



Tactile paving surface detection with deep learning methods

Abdulsamet Aktaş*^{ID}, Buket Doğan^{ID}, Önder Demir^{ID}

Faculty of Technology Department of Computer Engineering, Marmara University, Istanbul, 34730, Turkey

Highlights:

- A new method for detection tactile paving surfaces
- A new data set in the field of tactile paving surface detection
- The use of deep learning techniques in the field of tactile paving surface detection
- High performance values

Keywords:

- Tactile paving
- Deep Learning
- Convolutional neural net
- Visually impaired person
- YOLO

Article Info:

Research Article
Received: 28.11.2019
Accepted: 11.03.2020

DOI:

10.17341/gazimmfd.652101

Correspondence:

Author: Abdulsamet Aktaş
e-mail:
abdulsamet.aktas@marmara.edu.tr
phone:+90 216 777 3809

Graphical/Tabular Abstract

Image processing applications in real-time systems have become a popular topic in recent years. Deep learning methods, which are one of the sub-branches of artificial intelligence, and image processing algorithms are used together to develop applications in many fields such as autonomous automobiles, autonomous unmanned aerial vehicles, assistive robot technologies, assistant technologies for disabled and elderly people. This study aims to detect the tactile paving surfaces with deep learning methods in order to design an assistive technology system that can be used by visually impaired people, autonomous vehicles and robots. Contrary to traditional image processing algorithms, deep learning methods and image processing algorithms are used together in this study. The YOLO-V3 model, which is one of the best methods of object detection, is combined with the DenseNet model to create the YOLOV3Dense model. YOLO-V2, YOLO-V3 and YOLOV3Dense models were trained on the Marmara Tactile Paving Surface (MDPY) dataset, which was created by the researchers and included 4580 images, and their performances were compared with each other on the test dataset.

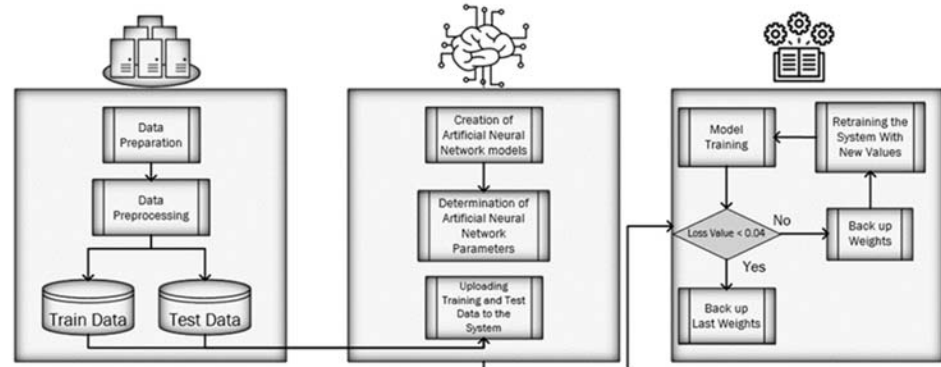


Figure A. System Architecture

Purpose: This study aims to detect the tactile paving surfaces with deep learning methods in order to design an assistive technology system that can be used by visually impaired people, autonomous vehicles and robots.

Theory and Methods:

The YOLO-V3 model, which is one of the best methods of object detection, is combined with the DenseNet model to create the YOLOV3Dense model. YOLO-V2, YOLO-V3 and YOLOV3Dense models were trained on the Marmara Tactile Paving Surface (MDPY) dataset, which was created by the researchers and included 4580 images, and their performances were compared with each other on the test dataset.

Results:

It was observed that YOLOV3-Dense model is better than other models in detecting tactile paving surface with 89% F1-score, 92% mean average Precision(mAP) and 81% IoU values. At 60 frames per second, the YOLOV3-Dense model can also be used in real-time systems.

Conclusion:

Thanks to YOLOV3Dense model, the features in the images can be used in all layers without any loss. In this way, small tactile paving surfaces can be detected with mAP value. This study shows that tactile paving surfaces can be detected in real time with deep learning methods.



