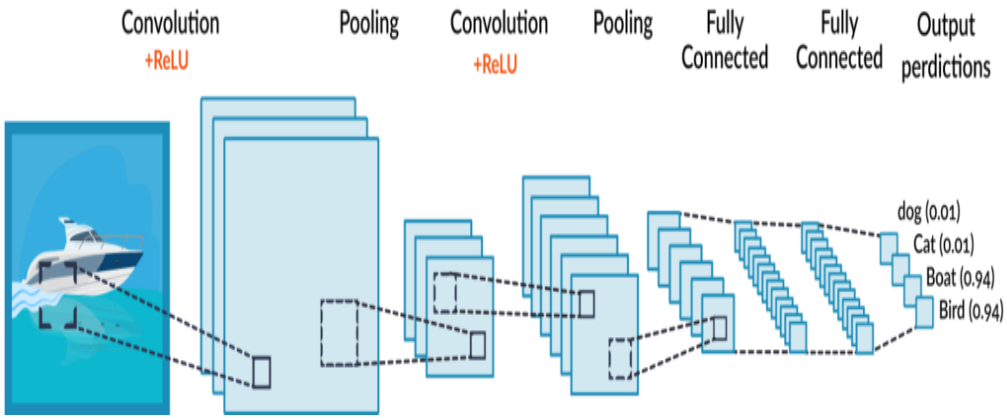


Şekil 3.9. Dropout katmanı [71].

3.1.7. Sınıflandırma

Tam bağlı katmandan sonra gelen ve son katman olan sınıflandırma katmanında sınıflandırma işlemi yapılır. Sınıflandırma yapılacak öğe kadar sonuç üretir ve her biri bir sınıfı temsil eder. Örnek olarak bir problem 4 sınıfa ayrılacaksa, 4 tane çıkış üretilir ve her bir sınıfa 0-1 arasında değer üretilir. 1'e en yakın olan çıkış, ağın tahmin etmiş

olduğu sınıftır. Softmax sınıflandırıcı yüksek başarı göstermesi sebebiyle genellikle kullanılmaktadır. Konvolüsyon katmanları Şekil 3.10'da gösterilmiştir.



Şekil 3.10. Konvolüsyon katmanları [72].

4. NESNE TANIMA (OBJECT DETECTION)

Sayısal görüntü işleme uygulamalarının vazgeçilmez unsurlarından olan nesne algılama ve nesne tanıma ile uzun yıllardır çalışılmaktadır. Nesne tanıma, bir görüntüdeki nesneleri belirleme sürecidir. Bu, görüntüdeki nesnenin türünü (araba, bisiklet, insan, köpek vb.) Tanımlamak için yapay zeka algoritmalarının kullanılması anlamına gelir. Nesne tanıma genellikle sınıflandırma olarak da adlandırılır. Örneğin, bir resimdeki araba, sınıflandırma yöntemleri kullanılarak tanınabilir.

Nesne algılama ise nesne tanımaya benzer, ancak bir görüntüdeki nesnelerin konumlarını da belirler. Yani nesne algılama, sınıflandırma ve nesne pozisyonunu bulma işlemidir. Bu, görüntüdeki nesnenin sınıfını belirlemekle birlikte, nesnenin hangi bölgede olduğunu belirler ve sınırlayıcı kutu (bounding box) olarak adlandırılan bir çerçeve kullanarak nesneyi işaretler. Örneğin, bir resimdeki araba, nesne algılama algoritmaları kullanılarak bulunabilir ve arabayı sınırlayan bir kutuyla işaretlenebilir.

Kısacası, nesne tanıma sınıflandırmaya odaklanırken, nesne algılama sınıflandırma ve nesne pozisyonunu belirleme işlemlerine odaklanır. Bu süreçte, bir bilgisayar programı veya yapay zeka algoritması, görüntüdeki nesneleri tanımlamak için farklı özellikleri analiz eder ve belirli bir sınıfa (örneğin insan, araba, evcil hayvan, mobilya vb.) atar. Nesne tanıma teknolojisi, otonom araçlar, güvenlik sistemleri, sanal gerçeklik, robotik ve daha pek çok alanda kullanılmaktadır.

Bu algoritmalar farklı mimari, özellik çıkarma ve sınıflandırma yöntemleri kullanarak nesneleri tanımlamak için tasarlanmıştır. Bu algoritmaların performansı, özellikle doğruluk ve hız açısından, birbirinden farklıdır ve uygulamanın gereksinimlerine göre seçilmelidir.

Nesne tespiti algoritmaları, bir görüntü veya videoda nesnelerin varlığını ve konumunu tespit eden bilgisayar görüsü algoritmalarıdır. Bu algoritmalar genellikle, nesnelerin bulunduğu görüntü bölgelerini belirlemek, bu nesneleri önceden tanımlanmış kategorilere sınıflandırmak ve bunları sınırlayıcı kutular veya maskelerle lokalize etmek gibi işlemleri içerir. Nesne tespiti için kullanılan bazı algoritmalar arasında

YOLO (You Only Look Once), Faster R-CNN (Faster Region-based Convolutional Neural Network), Mask R-CNN (Mask Region-based Convolutional Neural Network) ve SSD (Single Shot MultiBox Detector) bulunmaktadır.

Sınıflandırma çalışmalarında eğitim seti hazırlanırken, sınıflandırılacak veriler ayrı bir şekilde eğitilirken, nesne tanıma problemlerinde ise tüm nesneler aynı resim üzerinde bulunabilir. Örneğin, sınıflandırma çalışmasında hem kedi hem köpek aynı anda sisteme verilmez, nesne tanımada ise ikisi de aynı anda verilebilir.

Nesne tanıma işlemi ile hem resimde kedi veya köpek var mı, hem de kedi ve köpeğin bulunduğu bölgeler belirlenir. Son yıllarda, nesne tanıma alanında hızlı gelişmeler kaydedilmiştir. Nesne tanıma işlemi, belirli kategoriler temel alınarak, görüntü üzerinde nesne araması yapılmasıyla gerçekleştirilir. Örneğin, kedi ve köpek gibi iki sınıfımız varsa, bu iki sınıf üzerinde model eğitilerek, görüntüde kedi veya köpek aranır. Şekil 4.1'de gösterildiği gibi, hem kedi hem de köpek var mı diye bakılır ve var olan nesnelerin görüntü üzerindeki konumu belirlenir. Bir görüntüde birden fazla kedi veya köpek olabilir ve bu durumda tüm kedi ve köpeklerin tespiti yapılır. Nesne tanıma, normal sınıflandırmadan farklı olarak, görüntü üzerinde birden fazla nesneyi tanıyor olmasıyla öne çıkar. Normal sınıflandırmada, görüntüde tek bir nesne odak noktasıdır. Ancak kedi-köpek ayrımı yapan bir sinir ağı oluşturulduğunda, köpek ve kedi resimleri sisteme ayrı ayrı verilir ve her ikisinin olduğu resimler sisteme verilmez. Nesne tanıma işleminde ise görüntüdeki tüm nesnelerin tespiti yapılmalıdır.

Nesne algılama ve tanıma için çeşitli teknikler ve yöntemler geliştirilmiştir. Viola-Jones, dijital görüntülerdeki nesneleri etkili bir şekilde tespit eden ilk algoritmadır. Son yıllarda grafik işleme birimlerindeki ve derin öğrenmedeki gelişmeler sayesinde nesneleri daha başarılı bir şekilde tespit edip tanımlayabilen yöntemler geliştirilmiştir.

Nesne algılama çalışmalarında önce R-CNN, ardından Fast R-CNN yapısı ortaya çıkmıştır. Günümüzde yaygın olarak kullanılan versiyon olan Faster R-CNN olarak 2015 yılında ortaya çıkmıştır. R-CNN ailesinde, sürümler arasındaki farklılıklar genellikle hesaplama verimliliği, azaltılmış test süresi ve performans iyileştirmesi (mAP) ile ilgilidir.

Şekil 4.1. Nesne tanıma için bir örnek

Nesne tanıma ağları tipik olarak aşağıdaki bileşenlerden oluşur:

1. Görüntüdeki "sınırlayıcı kutular" veya olası nesnelerin konumlarını oluşturmak için bir bölge önerisi algoritması,

2. Nesnelerin özellikleri bulunması için kullanılan CNN,

3. Bu nesnenin hangi sınıfa ait olduğunu tahmin etmek için bir sınıflandırma katmanı,

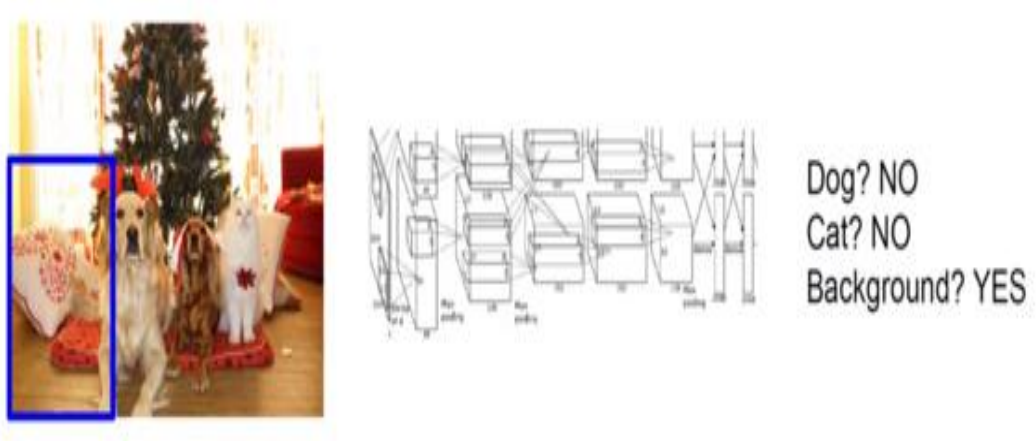
4. Bir regresyon katmanı, sınırlayıcı kutunun koordinatlarında daha kesin olmalıdır.

CPU tabanlı bölge öneri teknikleri, hem R-CNN'de hem de Fast R-CNN'de (Seçici Arama algoritması) kullanılmaktadır. Faster R-CNN, bölge öneri süresini görüntü başına, 2 saniyeden 10 milisaniyeye düşüren başka bir evrişimli ağ kullanarak bölge önerileri üretir. Faster R-CNN mimarisi, bölge öneri algoritması olarak RPN (Region Proposal Network) ve dedektör ağı olarak Fast R-CNN'den oluşmaktadır. Faster R-CNN'de, ilk CNN uygulanır ve bir özellik haritası oluşturulur.

4.1. Pencere Kaydırma Yöntemi

Pencere kaydırma yöntemi, bir veri setindeki büyük bir görüntü veya videoyu küçük parçalara bölerek, her bir parça üzerinde nesne tespiti veya sınıflandırma işlemlerini gerçekleştirmek için kullanılan bir tekniktir. Bu yöntemde, bir kaydırma penceresi, bir görüntünün veya videonun bölünmüş parçaları üzerinde sırayla gezdirilir ve her bir parça için nesne tespiti veya sınıflandırma işlemi yapılır. Bu sayede, büyük boyutlu görüntüler veya videolar için nesne tespiti veya sınıflandırma işlemlerini uygulamak

mümkün hale gelir. Ancak bu yöntem, gereksiz hesaplama yüküne neden olabilir ve yanlış pozitif sonuçlara neden olabilecek nesne kesişimleri gibi problemlerle de karşılaşılabilir. Pencere kaydırma yöntemi Şekil 4.2'de gösterilmiştir.



Şekil 4.2. Pencere kaydırma yöntemi [73].

Pencere kaydırma yönteminde çok fazla pencere oluşturulur ve oluşturulan tüm pencereler tek tek konvolusyon sinir ağından geçirilir. Pencere kaydırma methodu ile çok fazla, belki milyon tane pencere oluşturmak gerekecek ve bu da zaman kaybına sebep olacaktır. Resmin her yerinde farklı boyutlarda pencereler oluşturulup ve hesaplamalar yapmak, bilgisayar için çok ağır bir işlem oluşturmaktadır. Tüm resmi pencerelerle gezmek yerine, resim içerisinde nesne olabilecek 1000/2000 alan belirlemek ve akabinde bölge önerisi yapılan 1000/2000 alan içerisinde nesne aramak çok daha kolay olacaktır. Bu nedenle daha hızlı sonuç veren bölge önerisi yöntemi daha popüler hale gelmiştir.

4.2. Bölge Önerisi

Bölge önerisi, bir görüntüde potansiyel nesne konumlarını belirlemek için kullanılan bir tekniktir. Bu yöntem, bir görüntünün farklı bölgelerinde potansiyel nesne konumlarını öneren bir algoritma kullanarak çalışır. Bölge önerisi aşamasından sonra, önerilen nesne konumları daha ayrıntılı olarak incelenebilir ve nesne tespiti veya sınıflandırma algoritmalarına beslenebilir.

Bölge önerisi teknikleri, nesne tespiti sistemlerinin verimliliğini artırmak ve aşırı işlem yükünü azaltmak için yaygın olarak kullanılır. Özellikle, önerilen nesne konumlarının sayısını azaltmak, nesne tespiti ve sınıflandırma işlemlerinin daha hızlı ve verimli bir şekilde gerçekleştirilmesine olanak sağlar. Örneğin, popüler bir bölge önerisi yöntemi

olan Selective Search, nesne tespiti ve sınıflandırma işlemlerinde kullanılmak üzere yüzlerce potansiyel nesne konumunu önerir. Selective Search Algoritması ile nesne bulunma ihtimali olan alanlar için bölge önerisi yapılıyor ve öneri işlemi yapılırken bu bölgelerde, resimdeki renklerin tonların farklıklıklarına göre işaretleniyor ve daha sonra işaretlenen/belirlenen bölgeler konvolüsyondan geçiriliyor. CPU ile birkaç saniyede 2000 bölge önerisi yapılıyor.Bölge önerisi Şekil 4.3 ‘de gösterilmiştir.

Şekil 4.3.Bölge önerisi

4.3. Nesne Tanımada Kullanılan Algoritmalar

4.3.1. R-CNN (Region Based Convolutional Neural Network)

R-CNN (Region-based Convolutional Neural Network), nesne tespiti alanında oldukça popüler bir algoritmadır. Bu algoritma, ilk olarak 2014 yılında yayınlanmıştır ve o zamandan beri birçok araştırmacı tarafından kullanılmış ve geliştirilmiştir. R-CNN, bölge önerisi ve evrişimli sinir ağı (CNN) kullanarak nesne tespiti yapar. R-CNN, nesne tespiti için oldukça iyi sonuçlar verir ve birçok uygulama alanında kullanılabilir. Ancak, R-CNN oldukça yavaş bir algoritma olarak bilinir ve daha hızlı ve verimli alternatifler geliştirilmiştir.

Daha öncede belirtildiği üzere tüm görüntüyü pencerelerle gezmek yerine bölge önerisiyle belirlenen yaklaşık 2000 pencerede nesne aramak çok daha kolay olacaktır. R-CNN [74]’de bu işlevi görmektedir.

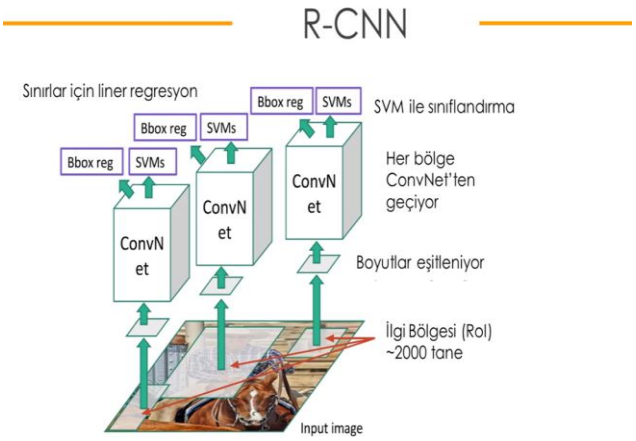
Resim üzerinde öncelikle ROI (Region of Interest) bölge önerisi yapılır. Daha sonra her bölge için CNN (ConvNet) uygulanır ve oluşturulan özellik haritaları (Feautre Map) için sınıflandırma algoritması olan SVM (Destek vektör makineleri) kullanılır ve nesnenin var olup olmadığı kontrol edilip sınıflandırılır. Ve linear regrasyon

nesnenin sınırları belirlenir. Bu yöntemin en büyük sıkıntısı zamandır. Her bölge ayrı ayrı CNN'den geçirildiği için eğitim yaklaşık 84 saat sürer.

R-CNN mimarisinde, önerilen bölgelerin boyutları farklı olabilir. Ancak, bu pencereleri CNN ile işlemek için boyutları eşitlenmelidir. Sınıflandırma yapan CNN'ler, belirli boyutlarda giriş alırlar. Bu nedenle, boyutları eşitlenen pencereler tek tek CNN'e gönderilir [75]. Sonuçta, gözetimli öğrenmede kullanılan bir makine öğrenmesi yöntemi olan SVM kullanarak sınıflandırma yapılır ve nesnenin sınırları lineer regresyon kullanarak belirlenir. Bu sayede, tanımlanan nesnelerin etrafına bir dörtgen çizilebilir. Lineer regresyon, bir veya birden fazla bağımsız değişken ile başka bir bağımlı değişken arasındaki bağlantıyı modellemek için kullanılan bir yöntemdir [76]. R-CNN, bir bölgede nesne arar ve nesneyi bulursa sınıfını döndürür. Aynı zamanda, nesnenin görüntüdeki konumunu belirleyen dört değer verir. Bu değerler, çizilecek dörtgenin koordinatlarını belirtir [74]. R-CNN mimarisinin yapısı Şekil 4.4'te gösterilmektedir.

R-CNN gibi bölge temelli nesne tespit algoritmaları öncelikle nesne tespiti için, nesnenin bulunması muhtemel alanları belirler ve ardından bu bölgelerde ayrı ayrı CNN işleminden geçirikmesi sebebiyle RCNN algoritmaları yavaş çalışmaktadır.

R-CNN in karşılaştığı problemlere bakacak olursan neden Fast R-CNN [71], [77] ve Faster R-CNN [75], [78]'in geliştirildiği daha net anlaşılacaktır. Öncelikle önerilen bölgelerin teker teker CNN'den geçmesi R-CNN'i oldukça yavaş çalışan bir ağ haline dönüştürmektedir. Bu yavaş çalışmayı gidermek için Fast R-CNN geliştirilmiştir [71], [77]. Fast R-CNN, R-CNN ile benzer yapıdadır. Sadece hızlandırmak için farklı teknikler kullanılmaktadır.