Derin öğrenme yöntemleri ile dokunsal parke yüzeyi tespiti

Abdulsamet Aktaş*^{ID}, Buket Doğan^{ID}, Önder Demir^{ID}

Marmara Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü, 34722, Kadıköy, İstanbul, Türkiye

Ö N E Ç İ K A N L A R

- Dokunsal parke yüzeylerini tespit etmek için yeni bir yöntem
- Dokunsal parke yüzeyi tespiti alanında yeni bir veri seti
- Dokunsal parke yüzeyi tespiti alanında derin öğrenme tekniklerinin kullanılması

Makale Bilgileri

Araştırma Makalesi

Geliş: 28.11.2019

Kabul: 11.03.2020

DOI:

10.17341/gazimmfd.652101

Anahtar Kelimeler:

Dokunsal parke,
derin öğrenme,
evrimsel sinir ağları,
görme engelli kişiler,
YOLO

ÖZET

Gerçek zamanlı çalışan sistemlerde görüntü işleme uygulamaları yapmak son yıllarda oldukça popüler bir konu haline gelmiştir. Yapay zekâ alanının alt dallarından biri olan derin öğrenme yöntemleri ve görüntülerden nesne tespiti yapma alanında kullanılan görüntü işleme algoritmaları birlikte kullanılabilir. Bu sayede otonom otomobiller, otonom insansız hava araçları, yardımcı robot teknolojileri, engelli ve yaşlı bireyler için asistan teknolojileri gibi birçok alanda uygulamalar geliştirilmektedir. Yapılan çalışmada, görme engelli bireyler, otonom araçlar ve robotlar tarafından kullanılabilir yardımcı bir teknoloji sistemi tasarlamak için dokunsal parke yüzeylerinin derin öğrenme yöntemleriyle tespit edilmesi gerçekleştirilmiştir. Geleneksel görüntü işleme algoritmalarının aksine bu çalışmada derin öğrenme yöntemleri ile görüntü işleme algoritmaları birlikte kullanılmıştır. Nesne tespit etme yöntemleri içinde en iyi yöntemlerden biri olan You Only Look Once-V3 (YOLO-V3) modeli DenseNet modeli ile birleştirilerek YOLOV3-Dense modeli oluşturulmuştur. YOLO-V2, YOLO-V3 ve YOLOV3Dense modelleri tarafımızca oluşturulmuş olan ve içerisinde 4580 etiketli görsel bulunan Marmara Dokunsal Parke Yüzeyi (MDPY) veri seti üzerinde ayrı ayrı eğitildikten sonra performansları test veri seti üzerinde birbirleri ile karşılaştırılmıştır. %89 F1-skor, %92 ortalama hassasiyet ve %81 IoU değerleri ile YOLOV3-Dense modelinin dokunsal parke yüzeyi tespit etmede diğer modellerden daha iyi olduğu gözlemlenmiştir.

Tactile paving surface detection with deep learning methods

H I G H L I G H T S

- A new method for detection tactile paving surfaces
- A new data set in the field of tactile paving surface detection
- The use of deep learning techniques in the field of tactile paving surface detection

Article Info

Research Article

Received: 28.11.2019

Accepted: 11.03.2020

DOI:

10.17341/gazimmfd.652101

Keywords:

Tactile paving,
deep learning,
convolutional neural
networks,
visually impaired persons,
YOLO

ABSTRACT

Image processing applications in real-time systems have become a popular topic in recent years. Deep learning methods, one of the sub-branches of artificial intelligence, and image processing algorithms used in the field of object detection from images can be used together. In this way, applications are developed in many areas such as autonomous cars, autonomous unmanned aerial vehicles, assist robot technologies, assistant technologies for disabled and elderly individuals. This study aims to detect the tactile paving surfaces with deep learning methods in order to design an assistive technology system that can be used by visually impaired individuals, autonomous vehicles and robots. Contrary to traditional image processing algorithms, deep learning methods and image processing algorithms are used together in this study. The YOLO-V3 model, which is one of the best methods of object detection, is combined with the DenseNet model to create the YOLOV3-Dense model. YOLO-V2, YOLO-V3 and YOLOV3Dense models were trained on the Marmara Tactile Paving Surface (MDPY) dataset, which was created by the researchers and included 4580 images and their performances were compared with each other on the test dataset. It was observed that YOLOV3-Dense model is better than other models in detecting tactile paving surface with 89% F1-score, 92% mean average Precision (mAP) and 81% IoU values.