Şekil 4.4. R-CNN mimari yapısı [79].

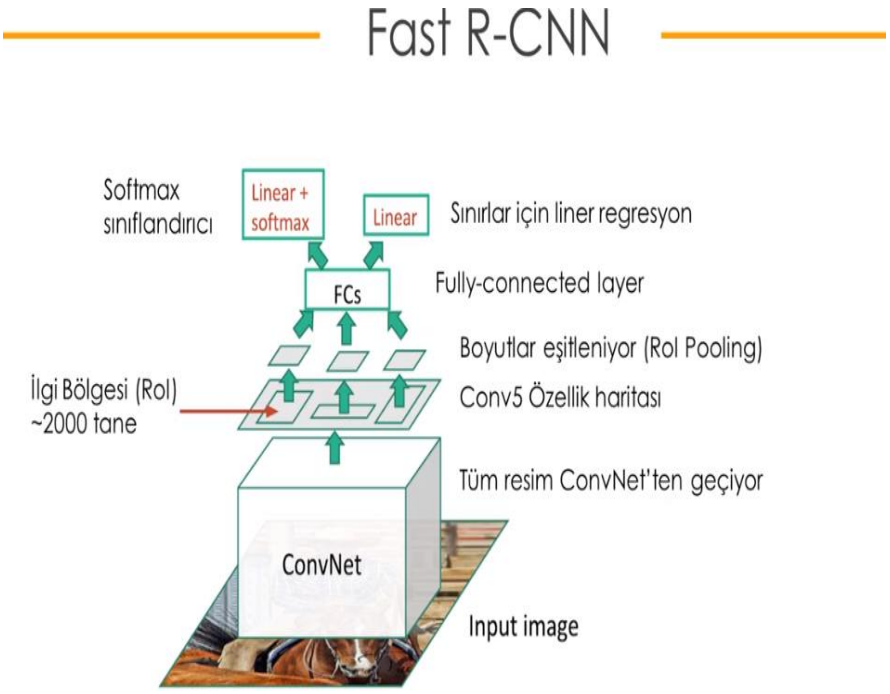
4.3.2. Fast R-CNN

Fast R-CNN, önceki bir nesne algılama modeli olan R-CNN'ye (Region-based Convolutional Neural Network) dayanan bir nesne algılama algoritmasıdır. Fast R-CNN, R-CNN'nin yavaş çalışmasına yönelik bir çözüm sunar ve aynı zamanda daha yüksek algılama doğruluğu sağlar.

Fast R-CNN, nesne tespiti işlemi için önce görüntüdeki bölge önerilerini hesaplar. Daha sonra, her bölge önerisi, bir özellik haritasında yeniden boyutlandırılır ve bir özellik vektörü olarak temsil edilir. Bu özellik vektörleri, bir tamamen bağlı katmandaki nöronlar tarafından işlenir ve nesne sınıflandırması ve sınırlayıcı kutuların düzenlenmesi gibi işlemler gerçekleştirilir.

Fast R-CNN, R-CNN'ye göre daha hızlıdır, çünkü aynı görüntüdeki bölge önerilerinin özelliklerini hesaplamak için sadece bir kez evrişim yapar. Ayrıca, Fast R-CNN, çoklu sınıf nesne algılama yeteneğiyle birlikte sınırlayıcı kutuların regresyonunu da gerçekleştirebilir.

Fast R-CNN, nesne tespiti ve nesne sınıflandırma alanında oldukça popüler bir algoritmadır ve birçok uygulama alanında kullanılmaktadır. Şekil 4.5'de Fast R-CNN yapısı görülmektedir.

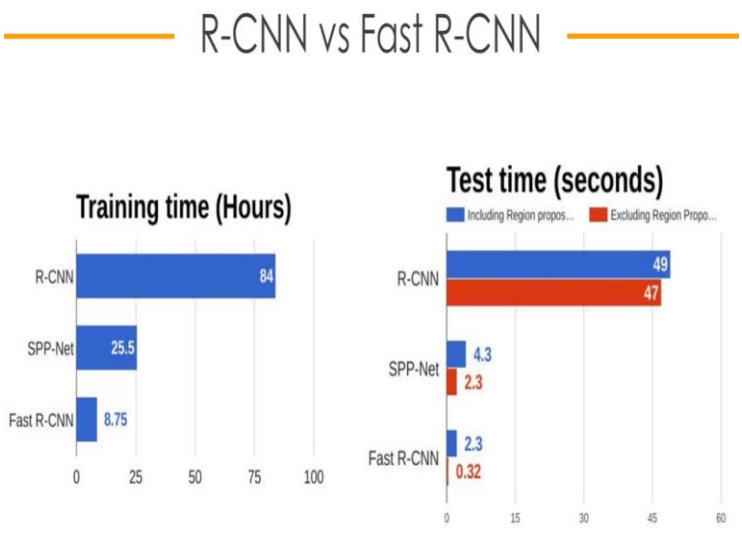


Şekil 4.5. Fast R-CNN [80].

Bu yöntemin R-CNN'den farkı önce resmi bölgelere bölmek yerine yani orijinal resim üzerinde 2000 tane bölge önerisi (RoI) oluşturmak yerine önce convolüsyon katmanından geçirilerek özellik haritası oluşturulur ve bu sayede her bölge için ayrı ayrı uygulamaya gerek kalmaz ve sadece sonrasında oluşan harita üzerinde bölge önerileri (selective search algoritması ile) yapılır. R-CNN'den farklı, direk bölge önerisi yapmak yerine ilk olarak, konvolüsyon katmanından geçirilir ve özellik haritasından sonra bölge önerisi (selective search algoritması kullanarak) yapılmaktadır.

Ayrıca sınıflandırma metodu olarak SVM (Support Vector Machine) yerine yapay sinir ağları katmanları içerisinde gerçekleşen softmax classification kullanılır.

Fast R-CNN, eğitim aşamasında oldukça hızlı çalışırken test aşamasında bölge önerisi yapmak için harcanan zamanın çoğunu tüketir. Eğer bölge önerisi için harcanan zaman kısaltılırsa modelin hızı daha da artar. Faster R-CNN, R-CNN'den farklı olarak bölge önerisini ağın içinde yaparak bu sorunu tam olarak çözer. Bölge belirledikten sonra boyutları uygun hale getirilir. R-CNN sınıflandırıcı SVM ile bir model oluşturmak yerine Fast R-CNN'de sinir ağını genişletip, sinir ağı içinde sınıflandırma yapar. Lineer regresyon ile ise, nesnenin sınırları belirlenir. R-CNN'den farklı, direk bölge önerisi yapmak yerine ilk olarak, konvolüsyon katmanından geçirilir. Özellik haritasından sonra bölge önerisi (selective search algoritması kullanarak) yapılır.



Şekil 4.6. R-CNN ile Fast R-CNN arasındaki fark [81].

Her ne kadar Fast R-CNN, R-CNN'e göre çok daha hızlı sonuç verse de, Şekil 4.6'dan da anlaşılacağı üzere, [81]'deki çalışmada bölge önerisi için ayrılan süre yaklaşık 2 saniyedir ki bu da aslında diğer işlemlere kıyasla bölge önerisinin çok daha fazla zaman aldığını göstermektedir. Bu zaman kaybının önüne geçmek için, Faster R-CNN yapısı oluşturulmuştur.

R-CNN (Regions with Convolutional Neural Networks) ve Fast R-CNN (Fast Regions with Convolutional Neural Networks) arasındaki temel fark, nesne bölgesi önerilerinin oluşturulması ve kullanılmasıdır.

R-CNN, öncelikle seçilen bölge önerilerinin her biri için ayrı ayrı CNN'ler işlemleyerek özellik çıkarır. Bu, özelliklerin her bir bölge önerisi için ayrı ayrı hesaplandığı anlamına gelir. Bu yöntem doğru sonuçlar vermesine rağmen oldukça yavaştır.

Fast R-CNN, her bir bölge önerisi için özelliklerin ayrı ayrı hesaplanmasını ortadan kaldırır ve tüm görüntü için tek bir özellik haritası oluşturur. Böylece, aynı bölge önerilerinin birden fazla kez hesaplanmasını önleyerek R-CNN'den daha hızlıdır. Ayrıca Fast R-CNN, maliyet fonksiyonunu optimize etmek için bütünleşik bir bölge önerisi ağı kullanır.

Başka bir fark, R-CNN'de kullanılan seçimli araştırma yöntemi ile Fast R-CNN'nin RoI (Region of Interest - İlgili Bölge) havuzu kullanmasıdır. RoI havuzu, herhangi bir boyuttaki bölge önerilerinin tek boyutlu bir tensor'e dönüştürülmesini sağlar. Bu, farklı boyutlardaki bölge önerilerini tek bir boyutta işleyebilmeyi mümkün kılar. Seçimli araştırma yöntemi ise daha yüksek hesaplama maliyetine sahiptir ve daha fazla kaynak tüketir.

Sonuç olarak, Fast R-CNN, R-CNN'den daha hızlıdır ve tek bir özellik haritası kullanarak daha tutarlı sonuçlar üretir. Ancak, RoI havuzu yerine seçimli araştırma kullanımı gibi bazı yönlerden daha maliyetlidir.

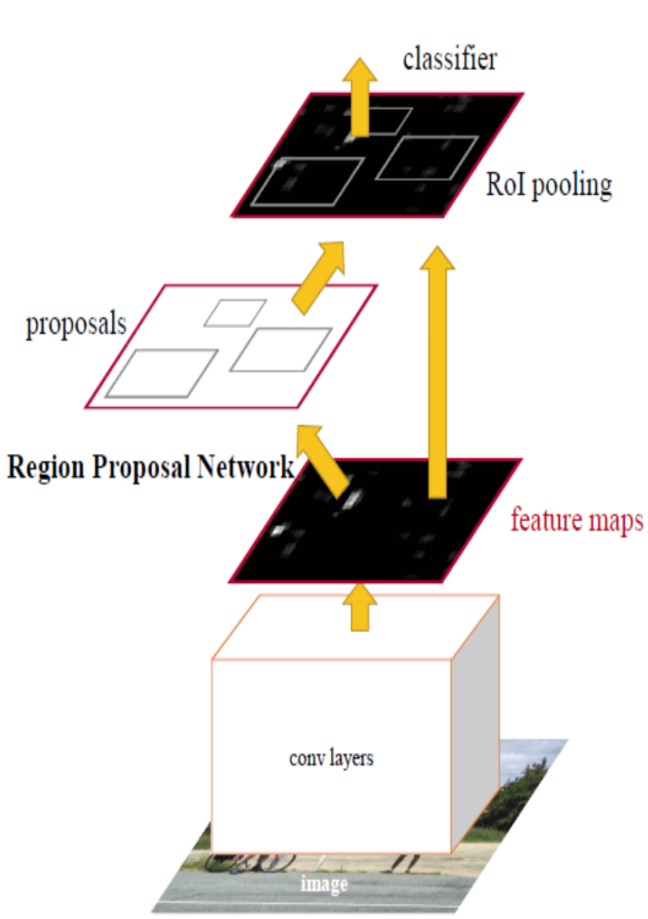
4.3.3. Faster R-CNN

Faster R-CNN ilk olarak 2015 yılında Ross Girshick tarafından yayınlanmıştır. Faster R-CNN, daha hızlı bölge önerisi yapabilen bir nesne algılama algoritmasıdır. Fast R-CNN algoritmasının iyileştirilmiş bir sürümüdür. Faster R-CNN, önceden eğitilmiş bir özellik çıkarıcı CNN kullanarak görüntüyü işler ve ardından bölge önerisi RPN

kullanarak ilgili alanları seçer. Bu sayede bölge önerisi aşamasında ayrı bir algoritma kullanmak yerine, RPN doğrudan CNN'e entegre edilerek, bütün nesne algılama süreci tek bir modele dönüştürülebilir. Bu da daha hızlı ve daha verimli bir nesne algılama sağlar.

Faster R-CNN, video gözetimi, otonom sürüş, güvenlik kamerası izleme, sanayi otomasyonu, tıbbi görüntüleme ve hava fotoğrafçılığı gibi birçok alanda kullanılabilir.

Bu yöntemde de ilk CNN uygulanıp özellik haritası oluşturulur. Bölge önerileri kısmında seçici bölge araması yerine ayrı bir bölge önerisi ağı oluşturarak bölgeler seçilir. Geri kalan kısımlar Fast R-CNN ile neredeyse aynıdır. Bu teknikle tahmin süresin 0.3 saniyeye kadar düşürülür.

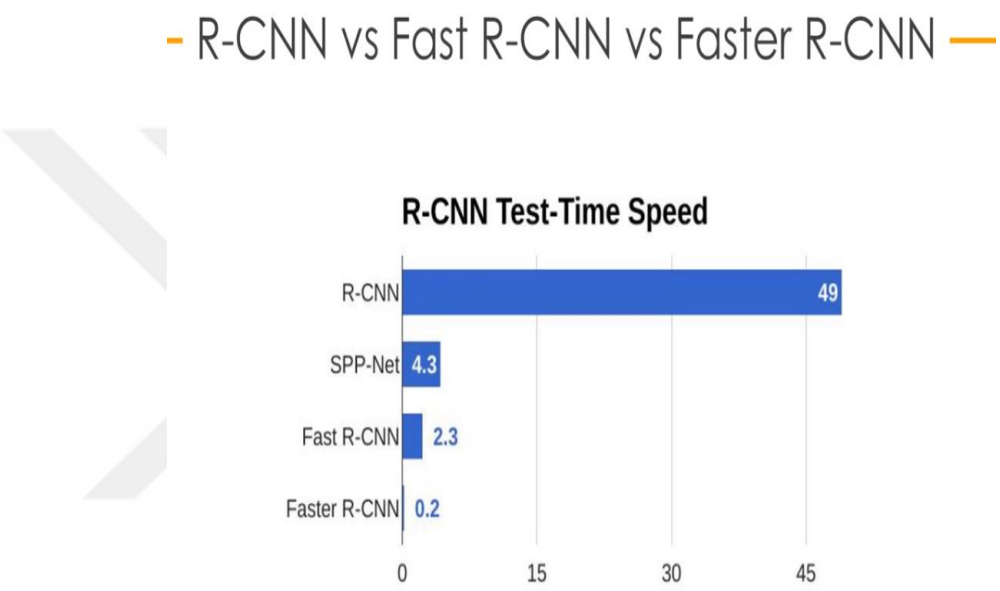


Şekil 4.7. Faster R-CNN mimari yapısı [82].

Faster R-CNN modeli önerilen bölgeleri oluştururken "Region Proposal Network" isminde bir algoritma kullanır. Bu algoritma bir "sliding window" oluşturur ve konvolüsyon katmanında oluşturulan özellik haritası üzerinde gezer. Her bölgede bir nesnenin olduğunu varsayar ve bölgelere tarafsız puanlar atar. Bunun için bitişik

piksellere, renk, yoğunluk vs. kriterlere bakar. Daha sonra “ReLU” aktivasyon fonksiyonu kullanılarak bir özellik haritası çıkartılır. Geri kalan adımlar Fast R-CNN ile benzerdir. Faster R-CNN mimari yapısı Şekil 4.7’de gösterilmiştir.

Selective Search ile bölge önerisi almak yerine, bu önerileri network içerisinde yaparak hız kazanılır. Fast R-CNN'den farkı, Selective Search ile bölge önerisi yapmak yerine, ayrı bir bölge önerisi ağı oluşturulur. Artık bölge önerileri bu ağ içerisinde yapılır. Devamı ise Fast R-CNN ile aynıdır, fully connected ve sınıflandırıcı kullanılmaktadır.



Şekil 4.8. R-CNN ile Fast R-CNN, Faster R-CNN hız karşılaştırılması [81].

Faster R-CNN, Fast R-CNN algoritmasının bir geliştirilmiş sürümüdür. İşlevsel olarak, Faster R-CNN, Fast R-CNN'in ana bileşenleri olan bölge önerileri ve özellik haritaları üretme gibi adımları daha verimli hale getirmek için :

1. Bölge önerilerinin üretilmesi: Fast R-CNN'de bölge önerileri önceden belirlenmiş özellik haritalarından çıkarılan özellik vektörleri kullanılarak üretilirken, Faster R-CNN, özellik haritalarını doğrudan bir RPN (Region Proposal Network) kullanarak üretir.

2. Daha hızlı işlem: Faster R-CNN, Fast R-CNN'e kıyasla daha hızlıdır. Bölge önerileri üretmek ve nesneleri tespit etmek için daha az hesaplama gerektirir.

3. Nesne tespiti doğruluğu: Faster R-CNN, Fast R-CNN'e kıyasla daha yüksek nesne tespiti doğruluğu sağlar.

4. Eğitim: Faster R-CNN, Fast R-CNN'e kıyasla daha kolay eğitilir.

Bu farklar göz önüne alındığında, Faster R-CNN'in Fast R-CNN'den daha iyi performans gösteren bir algoritma olduğu söylenebilir. R-CNN ile Fast R-CNN, Faster R-CNN hız karşılaştırılması Şekil 4.8'de görülmektedir.

4.3.4. YOLO algoritması

YOLO, konvolüsyonel sinir ağlarını kullanarak gerçek zamanlı nesne tespiti yapan bir algoritmadır. Nesne tespitini çok hızlı bir şekilde tek seferde yapabiliyor olmasından dolayı YOLO 'You Only Look Once' adını almıştır. YOLO algoritmasının diğer algoritmalardan farkı, çalışmaya başladığı anla birlikte hem görüntülerdeki nesneleri hem de bu nesnelere ait koordinatları aynı anda tespit eder. YOLO algoritması doğru sonuçları hızlı şekilde vermesinden dolayı son yıllarda oldukça önem kazanmıştır ve çok farklı alanlarda kullanılmaktadır.