1. GİRİŞ (INTRODUCTION)

Dünya genelinde yaklaşık olarak 2.2 milyar insan görme bozukluğu yaşamaktadır. Bu insanların yaklaşık 36 milyonu görme yetisini tamamen kaybetmiştir[1]. Görme engeline sahip bireyler daha rahat hareket edebilmek için rehber köpekler ve baston gibi yardımcı unsurlar kullanmaktadırlar. Bu yardımcı unsurların yanı sıra gerek dış mekânlarda gerekse iç mekânlarda daha rahat hareket edebilmelerini sağlamak amacıyla zemine döşenmiş kabartmalı yüzeyler bulunmaktadır. Bu yüzeylerin şekilleri, renkleri ve isimleri ülkelere göre değişebilmektedir[2]. Yaygın olarak dokunsal parke yüzeyi (DPY)olarak adlandırılan ve sarı renkte olan bu yüzeyler görme engelli bireylerin hareket etmelerinde kolaylık sağlasalar da kesin bir çözüm olamamışlardır. Zamanla bu yüzeylerin eskimesi, yüzeylerin olması gerektiği gibi konumlandırılmaması, hareketi engelleyecek şekilde yüzeylerin üzerinde veya etrafında engellerin olması gibi problemler yaşanabilmektedir. Bu problemleri görme engelli kişilerin fark edebilmesi ve problemi çözebilmesi her zaman mümkün olamamaktadır. Problemleri çözmek amacıyla günümüz teknolojileri kullanılarak görme engelli bireylere yardımcı olabilecek uygulamalar geliştirilmektedir. Görme engelli bireylerin yanı sıra otonom bir şekilde hareket edebilen robotlar ve araçlar da bu yardımcı uygulamalara ihtiyaç duymaktadır. Geliştirilen yardımcı uygulamalarda görüntü işleme algoritmaları [3-5], ultrasonik sensörler [6] ve ışık kodlama teknolojisi [7]gibi yöntemler kullanılmıştır. Görüntü işleme algoritmaları ile DPY'nin renk ve şekil özellikleri kullanılarak, ultrasonik sensörler ve ışık kodlama teknolojisi ile yardımcı elemanların birbiri ile haberleşmesi ve bilgi aktarımı sağlanarak DPY tespiti yapılmıştır.

Son yıllarda yapılan bilgi depolama teknolojilerindeki gelişmelerin etkisiyle işlenmemiş ham veri miktarı oldukça artmıştır. Bu artış ile birlikte paralel hesaplama yöntemleri ve yüksek işlem gücü, makine öğrenmesi ve görüntü işleme gibi alanlar için vazgeçilmez bir ihtiyaç haline gelmiştir. Bütün bu gelişmeler ile birlikte işlem gücü yüksek ekran kartları üretilebilir hale gelmiştir. Bu gelişim sayesinde ekran kartları üzerinde paralel hesaplama imkânı veren yeni kütüphaneler ile aritmetik işlemler yapmak oldukça kolay ve kullanışlı bir hale gelmiştir. Çok fazla maliyet, zaman ve yüksek işlem gücü gerektiren hesaplama işlemlerini daha kısa sürede ve kullanışlı bir şekilde yapmak mümkün olmuştur. Oluşturulan bu hesaplama ve işlem gücü, yapay sinir ağlarında derin ağ olarak adlandırılan çok katmanlı ağlar ile çalışmayı mümkün kılmıştır. Derin ağlar sayesinde görüntü işleme alanında oldukça büyük ilerlemeler kaydedilmiştir. Çok katmanlı sinir ağları görüntülerde bulunan nesnelere sınıflandırılması ve tespiti ile ilgili yapılan yarışmalarda normal makine öğrenmesi yöntemlerinden çok daha iyi derecelere alanının en iyisi olduğunu kanıtlamıştır[8]. Alex [9] tarafından 2012 ImageNet [10] sınıflandırma yarışmasında çok katmanlı evrimsel sinir ağları kullanılmış ve %16,42 hata oranı ile derin öğrenme yöntemlerinin diğer sığ ağlara göre daha iyi

bir performans gösterdiği kanıtlanmıştır. Derin öğrenme yöntemleri kullanılarak oluşturulan ağlardan alınan başarı giderek artmıştır. ImageNet verilerini sınıflandırmada en iyi insan hata seviyesi %5.1'dir [11]. PReLU-Net %4,94 hata oranına ulaşarak insan doğruluk seviyesini aşan ilk yapay sinir ağı haline gelmiştir[12]. Bu değer ilerleyen yıllarda daha küçülerek 2019 yılında EfficientNet [13] ile %2.9 olmuştur.