Nesneleri gerçek zamanlı olarak algılamak için CNN (konvolüsyonel sinir ağlarını) kullanır. YOLO algoritmasının adından da anlaşılacağı üzere sinir ağı üzerinde tek bir ileri yayılım yaparak, algoritma nesneleri algılar. CNN, çeşitli sınıf olasılıklarını ve sınırlayıcı kutuları aynı anda tahmin etmek için kullanılır ve bu özelliklerinden dolayı diğer nesne tanıma algoritmalarına göre gerçek zamanlı nesne tespitinde çok daha üstün başarı göstermektedir.

YOLO ve Faster R-CNN gibi algoritmalar, nesne tanıma alanında kullanılan iki popüler yöntemdir. YOLO, tek bir derin sinir ağı kullanarak nesne tespiti ve sınıflandırma yaparken, Faster R-CNN, bölge önerisi (region proposal) ve bölge tabanlı öznitelik çıkarımı (region-based feature extraction) gibi aşamaları içeren bir iki aşamalı bir yaklaşım kullanır.

YOLO, daha hızlı ve daha verimli bir şekilde çalışırken, Faster R-CNN daha yüksek doğruluk elde edebilir. Ancak, Faster R-CNN daha yavaş çalışır ve daha fazla hesaplama gücü gerektirir. Ayrıca, YOLO, eşzamanlı olarak birden çok nesne tespit edebilirken, Faster R-CNN tek bir nesneyi tespit etmek için daha uygundur.

YOLO algoritması birkaç bileşeni barındırır:

1. Izgara Blokları (Residual blocks)
2. Sınırlayıcı kutu regresyonu (Bounding box regression)
3. Intersection Over Union (IoU)

4.3.4.1. Izgara Blokları (Residual blocks)

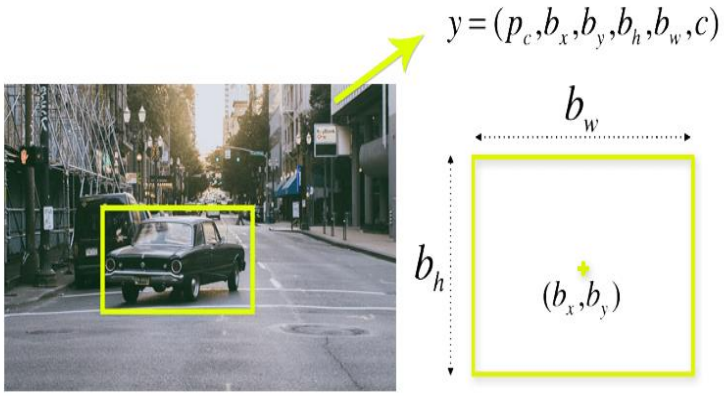
Şekil 4.9'da görüldüğü üzere öncelikle görüntü çeşitli ızgaralara bölünür. Her ızgara S x S boyutuna sahiptir.

Şekil 4.9. Izgara blokları [83].

Şekil 4.9'da, ızgara hücreleri eşit boyuttadır ve her bir ızgara hücresi, içinde görünen nesneleri algılayacaktır. Örneğin, belirli bir ızgara hücresi içinde bir nesne merkezi varsa, bu hücre o nesneyi algılamaktan sorumlu olacaktır.

4.3.4.2. Sınırlayıcı kutu regresyonu (Bounding box regression)

Sınırlayıcı kutu (Bounding box), görüntüdeki bir nesneyi vurgulayan bir çerçevedir. Bir görüntüdeki her bir sınıflayıcı kutu; genişlik (b_w), yükseklik (b_h), sınıf, sınırlayıcı kutu merkezi (b_x, b_y) özniteliklerine sahiptir. Sınırlayıcı kutu örneği Şekil 4.10'da gösterilmektedir. Sınırlayıcı kutu görselde sarı bir çerçeve ile gösterilmiştir.



Şekil 4.10. Sınırlayıcı kutu regresyonu (Bounding box regression) [83].

YOLO algoritması, nesnelerin yüksekliğini, genişliğini, merkezini ve sınıfını tahmin etmek için tek bir sınırlayıcı kutu regresyonu kullanır.

YOLO algoritması çeşitli alanlarda kullanılır bunlardan bazıları:

1. Otonom araçlar : Otomobillerin etrafındaki bir çok nesneyi(araçlar, insanlar ve park sinyalleri) algılamak için otonom otomobillerde kullanılır. Nesne tespiti sayesinde, çarpışmaların önüne de geçilmektedir.
2. Güvenlik: YOLO güvenlik sistemlerinde de kullanılmaktadır, örneğin güvenlik sebebiyle insanların bazı bölgelerden geçişlerinin kısıtlanması durumunda, ihlal yapan alanlardan geçenler gerçek zamanlı nesne tespiti yapan YOLO algoritması tarafından tespit edilir.
3. Tıbbi Görüntüleme: YOLO algoritması, tıbbi görüntülerdeki tümör, lezyon veya diğer patolojik bölgeleri algılamak için kullanılabilir. Bu sayede, erken teşhis ve tedavi mümkün hale gelebilir.
4. Tarım: YOLO algoritması, tarım sektöründe bitkilerin büyüme aşamalarını, hastalık ve zararlıları tespit etmek için kullanılabilir. Bu sayede, çiftçiler verimliliği artırabilir ve tarımsal kayıpları minimize edebilirler.

4.3.4.3. Intersection over union (IOU)

IoU, "Intersection over Union" teriminin kısaltmasıdır ve genellikle nesne tespiti gibi görevlerde kullanılan bir değerlendirme metriğidir. IoU, nesne algılama ve nesne tanıma gibi görsel işleme uygulamalarında sıkça kullanılan bir değerlendirme metriğidir. IoU, tahmin edilen bir nesnenin gerçek bir nesneyle ne kadar örtüştüğünü ölçer.

IoU değeri, tahmin edilen bir nesnenin gerçek bir nesneyle kesişim alanının gerçek nesne alanına bölünmesiyle hesaplanır. Bu değer, 0 ile 1 arasında bir sayıdır ve 1'e ne kadar yakınsa, tahmin edilen nesne gerçek nesneye o kadar çok benzer.

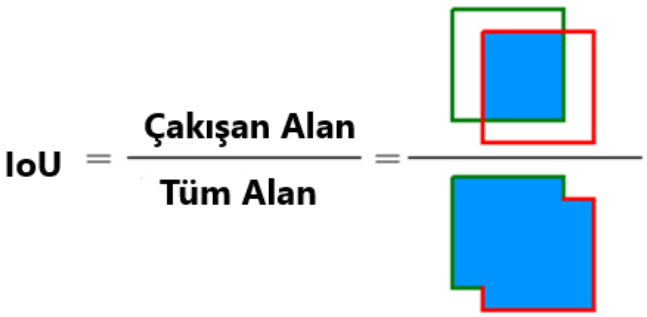
IoU, özellikle nesne algılama ve nesne tanıma modellerinin eğitiminde ve performansının ölçülmesinde sıkça kullanılır. Örneğin, bir nesne algılama modelinin doğruluğunu ölçmek için, modelin tahmin ettiği nesnelerle gerçek nesneler arasındaki IoU değerlerinin hesaplanması ve ortalama alınması gerekebilir. Bu sayede, modelin ne kadar doğru sonuçlar ürettiği hakkında bilgi edinilebilir.

IoU, bir tahminin doğruluğunu değerlendirmek için gerçek etiketlenmiş bir nesneyle örtüşen piksel sayısını gerçek ve tahmin edilen alanların birleşimine bölen bir oran olarak hesaplanır.

Örneğin, bir nesne tespiti modeli bir nesneyi bir dikdörtgen olarak tahmin ederken, gerçek nesne etiketinin de bir dikdörtgen olduğunu varsayalım. IoU, bu iki dikdörtgenin kesişim alanının birleşim alanına oranı olarak hesaplanır. IoU değeri, 0 ile 1 arasında bir değer alır ve 1'e ne kadar yakınsa, tahminin gerçek etikete o kadar yakın olduğunu gösterir. IoU, özellikle nesne tespiti ve segmentasyonunda sıklıkla kullanılan bir metriktir ve model performansının ölçülmesinde ve geliştirilmesinde yardımcı olur.

Jaccard indeksi olarak da adlandırılan IoU, gerçek değer ile (gt) ile tahmin edilen değer (pd) arasındaki örtüşmeyi değerlendiren bir ölçümdür. Intersection over Union (IoU) basitçe bir değerlendirme metriğidir. Daha açık olarak, bir nesne dedektörünü değerlendirmek için Intersection over Union uygulamak için ihtiyacımız olan:

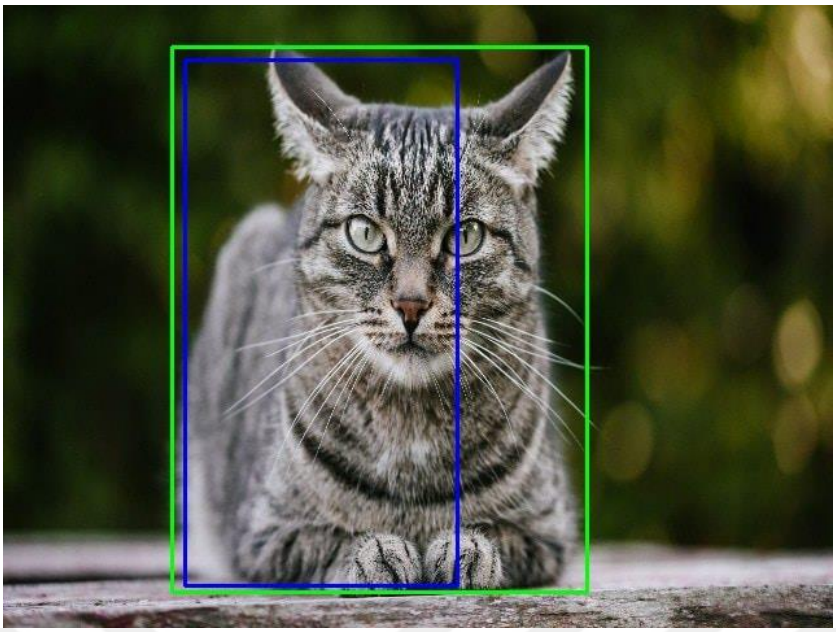
1. The ground-truth(kesin referans) bounding boxes (yani, nesnemizin görüntüde nerede olduğunu belirten test kümesinden elle etiketlenmiş sınırlayıcı kutular).
2. Modelimizden tahmin edilen sınırlayıcı kutular (bounding boxes).



Şekil 4.11. IoU formülü [85].

Nesne algılama problemlerinde, birleşim üzerinden kesişme (IoU), önemli bir kavramdır. Nesneleri başarılı bir şekilde berlirleyen bir kutu oluşturmak için YOLO algoritması IoU'yu kullanır. Tahmin edilen çerçeve, gerçek çerçeve ile aynıysa IoU değeri 1'e eşittir.

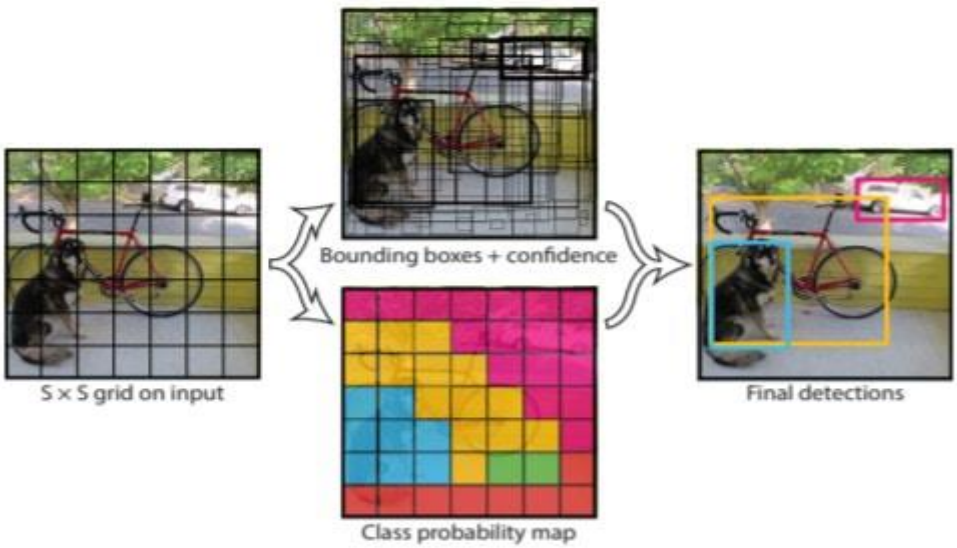
IoU metriği, 0 ile 1 arasında olup, 0 örtüşme olmadığını ve 1, gt ve pd arasında mükemmel bir örtüşme anlamına gelir.



Şekil 4.12. IoU'nun uygulanma örneği gösterilmiştir [83].

Şekil 4.11'de görüleceği üzere, 2 farklı renk tonunda çerçeve vardır. Mavi ile çerçeveye alınan kutu tahmin edilen kutu iken, yeşil ile belirlenen kutu ise gerçek olması gereken alandır.

Şekil 4.12, YOLO algoritmesının sonuçlarını oluşturmak için üç yapının nasıl uygulandığını göstermektedir.



Şekil 4.13.YOLO ile nesne tanıma işlemleri [83].

Özetle, başlangıç aşaması olarak bir görüntü ızgara hücrelerine bölünür ve her hücre sınırlayıcı kutuları tahmin eder, güven skorlarını oluşturur ve sınıf olasılıklarını tahmin

eder. Şekil 4.12'de en az 3 nesne sınıfına (bisiklet, araba, köpek) ait bir örnek verilmiştir. Tüm tahminler tek bir konvolüsyonel sinir ağı kullanarak aynı anda yapılır. YOLO (You Only Look Once) ilk kez Joseph Redmon tarafından 2016 yılında yayınlanmıştır. İlk sürümü YOLOv1 olarak adlandırılmıştır. YOLOv1, tek bir derin sinir ağı kullanarak görüntülerde nesne algılama yapabilen ilk algoritmalardan biriydi. Diğer nesne algılama algoritmalarından farklı olarak, YOLOv1, nesnelerin bulunduğu bölgeyi ayrıntılı bir şekilde bölümlere ayırmak yerine, tüm bölgenin tek bir kez işlenmesini sağladı. Bu sayede, YOLOv1 daha hızlı ve daha az işlemci gücü tüketerek çalışabilen bir algoritma haline geldi. YOLOv1, sınırlı sayıda nesne sınıfını tanıyabiliyordu ve daha küçük nesneleri algılamakta zorlanabiliyordu. Bu nedenle, Redmon ve ekibi, YOLOv2 ve sonrasındaki sürümleri geliştirmeye devam ettiler. YOLOv2 ve sonrası sürümleri, daha yüksek doğruluk oranları, daha hızlı işleme süreleri ve daha fazla nesne sınıfını algılama özelliği gibi iyileştirmeler içeriyor.

YOLOv3, YOLO algoritmasının üçüncü sürümüdür ve 2018'de piyasaya sürülmüştür. YOLOv3, görüntüleri saniyede 60 kare hızında işleyebilen gerçek zamanlı bir nesne algılama algoritması Darknet sinir ağı mimarisinin bir varyantını kullanır. Girdi görüntüsünü bir SxS hücre ızgarasına böler ve her hücre için sınırlayıcı kutuları ve nesnellik puanlarını tahmin eder. Tespit edilen hedef dışında kalan noktaları ortadan kaldırmak için Maksimum Olmayan Bastırma (Non-Maximum Suppression) yöntemi kullanılarak nesne tespiti gerçekleştirilmektedir.

YOLOv3, farklı ölçeklerde nesne algılamayı iyileştirmek için bir özellik FPN (Feature Pyramid Network) piramit ağı ve farklı şekillere sahip nesnelerin algılanmasını iyileştirmek için her hücre için birden fazla bağlantı kutusu (önceki kutular) kullanır.

YOLOv3 mimarisi ile artık blok tabanlı omurga, çok ölçekli tahmin piramit ağı, normalleştirme, bağlantı kutuları tahmini gibi birçok teknoloji bir arada kullanılmaktadır.

YOLOv3 bir nesne algılama (object detection) modelidir. YOLOv3, önceki versiyonlarına kıyasla daha hızlı ve daha doğru bir nesne algılama performansı sunar. Model, görüntüyü tek seferde taramak için öğrenme tabanlı bir algoritma kullanır ve görüntüyü birçok küçük bölgeye bölmek yerine, ağın tamamını tek bir devrede çalıştırır. Bu nedenle YOLOv3, diğer geleneksel nesne algılama yöntemlerine kıyasla daha hızlı ve daha az hesaplama gerektirir.

YOLOv3, nesne tespiti ve sınıflandırma için kullanılabilir ve geniş bir uygulama yelpazesine sahiptir. Örneğin, trafikteki araçları, yayaları ve bisikletleri tespit etmek için kullanılabilir. Ayrıca, nesneleri tespit etmek için endüstriyel kalite kontrol ve güvenlik sistemlerinde kullanılabilir. Bunun yanı sıra, tıbbi görüntüleme uygulamalarında da kullanılabilir, örneğin X-ışını görüntüleri üzerinde anormallikleri tespit etmek için kullanılabilir.

YOLOv3 mimarisi birkaç farklı bileşen içerir. İlk olarak, ağ bir giriş görüntüsünü alır ve konvolüsyonel tabakalar kullanarak bu görüntüyü özellik haritalarına dönüştürür. Bu özellik haritaları daha sonra algılama için kullanılacak nesnelerin özelliklerini içeren özellik vektörleri haline getirilir. Ardından, özellik vektörleri bir veya daha fazla tamamen bağlı (fully connected) katmanla birleştirilir. Son olarak, bir sınıflandırma çıktısı ve bir nesne sınırı tahmini oluşturmak için bu tamamen bağlı katmanların çıktıları kullanılır.

YOLOv3, YOLO ailesinin önceki sürümlerine kıyasla daha derin ve daha karmaşık bir mimariye sahiptir. Bu mimaride, çeşitli ölçeklerdeki nesneleri algılamak için birçok farklı boyutlu özellik haritası kullanılır ve bu haritalar daha sonra birleştirilir. Ayrıca, YOLOv3, önceki sürümlerinden daha hızlıdır ve daha yüksek doğruluk oranlarına sahiptir.

4.3.5. YOLOv5

YOLOv5, Ultralytics tarafından geliştirilen açık kaynaklı YOLO algoritmasının beşinci versiyonu olan, tek aşamalı bir dedektör ve bölge tabanlı nesne algılama ağıdır. Bu model, PyTorch kütüphanesinde geliştirilmiştir.
Bir görüntü veya video içindeki nesneleri gerçek zamanlı olarak yüksek doğruluk ve hız ile algılayıp sınıflandırabilen son teknoloji ürünü bir bilgisayarla görme modelidir. YOLOv5, den tasarlanmış bir mimari ve daha küçük veri kümelerinde eğitime odaklanma, çeşitli nesne boyutları için destek ve iyileştirilmiş hız ve doğruluk gibi özelliklerle YOLO'nun önceki sürümlerine göre bir gelişmedir.

YOLOv5, 2020 yılında piyasaya sürülen YOLO algoritmasının beşinci sürümüdür. YOLOv3'e göre çeşitli iyileştirmelere sahiptir ve daha hızlı ve daha doğru olduğu düşünülmektedir. YOLOv3 ile benzer bir sinir ağı mimarisi kullanır, ancak bazı modifikasyonlar ve optimizasyonlar vardır. Farklı ölçeklerdeki özellikleri yakalamak

ve nesne algılama doğruluğunu iyileştirmek için "SPP" (Spatial Pyramid Pooling) adı verilen bir teknik kullanır.

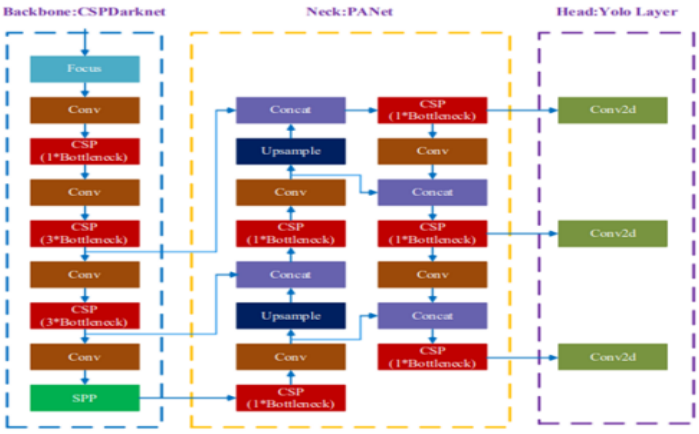
YOLOv5, parametre sayısını azaltan ve ağın verimliliğini artıran bir "CSP" (Kademeler Arası Kısmi) bağlantı modülü sunar.

Önceki sürümlerden daha hızlı ve daha doğru olan sınırlayıcı kutuları ve sınıf olasılıklarını tahmin etmek için bir "YOLOv5 head" kullanır ve önceki sürümlere göre daha fazla konvolüsyon katmanı ve daha fazla parametre içerir. Bu sayede, daha yüksek hassasiyetle ve daha hızlı sonuçlar elde etmek mümkün olur.

YOLOv5, farklı donanım ve performans gereksinimlerine uyacak şekilde çeşitli boyutlarda (küçük, orta, büyük ve ekstra büyük) olabilir.

Genel olarak YOLOv5, hız ve doğruluk açısından YOLOv3'e göre önemli bir gelişme olarak kabul edilir ve çeşitli uygulamalarda nesne algılama için yaygın olarak kullanılır.

Yolov5, 3 ana bölümden oluşan bir mimari yapısı kullanır: omurga, baş ve boyun. Omurga üzerindeki ESA katmanında, farklı ölçeklerdeki giriş görüntüsünün özellikleri çıkarılır. Boyun bölümü, omurgadan aldığı özelliklerle bir özellik haritası oluşturur ve bunu bir sonraki bölüm olan tahmin katmanına taşır. Baş bölümünde, boyun bölümünden alınan özelliklerle yerelleştirme ve sınıflandırma yapılır.



Şekil 4.14.YOLOv5 mimari yapısı [84].

Faster R-CNN ve YOLOv5 arasındaki farklar:

YOLOv5 ve Faster R-CNN, derin öğrenmedeki popüler nesne tanıma algoritmalarıdır. Her birinin avantajları ve dezavantajları aşağıdaki gibidir:

YOLOv5:

Avantajları:

1. Faster R-CNN ve diğer iki aşamalı algılayıcılardan daha hızlıdır.
2. Basit mimarisi nedeniyle eğitimi ve kullanımı daha kolaydır.
3. Pek çok benchmarkta rekabetçi performans sağlar.
4. Otonom sürüş, video gözetimi ve robotik gibi gerçek zamanlı uygulamalarda kullanılabilir.

Dezavantajları:

1. Özellikle küçük nesnelerde iki aşamalı algılayıcılara göre daha düşük doğruluğa sahip olabilir.
2. Karmaşık nesne yerleşimleri veya karmaşık sahnelerde iyi performans gösteremeyebilir.

Faster R-CNN:

Avantajları:

1. Küçük nesneler dahil olmak üzere nesne algılama benchmarklarında yüksek doğruluk sağlar.
2. Karmaşık nesne yerleşimleri ve yoğun sahnelerle başa çıkabilir.
3. Nesne izleme ve örnek bölütleme gibi diğer görevler için kullanılabilen bölge önerileri sağlar.
4. Nesnelerin ölçek ve en-boy oranı farklılıklarına karşı sağlamdır.

Dezavantajları:

1. YOLOv5 ve diğer bir aşamalı algılayıcılardan daha yavaştır.
2. Daha karmaşık bir mimari ve eğitim prosedürü vardır.
3. Daha zor kullanılır ve daha fazla hesaplama kaynağı gerektirir.

Özetle, YOLOv5 daha hızlı ve daha kolay kullanılır ancak bazı doğruluk özelliklerinden ödün verirken, Faster R-CNN daha doğru ama daha yavaş ve daha karmaşık. İki algoritma arasındaki tercih, uygulamanın özel ihtiyaçlarına ve kısıtlamalarına bağlı olacaktır.

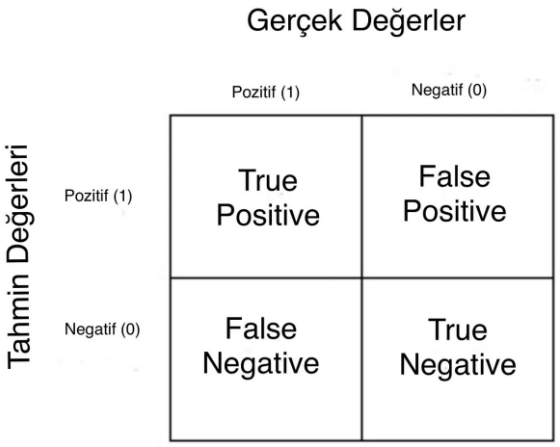
4.4. Nesne Tanıma Değerlendirme Metrikleri

4.4.1. Nesne tanıma için karmaşıklık matrisi

Bir sınıflandırma modelinin performansını değerlendirmek için kullanılan bir matrisdir. Karmaşıklık matrisi (Confusion matrix), gerçek sınıfların ve tahmin edilen sınıfların sayısal değerlerini gösterir ve sınıflandırma modelinin doğruluğunu, hassasiyetini, özgüllüğünü ve diğer performans metriklerini hesaplamak için kullanılabilir. Karmaşıklık matrisi, dört farklı değeri içerir: true positive (TP), false positive (FP), true negative (TN) ve false negative (FN). TP, gerçek pozitif sayısını, FP yanlış pozitif sayısını, TN gerçek negatif sayısını ve FN yanlış negatif sayısını ifade eder. Bu değerlerin kombinasyonu, sınıflandırma modelinin performansını değerlendirmek için kullanılabilir.

Karmaşıklık matrisi, değerlendirme ölçütlerinden biridir. Bir karmaşıklık matrisi, doğruluk değerleri/örnekleri verilen bir sınıflandırıcının performansını gösteren bir tablodur. Ancak nesne tanıma için karmaşıklık matrisinin hesaplanması için ilk olarak, başka bir destekleyici metriği anlamak gerekir. Nesne tanıma metriklerinin hesaplanmasında Intersection over Union (IoU) önem taşır.

Bir karmaşıklık matrisi, doğru pozitif (TP), gerçek negatif (TN), yanlış pozitif (FP) ve yanlış negatif (FN) olmak üzere 4 bileşenden oluşur. Tüm bileşenleri tanımlamak için, IoU'ya dayalı bir eşik (örneğin α) tanımlamamız gerekir.



Şekil 4.15. Karmaşıklık matrisi [86].

1) Gerçek Pozitif (TP): Bu, gerçek gerçekten pozitif olduğunda sınıflandırıcının pozitif tahmin ettiği bir örnektir, yani IoU ≥ α olan bir tespittir.

2) Yanlış Pozitif (FP): Bu yanlış bir pozitif algılamadır, yani IoU < α olan bir algılamadır.

3) Yanlış Negatif (FN): Bu, sınıflandırıcı tarafından algılanmayan gerçek bir örnektir.

4) Gerçek Negatif (TN): Bu metrik, gerçek örneğin de negatif olduğu göz önüne alındığında negatif algılama anlamına gelir.

4.4.2. Ground truth (Kesin Referans)

Ground truth, nesne tanıma algoritmalarında kullanılan bir terimdir ve algoritmanın doğruluğunu ölçmek için kullanılan bir referans veri kümesini ifade eder. Ground truth verileri, gerçek dünya nesnelerinin özelliklerinin doğru bir şekilde etiketlendiği ve belgelendiği bir veri kümesidir. Bu veriler, bir nesne tanıma algoritması için bir eğitim veri kümesi veya test veri kümesi olarak kullanılabilir. Eğitim veri kümesi olarak kullanıldığında, ground truth verileri, algoritmanın öğrenmesine yardımcı olur ve algoritmanın doğru sonuçları üretmesi için gereken doğru özellikleri tanımlar. Test veri kümesi olarak kullanıldığında, ground truth verileri, algoritmanın doğruluğunu ölçmek için kullanılır. Algoritmanın tahmin ettiği sonuçlar ground truth verileriyle karşılaştırılır ve algoritmanın doğruluğu ölçülür. Bu kavramlar bazı şematik örneklerle sezgisel olarak anlaşılabilir (IoU eşiğini α = 0.5 olarak belirlediğimizde).

Şekil 4.16. Sol: False Negative (FN) , Sağ: True Positive (TP) [86].

Şekil 4.17. İki görüntüde False Positive (FP) [86].

IoU eşik değerinin tanımına göre, Eşik değeri 0.86'nın üzerinde seçilirse Şekil 4.14 sağ tarafı doğru pozitif (TP) olarak kabul edilir ve IoU eşik değeri 0.14'ün altına düşürülürse Şekil 4.15 sağ tarafı yanlış pozitif (FP) olarak kabul edilir.

4.4.3. Precision ve recall değerleri

Nesne tanıma algoritmaları, görüntülerdeki nesneleri tanımak için kullanılır. Bu algoritmalar, önceden eğitilmiş bir model kullanarak bir görüntüdeki nesneleri tespit etmeye çalışır. Ancak, tespit edilen nesnelerin doğruluğu ve eksiksizliği algoritmanın performansı açısından önemlidir. Precision ve Recall, nesne tanıma algoritmalarındaki performans ölçümleridir.

Precision (kesinlik), doğru pozitif olarak tahmin edilen nesnelerin toplam tahmin edilen nesnelere oranıdır. Yani, doğru olarak sınıflandırılan nesnelerin oranıdır. Örneğin, bir nesne tanıma algoritması, bir görüntüdeki nesneleri tanımlamak için kullanılır ve 100 örnek üzerinde test edildiğinde, 80 örnek doğru bir şekilde nesne olarak tanımlandı, ancak 20 örnek yanlışlıkla tanımlandı. Bu durumda, Precision, 80/100 = 0.8 olarak hesaplanır. Yani, sınıflandırıcının doğru olarak sınıflandırdığı nesnelerin yüzdesidir. Precision, yanlış pozitifleri en aza indirmek için önemlidir. Çünkü yanlış pozitifler, gerçek pozitiflerin sayısını azaltarak, sınıflandırıcının performansını düşürebilir. Formülü denklem 4.1'deki gibidir:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{4.1}$$

Recall, bir sınıflandırıcının ilgili tüm durumları (yani, tüm temel gerçekleri) bulma yeteneğini ölçen bir ölçümdür. Tüm temel gerçekler arasında tespit edilen gerçek pozitifin oranıdır, Recall, gerçek pozitiflerin sayısını tespit etmek için kullanılır. Recall, doğru pozitiflerin gerçek pozitiflerin toplam sayısına oranıdır. Örneğin, 100 nesneden oluşan bir görüntüde, 70 nesne tanınmış ve 30 nesne tanınmamış olabilir. Bu durumda, Recall, 70/100 = 0.7 olarak hesaplanır. Recall, gerçek pozitifleri kaçırmamak için önemlidir ve denklem 4.2'deki gibi tanımlanır:

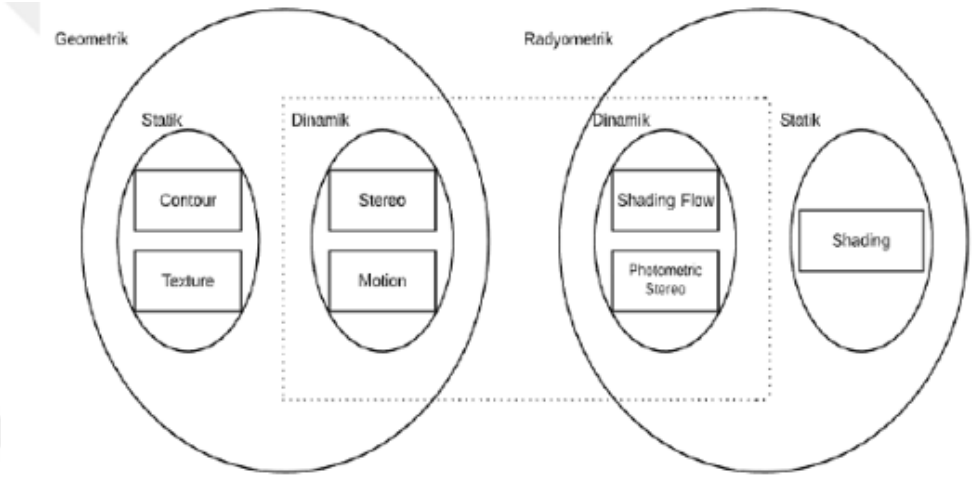
$$\text{Recall} = \frac{TP}{TP+FN} \tag{4.2}$$

Precision ve Recall değerleri, birbirleriyle bağlantılıdır ve birlikte kullanıldıklarında daha iyi bir performans ölçümü sağlarlar. Genel olarak, bir sınıflandırıcının performansı, Precision ve Recall değerleri arasındaki dengeye bağlıdır. Yüksek Precision ve düşük Recall, sınıflandırıcının daha az yanlış pozitif tahmin yapabileceğini, ancak gerçek pozitifleri kaçırma ihtimalinin daha yüksek olduğunu gösterir. Düşük Precision ve yüksek Recall, sınıflandırıcının daha fazla gerçek pozitifi tanıyabileceğini, ancak yanlış pozitiflerin sayısının da artabileceğini gösterir. Burada, True Positive (TP) doğru pozitif, False Positive (FP) yanlış pozitif ve False Negative (FN) yanlış negatif olarak adlandırılır. TP, algoritmanın doğru olarak tanımladığı nesnelerin sayısıdır. FP, algoritmanın yanlış olarak tanımladığı nesnelerin sayısıdır. FN ise algoritmanın tanıyamadığı nesnelerin sayısıdır.

4.5. Görüntü Tonlarından Şekil Elde Etme (Shape from shading)

Shape from shading (SFS), ışıklandırma ve gölgelemeye dayalı olarak nesnelerin 3B şeklini tahmin etmek için kullanılan bir görüntü işleme tekniğidir. Bu teknik, nesnelerin yüzeylerinin şekillerinin ve yüzey özelliklerinin, nesnenin üzerine düşen ışık kaynağının konumu, yoğunluğu ve yönelimi gibi ışıklandırma faktörleri tarafından belirlendiğini varsayar. Bu bilgiler kullanılarak, nesnenin yüzeyinin 3B haritası oluşturulabilir. SFS, Bilgisayar görmesi, robotik, tıp ve diğer endüstrilerde kullanılır.
Bir görüntü üzerindeki nesne ve yüzeylerden gerçek 3B şekillerin elde edilmesi ile ilgili literatürde çalışmalar bulunmaktadır. Konu ile ilgili Tonlama (Shading), Stereo, doku (texture), gölge (shadows), hareket (motion) vb. kullanarak şekil elde edilmesi

sağlanmıştır [87-88]. Seçilen yöntemi X olarak belirtirsek, genel olarak Şekil 4.16.'da gösterildiği gibi X'den şekil elde etme yaklaşımları sınıflandırılabilir.



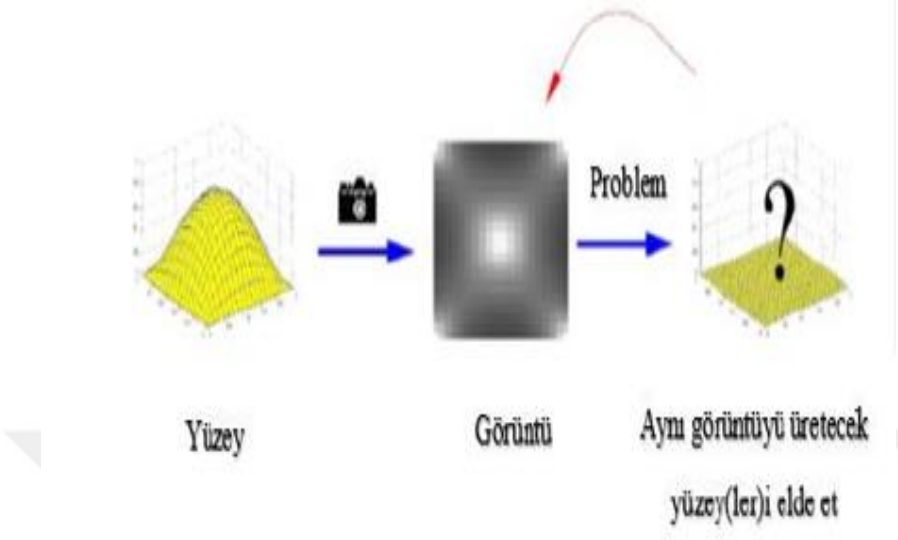
Şekil 4.18. X'den şekil elde etme yöntemleri [88].