Bu çalışmada hem görme bozukluğuna sahip bireylere yardımcı teknolojilerde kullanılabilecek hem de otonom bir şekilde hareket edebilen araçlarda kullanılabilecek görüntülerde, videolarda ve gerçek zamanlı çalışan sistemlerde dokunsal parke yüzeylerini tespit edebilen bir model gerçekleştirilmiştir. Derin öğrenme yöntemlerinin görüntü işleme alanındaki uygulanabilirliği ve uygunluğu göz önüne alınarak çalışmada temel olarak derin öğrenme yöntemleri ve görüntü işleme algoritmaları kullanılmıştır. Derin öğrenmeye dayalı nesne tespiti söz konusu olduğunda karşılaşılan üç temel nesne tanıma metodu vardır. Bunlar; Region-Convolutional Neural Network (R-CNN), Single Shot MultiBox Detector (SSD) ve You Only Look Once(YOLO) dur[14-16]. Yapılan çalışmada YOLO metodu ve evrimsel sinir ağları [17] kullanılmıştır. Veri seti olarak bu çalışma kapsamında araştırmacılar tarafından oluşturulan ve başka araştırmacıların erişimine sunulan 4580 etiketli görüntüden oluşan MDPY(Marmara Dokunsal Parke Yüzeyi) veri seti kullanılmıştır. Veri seti üzerinde yapılan deneylerde %89 F1-skor, %92 ortalama hassasiyet ve %81 IoU değeri ile dokunsal parke yüzeyleri tespiti gerçekleştirilmiştir. Yapılan katkılar:

- Gerçekleştirilen çalışmada; bu alanda yeni ve büyük veri seti oluşturulmuş ve bu veri seti üzerinde YOLO yöntemi kullanılarak, belirli bir kamera açısına bağlı kalmadan dokunsal parke yüzeyleri tespiti gerçekleştirilmiştir.
- Yüksek performans değerlerine sahip yeni bir model geliştirilmiştir.
- Derin öğrenme yöntemleri, dokunsal parke yüzeyi tespiti etme alanında kullanılmıştır.

Çalışma üç bölümden oluşmaktadır. Birinci bölümde dokunsal parke yüzeylerinin tespitine ilişkin çalışmalar sunulmuştur. İkinci bölüm ise çalışmanın yöntemini ve ön hazırlık sürecini içermektedir. Üçüncü bölümde ise sonuçlar, tartışma ve yapılması planlanan projeler sunulmaktadır.

1.1. İlgili Çalışmalar(Related Works)

Görme engelli bireylerin seyahat ederken engelleri algılaması ve kaçınması zordur. Teknolojinin gelişmesiyle birlikte engelleri önceden algılayabilen ve engelli bireyi yönlendiren sistemler tasarlanmıştır. Bu başlık altında, görme engelli bireyler için yapılan yardımcı sistemler ve evrimsel sinir ağları kullanılarak görüntülerde nesne tespiti yapmayı gerçekleştiren çalışmalar hakkında literatürde bulunan önemli çalışmalara değinilmektedir.

Marcelo C. Ghilardi[4], görme engelli bireylere yardımcı olmak amacıyla gerçek zamanlı olarak dokunsal parke yüzeylerinin tespitini gerçekleştirmiştir. Yapılan çalışmada Canny, Laplace ve Sobel gibi kenar belirleme algoritmalarını ve karar ağaçlarını birleştirerek yeni bir yaklaşım elde etmişlerdir. Araştırmacılar tarafından oluşturulan veri setinde 521 adet görüntü bulunmaktadır. Görüntüler, bir metre yükseklikte ve yer ile 45 derece açı yapılacak şekilde kişinin bel bölgesine bağlanmış olan akıllı telefon ile elde edilmiştir. 320 adet görüntüde dokunsal parke yüzeyi mevcutken 201 adet görüntüde dokunsal parke yüzeyi bulunmamaktadır. Yapılan çalışmada sistem yaklaşık olarak 16,27 fps. ile çalışmaktadır. Yapılan sistemin doğruluk oranı yaklaşık olarak %88,48 oranına sahiptir.

Takuya Asami ve Kengo Ohnishi [3] benzer bir çalışmada, görme engelli bireylerin yaya geçidini bulmasına yardımcı olan sistemi gerçekleştirmişlerdir. Sistem gözlüğe entegre edilmiş bir usb kamera, bileğe takılan bir adet titreşimli bileklik, görme engelli bastonuna entegre edilmiş görme engelli alfabesinde uyarı veren bir bildirim sistemi ve usb kameradan gelen verileri işlemek için kullanılan bir adet bilgisayar olmak üzere dört parçadan oluşmaktadır. Bu çalışmada, usb kameradan alınan görüntülerde normalize edilmiş çarpaz korelasyon yöntemi ile yaya geçidi algılanarak, kullanıcıya yönlendirme yapan bir sistem geliştirilmiştir. Yaya geçidi algılandığında kullanıcının bilekliği titreterek kullanıcının yaya geçidinin önünde durması sağlanmıştır. Geliştirilen sistem %86.7 doğrulukla yaya geçitlerini tespit etmektedir.