Şekil elde etme yöntemleri için Radyometrik bilgiler ise şekil, sahne yüzeyinin pozisyonu ve yansıtma karakteristiklerinin yanı sıra aydınlatma koşulları, ortam özellikleri, sensör karakteristikleri gibi birçok sahne özelliği ile bağlantılı olmakla birlikte geometrik bilgiler ise noktalara ait yansımaların görüntü düzlemi üzerindeki koordinatlarına göre konumları vb. özelliklere dayanır [88].

Bu çalışmada verisetimiz için bir ground truth etiketleme sistemi oluşturmak için görüntü tonlarından şekil elde etme (SFS) yöntemi kullanılmıştır. 1970'lerde Horn tarafından SFS problemi ilk olarak ortaya atılmıştır [89].

Horn çalışmalarında, görüntü tonlarından şekil elde etme yaklaşımını formülüze etmiş ve çözümle ilgili öneride bulunmuştur. Nesneye ait 2B görüntü üzerinden SFS teknikleri kullanarak 3B sahneyi elde etmek amaçlanmaktadır. Görüntü üzerindeki yüzey şeklinin 3B olarak çıkartılması için SFS yöntemi ile yansıma bilgileri kullanılarak yapılmaktadır. Görüntü yoğunluğuyla yüzey şeklinin arasındaki ilişki ile tonlamadan şekil bilgisi elde edilmektedir. Yüzey materyaline bağlı olan yüzey yansıtıcılık değeri, aydınlanma yönü ve yüzey normali kullanılarek görüntü üzerindeki bir noktadaki ışıma hesaplanabilir. Görüntüye ait yansıma haritası her bir noktadaki ışıma değeri hesaplanarak elde edilir.

SFS teknikleri ise Şekil 4.19.'da gösterildiği gibi 2B bir görüntüden 3B sahnenin elde edilmesini sağlarken, kamera cihazları ise 3B bir sahnenin 2B olarak görüntülenmesini sağlar. SFS tekniği kamera cihazlarının yaptığı işlemi tersine çeviren bir yöntemdir.

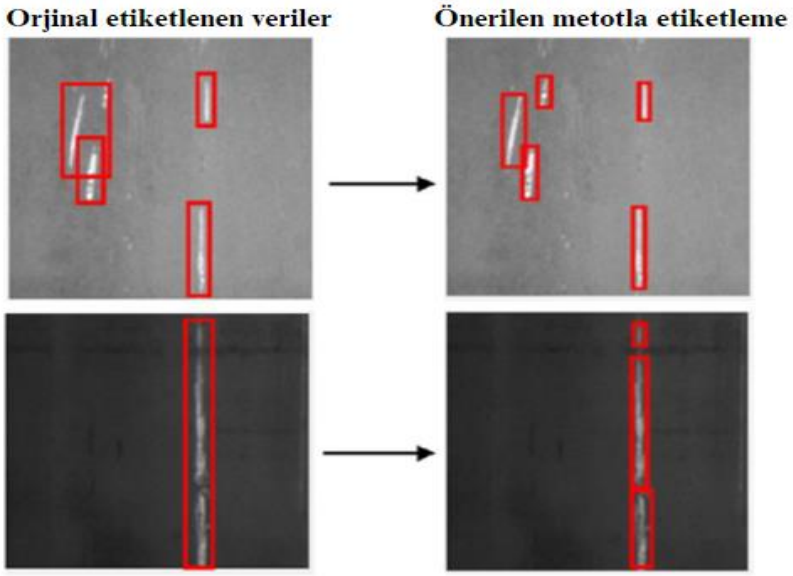


Şekil 4.19. Görüntü tonlarından şekil elde etme yöntemi [90].

3B oluşturma algoritması, geliştirilen bu sisteme uyarlanarak parçalara ait 3B derinlik haritaları elde edilmiştir ve kusurlu bölgeler etiketlenerek, derin öğrenmeye giriş olarak verilmiştir.

Görüntüler üzerinde verilen çeşitli ipuçlarını kullanarak derinlik bilgisi ve buna karşılık gelen yüzey şekilleri oluşturmayı amaçlayan 3 boyutlu rekonstrüksiyon, bilgisayarlı görü alanındaki en ilginç konulardan biridir. 3B şekiller hakkında doğru bilgi sağlayan en önemli ipuçlarından biri, gölgeleme [91-93] olarak bilinen yüzeylerden yansıyan ışığın uzamsal modelidir.

SFS, gölgelemeyi bir ipucu olarak kullanarak nesnelerin yüzeylerinde gözlemlenen yoğunluk değişimlerinin yerel yüzey hakkında nasıl bilgi sağladığını anlamaya çalışır. SFS yöntemleri, çözüm arama yöntemlerine [94], [95], dayalı olarak literatürde farklı şekilde sınıflandırılmaktadır. SFS yöntemleri, yüzey topografileri oluşturma, biyometrik çalışmalar, tıbbi görüntülerin yapılandırılması, yüzey incelemesi ve kusur tespiti dahil olmak üzere çok çeşitli uygulamalarda kullanılmaktadır. Uygun aydınlatma koşulları olduğunda, kalite kontrol sistemlerinde yaygın olarak kullanılan SFS yöntemleri tek bir görüntü üzerinde çalışıp başarılı sonuçlar üretebilmektedir. Şekil 4.20'de görüldüğü gibi, SFS kullanılarak daha hassas etiketleme yapılabilir.



Şekil 4.20. Önerilen yaklaşım (SFS) kullanılarak kusurların daha kesin olarak etiketlenmesi

5. UYGULAMA

Üretim hattında ürünlerin hızlı ve hatalı olmadan kontrol edilmesi için bilgisayar görme teknolojisi önemlidir. Klasik yöntemlerin sorunları göz önüne alındığında, bilgisayar görme kavramının önemi anlaşılmaktadır.

Bilgisayar görmesinin başlıca uygulama alanlarından bir tanesi endüstriyel otomasyonlardır. Kalite kontrol işlemi, üretim aşamasında çok önemli bir yer tutmaktadır. Üretim sürecinde kalite kontrol işlemi önem taşımaktadır çünkü üretilen ürünlerin kalitesi, hem müşteri memnuniyetini sağlamak hem de şirketin itibarını korumak için kritik bir faktördür. Kalite kontrol işlemi, ürünlerin belirlenmiş kalite standartlarına uygunluğunu kontrol etmek ve üretim sürecinde meydana gelebilecek kusurları tespit etmek için yapılmaktadır. Bu sayede, üretim sürecindeki hatalar erken aşamalarda tespit edilebilir ve düzeltilerek, ürünlerin kalitesi arttırılabilir. Kalite kontrol işlemi aynı zamanda, şirketin ürettiği ürünlerin güvenilirliği ve dayanıklılığı gibi faktörleri de sağlamaya yardımcı olur. Ürünlerin kusurlarını tespit etmek ve yerlerini belirlemek, önemli ve gerekli bir kalite kontrol sürecidir. Arıza tipinin ve arızalı alanın kısa sürede tespit edilmesi de kalite kontrol performansı açısından oldukça önemlidir.

Bilgisayar görmesi (computer vision), bir bilgisayarın dijital görüntüleri işleyerek çıktılar üretmesini sağlayan bir alan olarak tanımlanabilir. Görüntü işleme, modelleme ve analiz teknikleri kullanılarak, bilgisayarlar nesneleri ve sahneleri algılayabilir, tanıyabilir ve sınıflandırabilir. Endüstriyel otomasyon ise bir endüstriyel üretim ortamındaki üretim süreçlerinin otomatikleştirilmesini sağlayan bir alandır. Endüstriyel otomasyon uygulamaları, işletmelerin verimliliğini artırmak, ürün kalitesini sağlamak, insan faktöründen kaynaklanan hataları azaltmak ve güvenliği artırmak gibi birçok fayda sağlar.

Bilgisayar görüsü ve endüstriyel otomasyon bir araya geldiğinde, birçok uygulama ortaya çıkabilir. Örneğin, bir görüntü işleme algoritması, fabrikalarda ürünlerin kalitesinin kontrol edilmesi amacıyla kullanılabilir. Kamera sistemleri aracılığıyla toplanan veriler, bilgisayar tarafından analiz edilebilir ve üretim sürecindeki kusurları

veya yanlışlıkları tespit etmek için kullanılabilir. Benzer şekilde, bir görüntü işleme algoritması, otomotiv endüstrisinde, ilaç endüstrisinde ve birçok farklı endüstriyel uygulamada kullanılabilir. Şekil 5.1'de örnek bir endüstriyel kalite kontrol sistemi gösterilmiştir.

Endüstriyel otomasyon ve bilgisayar görüsü, işletmelerin üretim süreçlerini optimize etmelerine, hataları azaltmalarına ve ürün kalitesini artırmalarına yardımcı olur. Bu alanlardaki gelişmeler, verimlilik, güvenlik ve sürdürülebilirlik açısından büyük faydalar sağlar.

Şekil 5.1. Örnek endüstriyel kalite kontrol sistemi

Günümüzde hala, insan iş gücüne dayalı kalite kontrolü yapılmaktadır fakat bu, yorgunluk, dalgınlık ve hız kaybını beraberinde getirmektedir. Bilgisayar görmesine dayalı kalite kontrol sistemleri ile kusurun daha doğru ve daha hızlı bir şekilde ayırt edilmesini sağlamakla birlikte insan hatalarının önüne geçilmesi nedeniyle endüstriyel alanlarda çok fazla tercih edilmektedir.

Birçok endüstriyel uygulamada, tekstil, metal ve cam kusur tespiti gibi yüzey kusur tespiti yaygın olarak gerçekleşir. Metal yüzeyler aydınlatma ve ışık yansıması gibi birçok çevresel faktörden kolayca etkilendiğinden, metalik kusur tespiti zor bir problemdir. Metal yüzeylerde çeşitli ve karmaşık tipte kusurlar (Çatlaklar, yamalar, çizikler, çukurlu yüzey vb.) vardır. Gerçek zamanlı metalik hata tespit sistemlerinde hız ve yüksek doğruluk üretim aşamasına olumlu etki yapmaktadır.

Metal yüzeyler üzerindeki kusur tespit çalışmaları, üretim kalitesinin ve ürün güvenliğinin sağlanması için son derece önemlidir. Geleneksel olarak, bu tür tespit işlemleri insan gözü ile yapılan gözlemle, deneyim ve bilgi birikimine dayanır. Ancak, son yıllarda yapay zeka teknikleri, bu süreci daha hızlı, doğru ve etkili hale getirmek

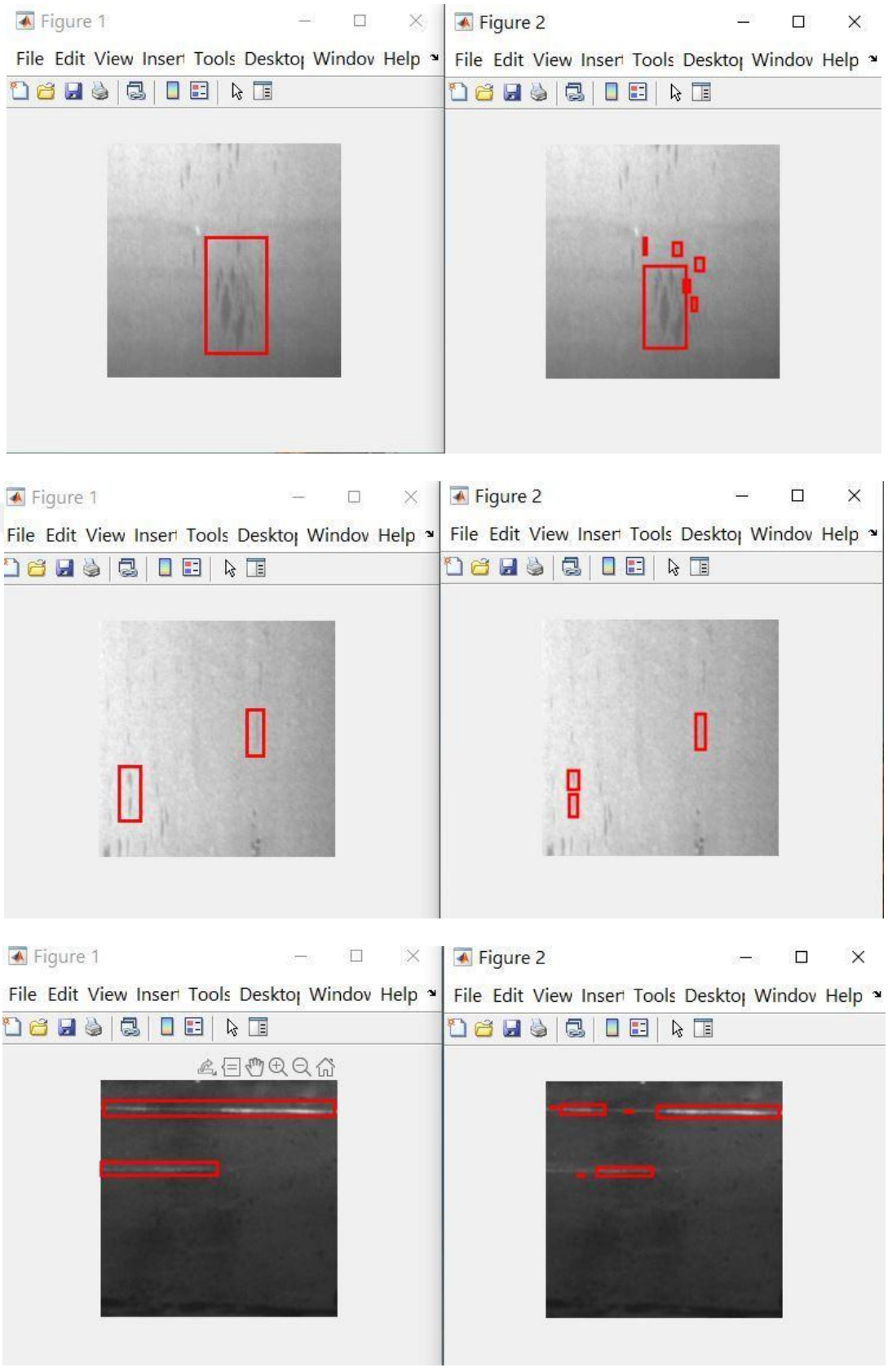
için kullanılmaktadır. Yapay zeka teknikleri, daha az insan müdahalesi ile daha doğru sonuçlar elde edilmesine olanak tanıyarak, maliyet ve zaman tasarrufu sağlar. Ayrıca, yapay zeka modellerinin doğru bir şekilde eğitilmesiyle, üretim sürecindeki kusurların önlenmesine yardımcı olabilecek veriler toplanabilir.

Yapay zeka teknikleri, görüntü işleme, makine öğrenmesi ve derin öğrenme gibi alanlarda kullanılarak, metal yüzeyler üzerindeki kusurları tespit etmek için geliştirilen algoritmalarla birlikte kullanılabilir. Görüntü işleme teknikleri, kusurlu bölgeleri otomatik olarak tespit etmek için kullanılabilir. Makine öğrenmesi ve derin öğrenme teknikleri, model öğrenme ve kusurları tespit etmek için doğru parametrelerin öğrenilmesi için veri kullanımı gibi işlemler yapabilir.

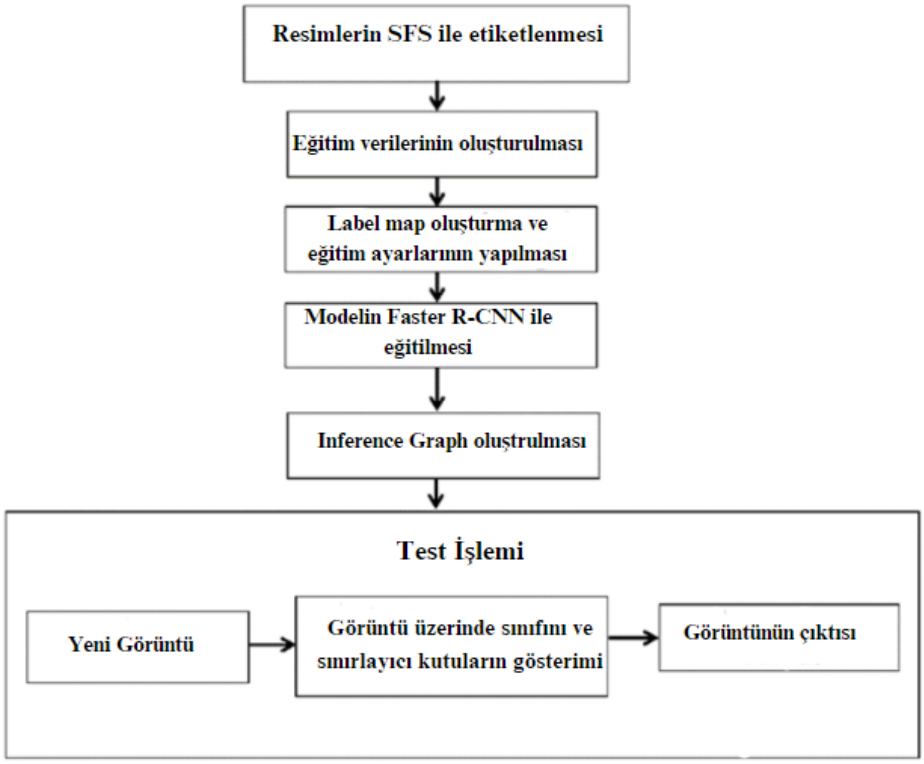
Geleneksel tespit algoritmaları karmaşık problemlerde verimsiz olduğundan, bu çalışmada, çatlamalar, çizikler, içerik vb. metal yüzey kusurlarının tespiti ve sınıflandırılması için yeni bir yöntem önerilmiştir. Sistemin eğitimi esnasında kusurlu bölgelerin yeri, yüzey özelliklerini çıkarabilen Shape From Shading (SFS) yöntemi ile etiketlendirilerek, kusurlu bölgenin yeri ve türü Faster Regional Convolutional Neural Network (Faster R-CNN) kullanılarak tespit edilmiştir.