Daniel Centeno Einloft [18], DPY tespiti için yapılan çalışmaya [4] benzer olarak alışveriş merkezi, metro ve otobüs terminalleri gibi iç mekânlarda kullanılabilen ve DPY tespiti yapabilen sistemi yapmayı amaçlamıştır. Yapılan çalışmada görüntü sınıflandırmada kullanılan gri seviyeli eş-oluşum matrisi metodunu ve denetimli öğrenme yöntemlerinden olan destek vektör makineleri kullanılmıştır. İlk sonuçlar umut vadetmesine rağmen çalışma doğruluğu ile ilgili herhangi bir sayısal değer verilmemiştir.

Xu Jie [19] tarafından yapılan çalışmada görüntü işleme algoritmaları kullanılarak dokunsal parke yüzeylerinin tespiti gerçekleştirilmiştir. Yapılan sistemde çok renkli görüntü segmentasyonu ve kirsch kenar belirleme algoritması kullanılmıştır. Hough dönüşüm oryantasyonu kullanılarak dokunsal parke yüzeyi şerit olarak elde edilmiştir. 2 fps çalışma hızına sahip olan ve portatif olarak kullanılabilen sistemin, çalışma doğruluğu ile ilgili herhangi bir sayısal değere değinilmemiştir.

A.M. Kassim [20] görme engelli bireylerin dokunsal parke yüzeylerini tespit edebilmeleri için yeni bir baston tasarlamıştır. Tasarlanan bastona eklenen web kamera ile görüntü alınmakta ve arduino Xbee verici ile, işlenmek üzere bilgisayara gönderilmektedir. Bilgisayar ortamında MATLAB programı ile görüntü farklı filtrelerden geçirilerek işlenmektedir. İşlenen görüntülerden elde edilen "A" alan değeri $m = 4\pi A/r^2$ formülünde kullanılarak m metrik değeri

hesaplanmaktadır. Hesaplanan m değerine göre alınan görüntünün dairesel veya doğrusal olduğuna karar verilmektedir. $0.15 < m < 0.30$ aralığında alınan görüntünün doğrusal bir görüntü olduğu yani kullanıcının ilerleyebileceği, $0.85 < m < 1.0$ aralığında alınan görüntünün dairesel yani uyarı parke yüzeyi olduğu kullanıcının durması gerektiği anlaşılmaktadır. Hesaplanan m metrik değerine göre alınan sonuçlar arduino Xbee verici ile tekrar bastona gönderilmektedir. Gelen verilere göre sisteme entegre olan kulaklık ile kullanıcı bilgilendirilmektedir. Gerçekleştirilen sistem belirli bir donanım ekipmanına sahip olmayı gerektirdiği için pratik bir kullanıma sahip değildir.

Jiaquan Shen [21] tarafından yapılan çalışmada derin evrişimli ağlarda kullanılan hiper özellik haritası kullanılarak havadan çekilmiş görüntülerdeki araçlar tespit edilmiştir. Yapılan algoritmanın temelini Faster R-CNN[22] algoritması oluşturmaktadır. Bu algoritmada temel olarak doğal sahnelerdeki nesnelere tespit edilmektedir. Havadan çekilmiş görüntülerdeki küçük nesnelere tespit etmek için uygun değildir. Jiaquan Shen ve arkadaşları, havadan çekilmiş görüntülerdeki küçük araçları doğru ve etkili bir şekilde tespit etmek için Eltwise modeli ve Concat modelini hiperaktif özellik harita ağı ile birleştirerek yeni bir yaklaşım sunmaktadır. Tasarlanan algoritmanın tespit etme performansını Munich veri seti ve kendi oluşturdukları veri seti üzerinde değerlendirmişlerdir. Yapılan sistem 82.2% duyarlılık (Recall) ve 90.2% doğruluk oranına sahiptir.