Kusurlu örnekler için Northeastern University (NEU) yüzey kusur veritabanı kullanılmıştır. Önerilen algoritma ayrıca etiketleme performansını göstermek için etiketlenmemiş bir veri kümesi (KolektorSDD2/KSDD2) üzerinde test edilmiştir. Etiketlenmiş ve etiketlenmemiş veri kümelerindeki sonuçlar, otomatik hata tespiti, sınıflandırma ve etiketleme için en son teknoloji performansını göstermiştir. Önerilen yöntem, metal yüzeydeki kusurların tespiti için tatmin edici sonuçlar vermiş ve ortalama doğruluk oranı 0.83'tür. Çatlaklar, noktalı yüzeyler, yamalar, çizikler, madde karışımı ve girintili tufal ortalama doğruluğu sırasıyla 0.98, 0.81, 0.90, 0.79, 0.88 ve 0.62'dir.

Şekil 5.2’de görüntülerin orijinal etiketli hali ve SFS yöntemi ile etiketlenmiş hali görülmektedir. SFS ile etiketlenen görüntülerin çok daha detaylı olduğu görülmektedir.



Şekil 5.2. Figure 1: Verilerin orijinal etiketli hali, Figure 2: Verilen SFS yöntemi ile etiketlenmiş hali



Şekil 5.3. Önerilen yöntemin blok diyagramı

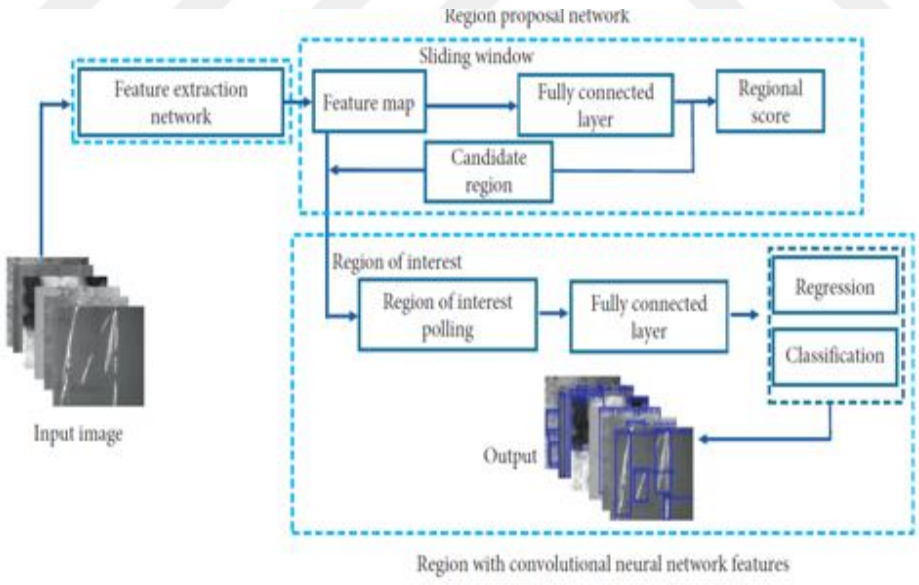
Şekil 5.3'de önerilen yönteme ait blok diyagramı gösterilmiştir. Günümüzde nesne tanıma algoritmaları kullanılması için öncesinde yapılan verilerin eğitimi için gerçekleştirilen etiketleme işlemi, hala manuel olarak yapılır. Eğitim sürecinde Faster R-CNN ile sınıflandırma yaparken, hatalı bölgelerin her biri ayrı ayrı etiketlenmelidir. Bu sürecin otomatik hale getirilmesi, endüstriyel kontrol sistemlerinde çok önemlidir, çünkü veri kümesinin tek tek etiketlenmesi çok zaman alıcıdır. Bu çalışmanın amacı, etiketleme sürecini otomatik hale getirerek zaman kazanmak için SFS ve Derin Öğrenme yöntemlerini birleştirilerek yeni bir yöntem oluşturmaktır. Faster R-CNN ile nesne tanıma ve sınıflandırma uygulamalarında başarılı performans elde etmek için, eğitim sürecinde büyük miktarda etiketli veriye ihtiyaç vardır. Bu çalışmada kullanılan NEU ve KolektorSDD2 veri kümeleri, sırasıyla 1800 ve 3335 görüntü içermektedir. Görüntülerin manuel olarak etiketlenmesi uzun zaman alırken, etiketleme işleminin SFS ile otomatikleştirilmesi, çalışmamızda çok daha kısa sürede önemli ölçüde daha iyi sonuçlar vermiştir.

Verilerin el ile etiketlenmesi, zaman alıcı bir işlemdir. Bu nedenle, algoritmaların kullanılmasıyla bu işlem hızlandırılabilir ve daha verimli hale getirilebilir. Böylelikle,

veri etiketleme sürecinde daha az insan hatası olmasına da yardımcı olur. Algoritmalar, verileri önceden belirlenmiş özelliklere göre otomatik olarak etiketleyebilir. Bu işlem, insanlar tarafından yapılan el ile etiketleme işlemine göre, özellikle büyük veri kümelerinde çok daha hızlı ve doğru sonuçlar vererek büyük bir avantaj sağlar. Bu yöntem ile yapılan uygulamalar veri bilimcilerin verileri daha hızlı analiz etmelerine ve daha doğru sonuçlar elde etmelerine olanak tanır.

Bu çalışmada, metal bileşenlerin yüzeylerinde meydana gelebilecek çeşitli hata tiplerinin tespit edilmesi amaçlanarak, yüzey özelliklerini çıkarabilen Shape From Shading (SFS) yöntemi ile NEU ve KSDD2 veri setlerinden alınan görüntülerin hatalı bölgelerin otomatik olarak etiketlendirilmiştir. Böylece büyük veri kümelerinin manuel olarak etiketlenmesi el ile yapılmak yerine, otomatik olarak gerçekleştirilerek zaman konusunda verim sağlanmaktadır.

Ayrıca, hataların pozisyonlarını etiketleme sırasında yorgunluk ve devamsızlık gibi insan hatalarını ortadan kaldırarak, hata yerlerinin daha hassas etiketlenmesi sağlanmıştır. SFS ile işlenen görüntüler, son zamanlarda popüler olan ve başarılı sonuçlar veren bir yöntem olan Faster R-CNN tarafından sınıflandırılmıştır.

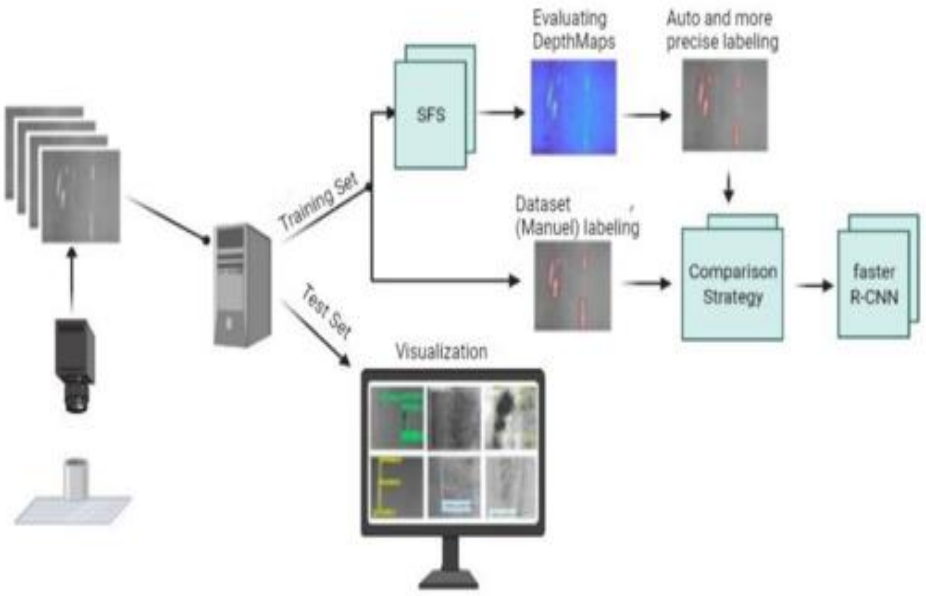


Şekil 5.4. Uygulamamızda kullanılan Faster R-CNN yapısı [96].

Görüldüğü gibi, Faster R-CNN mimarisi RPN (Region Proposal Network) olarak bölge öneri algoritması ve dedektör ağı olarak Fast R-CNN'i içerir. Faster R-CNN'de, ilk CNN uygulanır ve bir özellik haritası oluşturulur. Bölge önerileri bölümünde, seçici bölge araması yerine, ayrı bir bölge öneri ağı oluşturularak bölgeler seçilir.

Faster R-CNN modeli, önerilen bölgeleri oluştururken "Bölge Öneri Ağı" kullanır. Bu algoritma, evrişim katmanında oluşturulan özellik haritası üzerinde kayan bir pencere oluşturur ve her bölgede bir nesne varsayımında bulunarak bölgelere tarafsız puanlar atar. Bu işlem, bitişik pikseller, renk ve yoğunluk gibi kriterlere bakılarak yapılır. Ardından, "Relu" aktivasyon fonksiyonunu kullanarak bir özellik haritası oluşturulur.

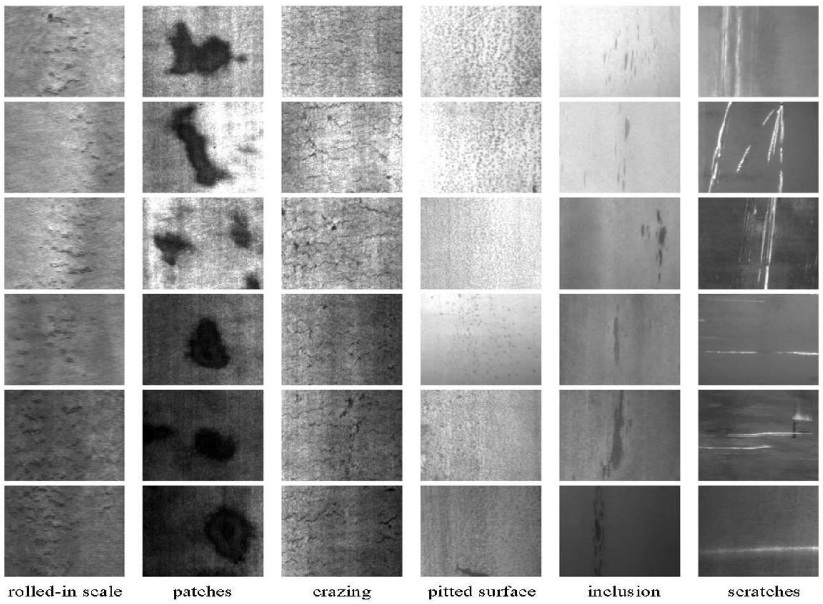
Geliştirilen yöntem, SFS ve Faster-RCNN yöntemlerinin güçlü yeteneklerini birleştirmeyi amaçlamaktadır. Ayrıca, kusur bölgelerini otomatik olarak etiketleyerek eğitim aşamalarını otomatikleştirmeyi hedefler. Başlangıçta, SFS yöntemi ve görüntü işleme teknikleri kullanılarak derinlik haritaları metal yüzey görüntülerindeki kusurlu alanları tespit etmek için kullanılır. Faster R-CNN yöntemi kullanılarak, etiketlenmiş veriler için bir kusur tespit modeli eğitilir ve otomatik olarak kusurlu alanları sınıflandırır ve tespit eder. Geliştirilen yöntemin genel şeması Şekil. 5.5'de sunulmaktadır.



Şekil 5.5. Önerilen yöntemin genel şeması

Eğitim görüntüleri önce SFS yöntemine girdikten sonra, görüntü gradyanları kullanılarak derinlik haritaları elde edilir. Farklı parametrik eşikler kullanılarak derinlik haritalarına piksel bazında maske ve ölçümler uygulanır ve görüntüdeki hatalı bölgeler tespit edilir. Binary görüntüdeki her bir 8-bağlantılı bileşen için özellik seti değerlendirilir ve her maske için tüm nesnelerin (hatalı bölgeler) min-max konumları belirlenir.

Hatalı bölgelerin sınırlama kutuları (xmin, xmax, ymin, ymax) ve ilgili genişlik ve yükseklik değerleri yapılara kaydedilir. Görüntü ve tüm hatalı bölgelere ilişkin bilgiler daha sonra Pascal VOC XML formatına dönüştürülür. Ayrıca, XML dosyaları her bir görüntü için oluşturulduktan sonra karşılaştırma stratejisi uygulanır. Karşılaştırma adımı, ground truth dışında ve içinde tespit edilen hataları incelemek için eklenmiştir. Veri kümeleri tarafından sunulan orijinal ground truth sınırlama kutuları ve SFS kullanılarak üretilen bounding boxes karşılaştırılır ve model doğruluğu analiz edilir. Kusurlu bölgelerin etiketleri ve girdi görüntüleri Faster R-CNN ile işlenerek eğitim aşamalarında kullanılarak hataların tespiti için bir model oluşturulmuştur.



Şekil 5.6. NEU veriseti [97].

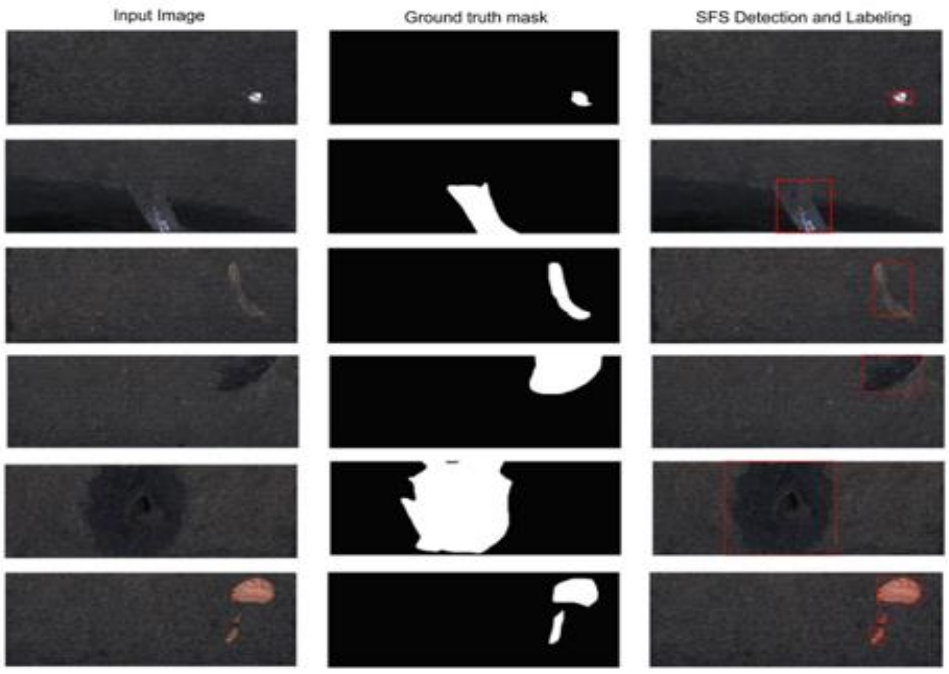
Şekil 5.6'da görüldüğü gibi, NEU veri seti (Neu Surface Defect Database) metal yüzeyinde oluşan kusur türlerini içerir. Neu veri seti altı farklı kusur türü içerir: Girintili tufal (rolled-in scale -Rs), madde karışımı (inclusion-In), yamalar (patches-P), noktalı yüzeyler (pitted surface-Ps), çatlak (cracked-Cr) ve çizik (scratches-Sc). Toplanan kusurlar sıcak haddelenmiş çelik şeridin yüzeyindedir. Veri seti 1800 gri tonlamalı görüntü içerir ve her yüzey kusur sınıfında 300 örnek bulunur. Kusurların ayrıntıları şöyledir:

Şekil 5.6'da, metal yüzeyindeki altı farklı yüzey kusurunun örnek bir görüntüsünü 200x200 piksel çözünürlükte göstermektedir. Aynı kusur sınıfı içinde benzer kusurlar olsa da, küçük farklılıklar vardır. Örneğin, çizik verileri incelendiğinde, çiziklerin

yatay ve dikey olarak farklı olduğu görülmektedir. Ancak, aydınlatma ve kusur tipi değişikliği nedeniyle farklı sınıflar arasında önemli farklılıklar vardır.

SFS algoritması, SFS'nin etiketleme performansını test etmek için Kolektor Surface Defect Dataset (KolektorSDD2 / KSDD2) üzerinde de test edilmiştir. KSDD2, metal yüzeyinde meydana gelen etiketlenmemiş ve sınıflandırılmamış farklı türlerdeki kusurları (çizikler, küçük noktalar, yüzey kusurları vb.) içerir ve veri setinde 356 adet defolu ve 2979 adet defosuz resim bulunmaktadır. Her görüntünün boyutu 230 x 630 pikseldir.

KSDD2 Veri seti, 246 (kusurlu) ve 2085 (kusursuz) görüntü içeren eğitim seti ve 110 (kusurlu) ve 894 (kusursuz) görüntü içeren test seti olarak bölünmüştür. Bu veri kümesinde yalnızca maske görüntüleri gerçek etiket olarak verilmiştir. Sonuçlar Şekil 5.7'de gösterilmektedir. Şekil 5.7'deki son sütunda görüldüğü gibi, giriş görüntülerindeki kusur bölgeleri etkili bir şekilde tespit edilmiş ve ayrıntılı olarak etiketlenmiştir. Ayrıca, hassasiyet derecesi parametrik olarak ayarlanabildiğinden, SFS daha hassas sonuçlar elde edebilmektedir.

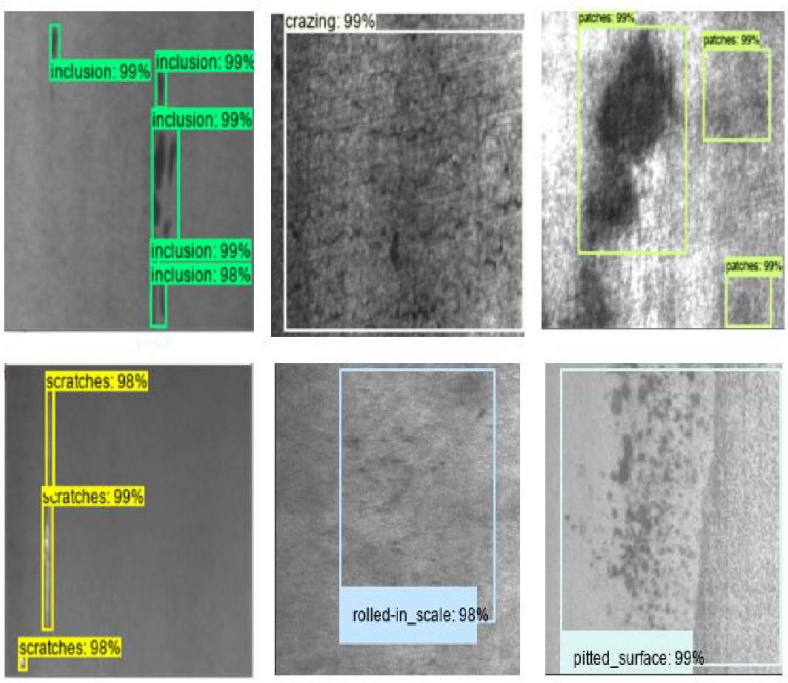


Şekil 5.7. Kolektor veri setinin SFS ile etiketlenmesi

5.1. Uygulama Sonuçları

Çalışmada, metaldeki çatlaklar, noktalı yüzeyler, yamalar, çizikler, madde karışımı ve girintili tufal gibi kusurlar SFS ve Faster R-CNN kullanılarak tespit edilmiştir. NEU veri setinden elde edilen metal yüzeyi görüntülerindeki kusur yerlerini belirlemek için SFS algoritması kullanıldı. Son yıllarda popüler hale gelen ve çok başarılı sonuçlar veren Faster R-CNN ise görüntüleri sınıflandırmak için kullanıldı.

Veri setindeki görüntüler, eğitim seti için 1440 veri ve test seti için 360 veri olarak ikiye ayrılmıştır. Yapılan çalışmamızın başarı oranı, %83 olarak belirlenmiştir. Şekil 5.1'de her bir kusur tipinin eğitim sonucu görülmektedir ve girintili tufal (rolled-in scale -Rs), madde karışımı (inclusion-In), yamalar (patches-P), noktalı yüzeyler (pitted surface-Ps), çatlak (cracked-Cr) ve çizik (scratches-Sc). gibi kusurların doğru şekilde kusur tipinin tespit edilebildiğini ve yerlerinin belirlendiğini göstermektedir.

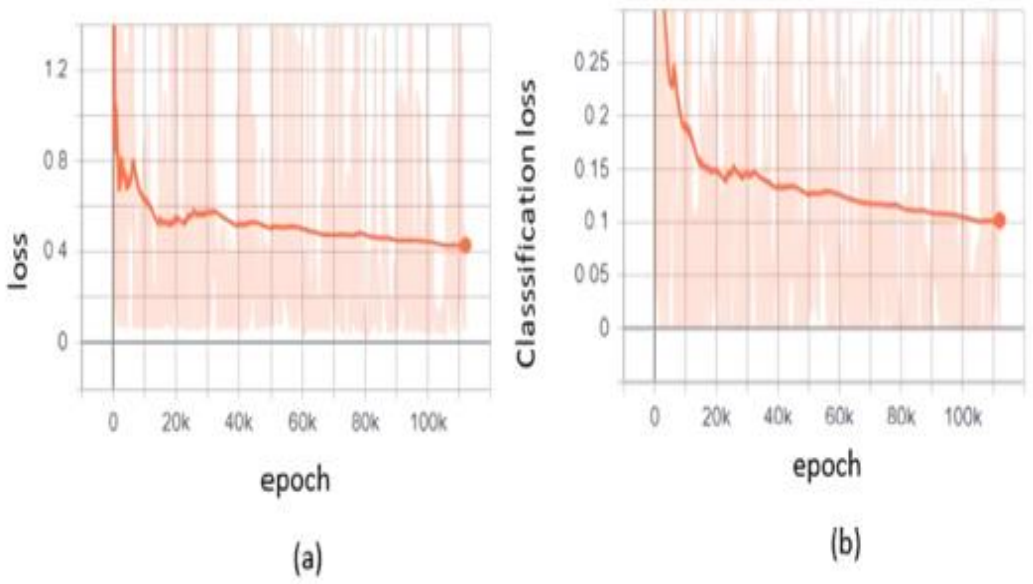


Şekil 5.1. Kusurlu bölgelerin önerilen yöntem (SFS-Faster R-CNN) ile etiketlenmesi

Geliştirilen uygulama, Intel i7 4700HQ işlemci ve NVIDIA GeForce GTX 850M grafik kartı bulunan bir bilgisayarda yazılım ortamında çalıştırılmıştır.

Ağın eğitimi, 1800 etiketli bozuk resim üzerinde 12 saat sürmüştür. Eğitim sürecimizin bir sonucu olarak, loss grafiği oluşturulmuştur. Şekil 5.2 (a) 'da gösterildiği gibi, loss değeri, adım 112k'da 0.4'e düşmektedir. Loss değeri, eğitimin erken aşamalarında

1.2'den büyüktür. Grafikte görülebileceği gibi eğitim ilerledikçe loss değeri hızla azalmaktadır. Bounding Box Classifier için loss değeri Şekil 5.2 (b)'de gösterilmiştir.



Şekil 5.2. a) Loss değerleri grafiği ve b) Classification Loss grafiği

Bu çalışmada, 1800 görüntünün etiketleme işlemi için, hatalı alanlar manuel olarak işaretlenmek yerine SFS yöntemi kullanılarak etiketlendi. Etiketlerden oluşan veri kümesi, Faster R-CNN modeline bir giriş olarak verildi. NEU Surface veri setinin %80'i eğitimde, %20'si testte kullanıldı. Çalışmamızın ortalama doğruluk oranı, eğitim seti için 1440 veri ve test seti için 360 veri kullanılarak 0.83 olarak belirlendi.

Tablo 5.1'de, NEU veri seti üzerindeki mevcut çalışmaların karşılaştırma tablosunu göstermektedir ve SFS ile birlikte kullanılan Faster R-CNN ve YOLOv5 algoritmalarının sonuçları da gösterilmektedir. YOLO ve Faster R-CNN belirli benzerliklere sahiptir. İkisi de sınırlayıcı regresyon kullanır ve çerçeve kutusu tabanlı ağ yapıları kullanır. YOLO, sınıflandırma ve çerçeve kutusu regresyonunu aynı anda gerçekleştirmesiyle Faster R-CNN'den farklıdır. Ayrıca YOLO algoritması küçük ve birbirine yakın nesneleri tespit etmekte zorluk çekmesi sebebiyle nesne tespiti konusunda bir dezavantaja sahiptir. Tablo 5.1'de görüldüğü gibi, önerilen yöntem (SFS-Faster R-CNN), en iyi mAP sonucunu vermektedir.

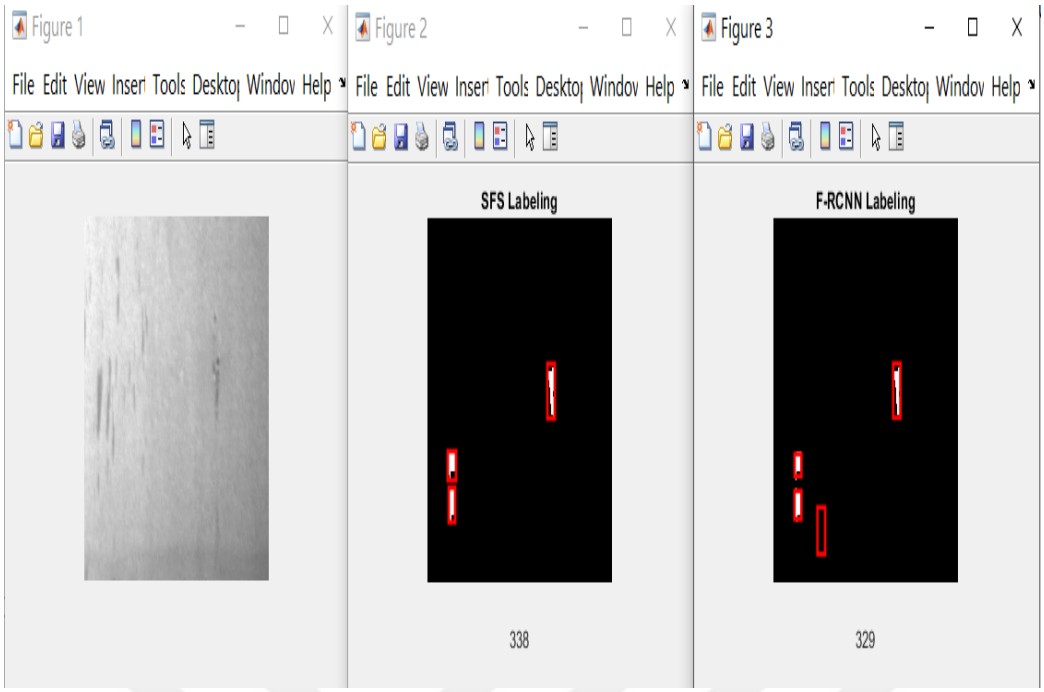
Tablo 5.1. Uygulanan algoritmaların sonuçları

Yöntem	mAP	Cr	Inc	Pa	Ps	Rs	Sc
Original Neu –Faster R-CNN	0.72	0.58	0.82	0.95	0.59	0.60	0.78
Original Neu –Yolo V5	0.48	0.31	0.65	0.83	0.47	0.36	0.3
SFS-Neu- Faster R-CNN(Proposed Method)	**0.83**	0.98	0.88	0.90	0.81	0.62	0.79
SFS- Neu- Yolo V5	0.72	0.51	0.76	0.85	0.81	0.62	0.82

Tablo 5.1'e göre, crazing en yüksek ortalama doğruluğa sahipken, patches ise ikinci sırada ve sırasıyla 0.96 ve 0.93 değerlerine sahiptir. Genel ortalama doğruluk 0.83'tür ve "rolled-in scale" için ortalama doğruluk en düşük değer olan 0.62'dir. Ayrıca, SFS-Faster R-CNN algoritması 0.83 mAP'lik sonuç vermiştir.

Bu çalışmada, metal yüzey görüntüleri üzerinde bir kusur tespiti yöntemi geliştirilmiş ve uygulanmıştır. Bu tanıma sistemi, kusur bölgelerini tespit etme ve kusur kategorilerini tanımlama açısından oldukça başarılı sonuçlar vermiştir.

Metal yüzeylerde kusur tespitinin başarı oranını hesaplatmak için piksel bazlı sonuçlar da hesaplatılmıştır. Şekil 5.3.'de Inclusion görüntüsünün piksel bazlı kusur tespiti sonuçları verilmiştir. Şekilde SFS ile etiketlenen kusurlardaki toplam kusurlu piksel sayısı 338, Faster R-CNN ile etiketlenen kusurlardaki toplam kusurlu piksel sayısı 329 olarak bulunmuştur. Burada SFS ile bulunan kusurlu piksel sayısı grand truth olarak alınmıştır. Bu durumda Faster R-CNN'nin piksel bazında kusur bulma oranı (329/338) %97'dir.



Şekil 5.3. Örnek bir görüntüye ait piksel bazlı kusur tespiti sonuçları (%97 başarı)

Her bir kusur tipi için piksel bazında ortalama sonuçlar Tablo 5.2'de görüldüğü gibidir.

Tablo 5.2. Her bir kusur tipi için piksel bazlı doğruluk oranlarının ortalama sonuçları

Kusur No ve Tipi	Sonucu (%)
Kusur no=0 , Crazing:	98.36
Kusur no=1, Inclusion:	88.28
Kusur no=2 , Patches :	96.18
Kusur no=3 , Pitted_Surface :	98.04
Kusur no=4 , Rolled_in_scale:	82.03
Kusur no=5 , Scratches :	91.724
Tüm Kusurların Ortalamaları :	**92.435**

Bu çalışmada, metal yüzey görüntüleri üzerinde bir kusur tespit yöntemi geliştirilmiş ve uygulanmıştır. Bu tanıma sistemi, kusurlu bölgelerinin tespiti ve kusur kategorilerinin tanınması açısından oldukça başarılı bir performans sergilemiştir. SFS yöntemi, kusurlu bölgelerinin daha net bir şekilde tespit edilmesi ve eğitim aşamasının güçlendirilmesi için kullanılmış, kusurun türü ve konumunun belirlenmesi için de Faster R-CNN yöntemi kullanılmıştır. Kusurlu örnekler için NEU Surface veritabanı kullanılmıştır. 1800 görüntü veri kümesinin %80'i eğitimde, %20'si testte kullanılmıştır. Ayrıca, etiketleme performansını göstermek için KSDD2 veri setinde de test edilmiştir. SFS ve Faster R-CNN yönteminin uyumlu entegrasyonu sonucu %0.83 mAP elde edilmiştir. Geliştirilen yöntemin performansı, literatürdeki diğer yöntemlerle karşılaştırılmıştır.

6. TARTIŞMA VE SONUÇ

Küçük metal parçalarının kalite kontrolünü bilgisayar görmesi yolu ile gerçekleştirmek zor bir problemdir. Otomasyonun ve seri üretimin en yaygın olduğu sektörlerden birisi de otomobil sektörüdür. Otomobil parçalarının kalite kontrolü, sürekli kalite parçaların elde edilmesi ve hatanın sıfıra indirilmesi istenmekle beraber otomobil sektöründe kullanılan parçaların kalite kontrolü gözle yapılamayacak kadar hız ve çabukluk gerektirmektedir. Genellikle seri üretimde kalite kontrolü bilgisayar ile özellikle de bilgisayar görmesi yolu ile yapılması gerekmektedir. Üretilen parçaların bir kısmı çok hassas nitelikte olup ezik, çizik kaynaklı en küçük bir fiziksel deformasyon durumunda ıskartaya çıkarılmaktadır.

Bu tez çalışmasında başta otomotiv sektörü olmak üzere, metalik parçaların yoğun olarak kullanıldığı ve kalite kontrolün elzem olduğu sistemlerde seri üretimde kullanılacak olan parçaların kalite kontrolünü başta yorgunluk olmak üzere göz yanılgısı, dalgınlık vb. insan hatalarından arındırarak elde edilebilecek en düşük hata payı ile gerçekleştirmek amaçlanmaktadır.

Ülkemizde bu tür problemler için genellikle yurt dışı kaynaklı yazılım ve donanım sistemleri kullanılmaktadır. Ayrıca geliştirilecek olan kalite kontrol sistemi ile kontrol işinde çalışan insanlarda duruş bozukluğu, göz hastalıkları vb. meslek hastalıklarının da önüne geçilecektir.

Proje çıktıları endüstriyel bilgisayar görmesi, kalite kontrolü ve otomasyon konularında birçok projenin gelişmesine fayda sağlayacaktır.

Metal yüzeylerinin aydınlatma ve ışık yansıması gibi çevresel faktörlerden kolayca etkilenmesi nedeniyle, metalik kusurların tespiti zordur. Bu çalışmada, metal yüzeyleri için uçtan uca bir otomatik kusur tespit sistemi geliştirildi. Bu tanıma sistemi, kusur bölgelerinin yanı sıra kusur kategorilerini tanıma açısından mükemmel bir performans sergilemektedir. Metal yüzeylerdeki kusurların daha kesin tespiti ve işaretlenmesi SFS algoritması kullanılarak yapılmıştır. İşaretlenen görüntüler, Faster R-CNN yöntemi kullanılarak sınıflandırılmış ve etiketlenmiştir. Kusurlu örnekler için NEU Surface veritabanı kullanılmıştır.

Bu çalışmada, CNN mimarisine dayalı bir yaklaşım olan Faster R-CNN modeli kullanılmıştır. Uygulama, Intel i7 4700HQ işlemci ve NVIDIA GeForce GTX 850M grafik kartı olan bir bilgisayarda çalıştırılmıştır. NEU veri kümesinin eğitimi, 1440 etiketli hatalı görüntü üzerinde 12 saat sürmüştür.

SFS yöntemi genellikle tek bir aydınlatma sistemi ile tek bir görüntü işler. Gelecekteki çalışmalar, fotometrik stereo gibi çoklu aydınlatma ve giriş görüntüleri olan sistemlere odaklanabilir. Bu yöntem kullanılarak, farklı yüzey malzemeleri (tekstil, cam vb.) için kusur tespiti çalışmaları yapılabilir.

KAYNAKLAR

[1] Pitts, W. S. ve .. McCullochWalter, «A logical calculus of the ideas immanent in nervous activity,» The bulletin of mathematical biophysics, cilt 5, pp. 115-133, 1943.

[2] Sejnowski, T. J., & Tesauro, G. (1989). The Hebb rule for synaptic plasticity: algorithms and implementations. In Neural models of plasticity (pp. 94-103). Academic Press.

[3] Russel, S. (2010). Artificial Intelligence: A Modern Approach. Stuart J. Russell and Peter Norvig.

[4] Seetharam, K., Kagiyama, N., & Sengupta, P. P. (2019). Application of mobile health, telemedicine and artificial intelligence to echocardiography. Echo research and practice, 6(2), R41-R52.

[5] S. J. (2012). Computer vision: models, learning, and inference Cambridge University.

[6] Szeliski, R. (2022). Computer vision: algorithms and applications. Springer Nature.

[7] X. Tao, D. Zhang, W. Ma, X. Liu, and De Xu, "Automatic metallic surface defect detection and recognition with convolutional neural networks," Appl. Sci., vol. 8, no. 9, pp. 1–15, 2018, doi: 10.3390/app8091575.

[8] E. Hoseini, F. Farhadi, and F. Tajeripour, "Fabric Defect Detection Using Auto-Correlation Function," vol. 5, no. 1, pp. 114–117, 2013, doi: 10.7763/IJCTE.2013.V5.658.

[9] S. Dash, "Handbook of Research on Modeling , Analysis , and Application of Nature- Inspired Metaheuristic Algorithms," no. May, 2018.

[10] A. K. Datta and J. K. Chandra, "Detection of Defects in Fabric by Morphological Image Processing," no. August, 2010, doi: 10.5772/10470.

[11] G. Moradi, M. Shamsi, M. H. Sedaaghi, and S. Moradi, "Using Statistical Histogram Based EM Algorithm for Apple Defect Detection," vol. 2, no. 2, pp. 10–14, 2012, doi: 10.5923/j.ajsp.20120202.02.

[12] B. Wang, "Circumferential defect detection using ultrasonic guided waves An efficient quantitative technique for pipeline inspection," vol. 37, no. 6, pp. 1923–1943, 2020, doi: 10.1108/EC-06-2019-0260.

[13] A. Kumar and G. Pang, "Defect detection in textured materials using Gabor filters," no. April 2002, 2013, doi: 10.1109/28.993164.

[14] M. Ghazvini, S. A. Monadjemi, N. Movahhedinia, and K. Jamshidi, “Defect Detection of Tiles Using 2D-Wavelet Transform and Statistical Features,” no. January, 2009.

[15] J. Zhou and J. Wang, “Fabric Defect Detection Using a Hybrid and Complementary Fractal Feature Vector and FCM-based Novelty Detector,” vol. 6, no. 126, pp. 46–52, 2017, doi: 10.5604/01.3001.0010.5370.

[16] H. Chen, J. Liu, S. Wang, and K. Liu, “Robust Dislocation Defects Region Segmentation for Polysilicon Wafer Image With Random Texture Background,” IEEE Access, vol. 7, pp. 134318–134329, 2019, doi: 10.1109/ACCESS.2019.2942218.

[17] H. Zhang, X. Jin, Q. M. J. Wu, S. Member, and Y. Wang, “Automatic Visual Detection System of Railway Surface Defects With Curvature Filter and Improved Gaussian Mixture Model,” IEEE Trans. Instrum. Meas., vol. 67, no. 7, pp. 1593–1608, 2018, doi: 10.1109/TIM.2018.2803830.

[18] F. Pernkopf, “Detection of surface defects on raw steel blocks using Bayesian network classifiers,” Pattern Anal. Appl., vol. 7, no. 3, pp. 333–342, 2004, doi: 10.1007/bf02683998.

[19] P. M. Shanbhag, “Fabric Defect Detection Using Principal Component Analysis,” vol. 2, no. 9, pp. 2863–2867, 2013.

[20] A. J. Chittilappilly and K. Subramaniam, “SVM based defect detection for industrial applications,” 2017 4th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2017, no. December 2018, 2017, doi: 10.1109/ICACCS.2017.8014696.

[21] N. J. Shipway, P. Huthwaite, M. J. S. Lowe, and T. J. Barden, “Performance Based Modifications of Random Forest to Perform Automated Defect Detection for Fluorescent Penetrant Inspection,” J. Nondestruct. Eval., vol. 38, no. 2, pp. 1–11, 2019, doi: 10.1007/s10921-019-0574-9.

[22] D. Wijesingha and B. Jayasekara, “Detection of defects on warp-knit fabric surfaces using self organizing map,” MERCon 2018 - 4th Int. Multidiscip. Moratuwa Eng. Res. Conf., pp. 601–606, 2018, doi: 10.1109/MERCon.2018.8421944.

[23] Z. Lin, H. Ye, B. Zhan, and X. Huang, “An efficient network for surface defect detection,” Appl. Sci., vol. 10, no. 17, 2020, doi: 10.3390/app10176085.

[24] L. Yi, G. Li, and M. Jiang, “An End-to-End Steel Strip Surface Defects Recognition System Based on Convolutional Neural Networks,” Steel Res. Int., vol. 88, no. 2, pp. 176–187, 2017, doi: 10.1002/srin.201600068.

[25] M. S. Kim, T. Park, and P. Park, “Classification of Steel Surface Defect Using Convolutional Neural Network with Few Images,” 2019 12th Asian Control Conf. ASCC 2019, pp. 1398–1401, 2019.

[26] S. Zhou, Y. Chen, D. Zhang, J. Xie, and Y. Zhou, “Classification of surface defects on steel sheet using convolutional neural networks,” Mater. Tehnol., vol. 51, no. 1, pp. 123–131, 2017, doi: 10.17222/mit.2015.335.

[27] D. Soukup and R. Huber-Mörk, “Convolutional neural networks for steel surface defect detection from photometric stereo images,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8887, pp. 668–677, 2014, doi: 10.1007/978-3-319-14249-4_64.

[28] D. Amin and S. Akhter, “Deep Learning-Based Defect Detection System in Steel Sheet Surfaces,” 2020 IEEE Reg. 10 Symp. TENSYMP 2020, no. July, pp. 444–448, 2020, doi: 10.1109/TENSYMP50017.2020.9230863.

[29] X. Lv, F. Duan, J. J. Jiang, X. Fu, and L. Gan, “Deep metallic surface defect detection: The new benchmark and detection network,” Sensors (Switzerland), vol. 20, no. 6, 2020, doi: 10.3390/s20061562.

[30] J. Li, Z. Su, J. Geng, and Y. Yin, “Real-time Detection of Steel Strip Surface Defects Based on Improved YOLO Detection Network,” IFAC-PapersOnLine, vol. 51, no. 21, pp. 76–81, 2018, doi: 10.1016/j.ifacol.2018.09.412.

[31] J. Fu, X. Zhu, and Y. Li, “Recognition of surface defects on steel sheet using transfer learning,” arXiv, 2019.

[32] S. Y. Lee, B. A. Tama, S. J. Moon, and S. Lee, “Steel surface defect diagnostics using deep convolutional neural network and class activation map,” Appl. Sci., vol. 9, no. 24, 2019, doi: 10.3390/app9245449.

[33] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, “PGAnet: pyramid feature fusion and global context attention network for automated surface defect detection,” IEEE Transactions on Industrial Informatics, vol. 16, no. 12, pp. 7448–7458, 2020.

[34] Q. Luo, X. Fang, Su et al., “Automated visual defect classification for flat steel surface: a survey,” IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 12, pp. 9329–9349, 2020.

[35] G. Fu, P. Sun, W. Zhu et al.,“A deep-learning-based approach for fast and robust steel surface defects classification,” Optics and Lasers in Engineering, vol. 121, pp. 397–405, 2019.

[36] X. Lv, F. Duan, and J. Jiang, “Deep metallic surface defect detection: the new benchmark and detection network,” Sensors (Basel, Switzerland), vol. 20, no. 6, 2020.

[37] M. Vannocci, A. Ritacco, and A. Castellano, “Flatness defect detection and classification in hot rolled steel strips using convolutional neural networks,” 2019.

[38] G. Song, K. Song, and Y. Yan, “EDRNet: Encoder-decoder residual network for salient object detection of strip steel surface defects,” IEEE Transactions on Instrumentation and Measurement, vol. 99, p. 1, 2020.

[39] H. Wang, J. Wang, and F. Luo, “Research on surface defect detection of metal sheet and strip based on multi-level feature Faster R-CNN,” Mechanical Science and Technology for Aerospace Engineering, vol. 2020, 2020.

[40] X. Dai, H. Chen, and C. Zhu, “Research on surface defect detection and implementation of metal workpiece based on improved Faster R-CNN,” Surface Technology, vol. 49, no. 10, pp. 362–371, 2020.

[41] Cheng, J.C.; Wang, M. Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques. Autom. Constr. 2018, 95, 155–171.

[42] Li, R.; Yuan, Y.; Zhang, W.; Yuan, Y. Unified vision-based methodology forsimultaneous concrete defect detection and geolocalization. Comput.-Aided Civil Infrastruct. Eng. 2018, 33, 527–544.

[43] Zhang, C.; Chang, C.C.; Jamshidi, M. Bridge damage detection using a single-stage detector and field inspection images. arXiv 2018, arXiv:1812.10590.

[44] Yin 2018, X.; Chen, Y.; Bouferguene, A.; Zaman, H.; Al-Hussein, M.; Kurach, L. A deep learning-based framework for an automated defect detection system for sewer pipes. Autom. Constr. 2020, 109, 102967.

[45] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.

[46] Y. LeCun, Y. Bengio ve G. Hinton, «Deep learning,» Nature, cilt 521, pp. 436-444,2015.

[47] Bozkurt, S. (2018). Derin öğrenme algoritmaları kullanılarak çay alanlarının otomatik segmentasyonu (Doctoral dissertation, Yüksek Lisans Tezi. İstanbul).

[48] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow ve A. Y. Ng, «On optimization methods for deep learning,» In Proceedings of the 28th International Conference on International Conference on Machine Learning, pp. 265-272, 2011.

[49] H. Lee, P. Pham, Y. Largman ve A. Y. Ng, «Unsupervised feature learning for audio classification using convolutional deep belief networks,» In Advances in neural information processing systems, pp. 1096-1104, 2009.

[50] Ö. İnik and E. Ülker, Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri, Gaziosmanpasa J. Sci. Res., vol. 6, no. 3, pp. 85–104, 2017.

[51] J.-S. Choi, W.-H. Lee, J.-H. Lee, J.-H. Lee, and S.-C. Kim, Deep learning based NLOS ıdentification with commodity WLAN devices, IEEE Trans. Veh. Technol., pp. 1–1, 2017.

[52] W. S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys., vol. 5, no. 4, pp. 115–133, 1943.

[53] B. Widrow and M. E. Hoff, Adaptive switching circuits, 1960 IRE WESCON Conv. Rec., pp. 96–104, 1960.

[54] F. Rosenblatt, The perceptron: a probabilistic model for ınformation storage and organization in the brain, Psychol. Rev., pp. 65–386, 1958.

[55] M. L. Minksy and S. A. Papert, Perceptrons, MIT Press, Cambridge, 1969.

[56] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning representations by back-propagating errors, Nature, vol. 323, p. 533, Oct. 1986.

[57] D. E. Rumelhart, J. Mcclelland, and L. James, Parallel distributed processing: explorations in the microstructure of cognition. Volume 1. Foundations. 1986.

[58] G. E. Hinton, S. Osindero, and Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput., vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[59] Y. Bengio and Y. Lecun, Scaling learning algorithms towards AI, in large-scale kernel machines, L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, Eds. MIT Press, 2007.

[60] O. Delalleau and Y. Bengio, Shallow vs. deep sum-product networks, in advances in neural ınformation processing systems 24, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 666–674.

[61] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[62] NVIDIA Developer Blog. (2021, March 3). Mocha.jl: Deep Learning In Julia. , https://devblogs.nvidia.com/mocha-jl-deep-learning-julia/ adresinden 1 Nisan 2022 tarihinde alınmıştır.

[63] Z. Al-Milaji, I. Ersoy, A. Hafiane, K. Palaniappan, and F. Bunyak, Integrating segmentation with deep learning for enhanced classification of epithelial and stromal tissues in H&E images, Pattern Recognit. Lett., Sep. 2017.

[64] V. Nair and G. E. Hinton, Rectified linear units ımprove restricted boltzmann machines, in Proceedings of the 27th International Conference n International Conference on Machine Learning, 2010, pp. 807–814.

[65] MathWorks.(n.d).Convolutional Neural Network(CNN), http://www.mathworks .com/solutions/deep-learning/convolutional-neural-network.html., adresinden 1 Nisan 2022 tarihinde alınmıştır.

[66] Singh, R. (n.d.). Convolutional Layer - Machine Learning. http://machinelearnin guru.com/computer_vision/basics/convolution/convolution_layer.html., adresinden 1 Nisan 2022 tarihinde alınmıştır.

[67] Roboflow Blog.(n.d). Preprocessing Step, https://blog.roboflow.com/when-to-use-grayscale-as-a-preprocessing-step/ adresinden 1 Nisan 2022 tarihinde alınmıştır.

[68] S. Srinivas, R. K. Sarvadevabhatla, K. R. Mopuri, N. Prabhu, S. Kruthiventi, and R. V. Babu, A taxonomy of deep convolutional neural nets for computer vision, Front. Robot. AI, Jan. 2016.

[69] G. G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv, Jul. 2012.

[70] Revolution Analytics Blog.(2016). Deep Learning Part 2: Setting up a Deep Learning Environment, and Building a Handwritten Digit Classifier using MNIST,https://blog.revolutionanalytics.com/2016/08/deep-learning-part-2.html. adresinden 1 Nisan 2022 tarihinde alınmıştır.

[71] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res., vol. 15, pp. 1929–1958, 2014.

[72] Gupta, A. (2019, January 28). Understanding Convolution Operations in CNN. https://medium.com/analytics-vidhya/understanding-convolution-operations in-cnn-1914045816d4, adresinden 1 Nisan 2022 tarihinde alınmıştır.

[73] DeepVisionGuru. (2021, August 6). Train License Plates Detection Model using Detectron2. https://medium.com/deepvisionguru/train-license-plates-detection-model-sing-detectron2-dd166154f604 adresinden 1 Nisan 2022 tarihinde alınmıştır.

[74] R. Girshick, « Fast r-cnn,» Proceedings of the IEEE international conference on computer vision, pp. 1440-1448, 2015.

[75] R. Girshick, J. Donahue, T. Darrell ve J. Malik, «Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,» The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580-587, 2014.

[76] V. VN., «An Overview of Statistical Learning Theory,» IEEE Transactions on Neural Networks, cilt 10, no. 5, pp. 988-999, 1999.

[77] T. Hill, L. Marquez, M. O'Connor ve W. Remus, «Artificial neural network models for forecasting and decision making,» Int. J. Forecast, cilt 10, no. 1, pp. 5-15, 1994.

[78] S. Ren, K. He, Girshick ve J. Sun, «Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Transactıons On Pattern Analysıs And Machıne Intellıgence, cilt 39, no. 6, pp. 1137-1149, 2017.

[79] Analytics Vidhya. (2018, October 25). A Step-by-Step Introduction to the Basic ObjectDetection Algorithmshttps://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/ adresinden 1 Nisan 2022 tarihinde alınmıştır.

[80] Region of Interest, https://towardsdatascience.com/region-of-interest-pooling-f7c637f409af adresinden 1 Nisan 2022 tarihinde alınmıştır.

[81] Patil, R. (2021, January 2). R-CNN, Fast R-CNN, Faster R-CNN & YOLO – Object Detection Algorithms. https://towardsdatascience.com/r-cnn-fast-r-cnn faster-r-cnn-yolo-object-detection-algorithms-36d53571365e adresinden 1 Nisan 2022 tarihinde alınmıştır.

[82] GeeksforGeeks. (n.d.). Faster R-CNN. https://www.geeksforgeeks.org/faster-r-cnn-ml adresinden 1 Nisan 2022 tarihinde alınmıştır.

[83] Engineering Education. (2021, March 9). Introduction to YOLO Algorithm for Object Detection. https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection adresinden 1 Nisan 2022 tarihinde alınmıştır.

[84] Fang, Y., Guo, X., Chen, K., Zhou, Z., & Ye, Q. (2021). Accurate and Automated Detection of Surface Knots on Sawn Timbers Using YOLO-V5 Model. *BioResources*, *16*(3).

[85] A Survey on Performance Metrics for Object-Detection Algorithms, Rafael Padilla.

[86] Koech, K. (2020). Confusion Matrix and Object Detection. https://towardsdata science.com/confusion-matrix-and-object-detection-f0cbcb634157 adresinden 1 Nisan 2022 tarihinde alınmıştır.

[87] Cryer, J. E., Tsai, P., Shah, M., Combining shape from shading and stereo using human visual model. UCF Technical Report, Florida, 1992.

[88] Negahdaripour, S., Revised definition of optical flow: Integration of radiometric and geometric cues for dynamic scene analysis. IEEE T. Pattern Anal., 20(9), 961-979,1998.

[89] Horn, B.K.P., Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View. Massachusetts Institute of Technology, 1970.

[90] Prados, E., Faugeras, O., Shape From Shading.: Handbook of Mathematical Models in Computer Vision, Springer, 375–388, 2006.

[91] A. van Doorn, J. J. Koenderink, and S. Pont, ``Shading, a view from the inside," Seeing Perceiving, vol. 25, nos. 3_4, pp. 303_338, 2012, doi: 10.1163/187847511x590923.

[92] A. J. van Doorn, J. J. Koenderink, and J. Wagemans, ``Light _elds and shape from shading," J. Vis., vol. 11, no. 3, p. 21, 2011, doi: 10.1167/11.3.21.

[93] D. Todorovic, ``How shape from contours affects shape from shading," Vis. Res., vol. 103, pp. 1_10, Oct. 2014.

[94] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, ``Shape-from-shading: A survey, " IEEE Trans. Pattern Anal. Mach. Intell., vol. 21, no. 8, pp. 690_706, Aug. 1999, doi: 10.1109/34.784284.

[95] J.-D. Durou, M. Falcone, and M. Sagona, ``Numerical methods for shape-from-shading: A new survey with benchmarks," Comput. Vis. Image Understand., vol. 109, no. 1, pp. 22_43, Jan. 2008, doi: 10.1016/j.cviu.2007.09.003.

[96] Zhao, Weidong, et al. "A new steel defect detection algorithm based on deep learning. “Computational Intelligence and Neuroscience 2021: 1-13

[97] Song, K. C. (n.d.). Computational Mechanics Group - Northeastern University. http://faculty.neu.edu.cn/songkc/en/zdylm/263265/list/index.htm adresinden Nisan 2022 tarihinde alınmıştır.

ÖZGEÇMİŞ

Ad-Soyad :Feyza SELAMET

ÖĞRENİM DURUMU:

- **Lisans :** 2013, Sakarya Üniversitesi, Bilgisayar ve Bilişim Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü
- **Yükseklisans :** 2015, Sakarya Üniversitesi, Bilgisayar ve Bilişim Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü

MESLEKİ DENEYİM :

- 2015 yılında Sakarya Üniversitesi'nde Araştırma Görevlisi olarak başladığı görevine halen devam etmektedir.

TEZDEN TÜRETİLEN ESERLER:

- Selamet, F., Çakar S., Kotan, M., (2022). Automatic Detection and Classification of Defective Areas on Metal Parts by Using Adaptive Fusion of Faster R-CNN and Shape From Shading. *Institute of Electrical and Electronics Engineers (IEEE)*, 10, 126030-126038., Doi: 10.1109/ACCESS.2022.3224037 (Yayın No: 7984646)
- Çerezci, F., Kazan, S., Öz, M. A, Öz., C, Taşcı, T., Hızal, S., Altay, Ç., (2020). Online Metallic Surface Defect Detection Using Deep Learning. *Emerging Materials Research*, 9(4), 1-8., Doi: 10.21203/rs.3.rs-41274/v1 (Yayın No: 6644927)