Literatür taraması sonucunda daha önce yapılan sistemlerin günlük hayatta kullanılabilirliğinin oldukça zor olduğu anlaşılmaktadır. Bu zorluklar; düşük fps değerine sahip sistemlerin[3,4,19] gerçek zamanlı olarak kullanılamaması, hem iç mekânlarda hem de dış mekânlarda yalnızca bir sistem [4,18] ile dokunsal parke yüzeylerinin tespit edilememesi, alışveriş merkezi, metro, vb. gibi iç mekânlarda için ayrı, yürüyüş yolu, cadde gibi dış mekânlarda için ayrı bir sistemin oluşturulması, [3,20] kullanıcının sistemi kullanmak için sırt çantasında bilgisayar, baston, bileklik gibi ekstra donanımlar kullanmak zorunda olması, [18] veri azlığı nedeni ile çalışmaların tamamlanamaması gibi sorunlardan oluşmaktadır. Literatürde bulunan çalışmaların aksine bu çalışmada yapılan sistem ile saniyede 60 kare ile işlem yaparak gerçek zamanlı olarak çalışılmış ve baston, bileklik, bilgisayar gibi ekstra donanımlara ihtiyaç duymadan hem iç mekânlarda hem de dış mekânlarda dokunsal parke yüzeylerinin tespiti yapılmıştır. Araştırmacılar tarafından oluşturulan ve üzerinde çalışılan veri seti sayesinde veri azlığı nedeniyle oluşan problemlerin önüne geçilmiştir.

Yuchuan Du[23] tarafından yapılan çalışmada YOLOV3 yapay sinir ağ mimarisi ile karayollarında bulunan asfalt kaplamada oluşmuş bozulmalar tespit edilmiş ve sınıflandırılmıştır. Yapılan deneysel çalışmalar yaklaşık 200km karayolunu kaplayabilecek kapasiteye sahip 45.788 adet asfalt kaplı görüntü içeren veri seti üzerinde yapılmıştır. Veri setine ait özellik çıkarım işlemleri için Darknet-53 mimarisi kullanılmıştır. YOLOV3 ağı ile eğitilen sistem %73.64 doğruluk yüzdesine ve yaklaşık 29 fps değerine

sahiptir. Bu hız değeri ile Faster R-CNN'den yaklaşık olarak 9 kat daha hızlı bir şekilde işlem yapmaktadır. Aynı zamanda yüksek doğruluk değerine sahip YOLO tabanlı bu çalışma sayesinde asfalt kaplamada bulunan bozuklukları bulmak için manuel özellik çıkarım işlemine ihtiyaç duyulmamaktadır.

Wang Juan[24] tarafından yapılan çalışmada YOLOV3 yapay sinir ağ mimarisi ile yumurta yetiştiricilerine ait kafeslerde bulunan hayvanların davranışları incelenmiştir. Hayvanlara ait toplam altı davranış günün sabah ve öğlen saatlerinde incelenerek bu davranışların üretime olumlu ve olumsuz etkileri tartışılmıştır. Davranışları tespit etmede YOLOV3 yapay sinir ağ mimari kullanılmıştır. Araştırmacılar tarafından 30 gün boyunca değişik kafeslerden elde edilen toplam 10.230 görüntü veri seti olarak kullanılmıştır. Veri seti üzerinde yapılan deneysel çalışmalar sonucunda %94.72 çiftleşme, %94.57 ayakta durmak, %93.10 beslenme, %92.02 kanat çırpma, %88.67 dövüşme ve %86.88 su içme hareketlerine ait ortalama hassasiyet değerleri elde edilmiştir. Yapılan bu çalışma ile kümes hayvanlarının davranışları ve yumurta üretimi arasında bir ilişki kurularak üreticiye teknik destek verilmesi sağlanmıştır.

Aydın Kaya[25] tarafından yapılan çalışmada akciğer nodül özelliklerinin tahmininde çeşitli sınıflandırma stratejilerinin incelenmesi yapılmıştır. Sınıflama çalışmalarında kullanılan öznetelik çıkarım sürecinde geleneksel öznetelikler, derin öznetelikler ve bu iki özneteliğin kombinasyonu kullanılmıştır. Bilgisayarlı tomografi görüntülerinden nodüllere ilişkin derin öznetelik çıkarımında, evrişimsel sinir ağlarından olan Alexnet ile transfer öğrenme yaklaşımı kullanılmıştır. Destek vektör makineleri, rastgele orman, k-en yakın komşu ve Adaboost yöntemleri kullanılarak yapılan çeşitli sınıflandırma işlemleri sonucunda, nodül tespitinde

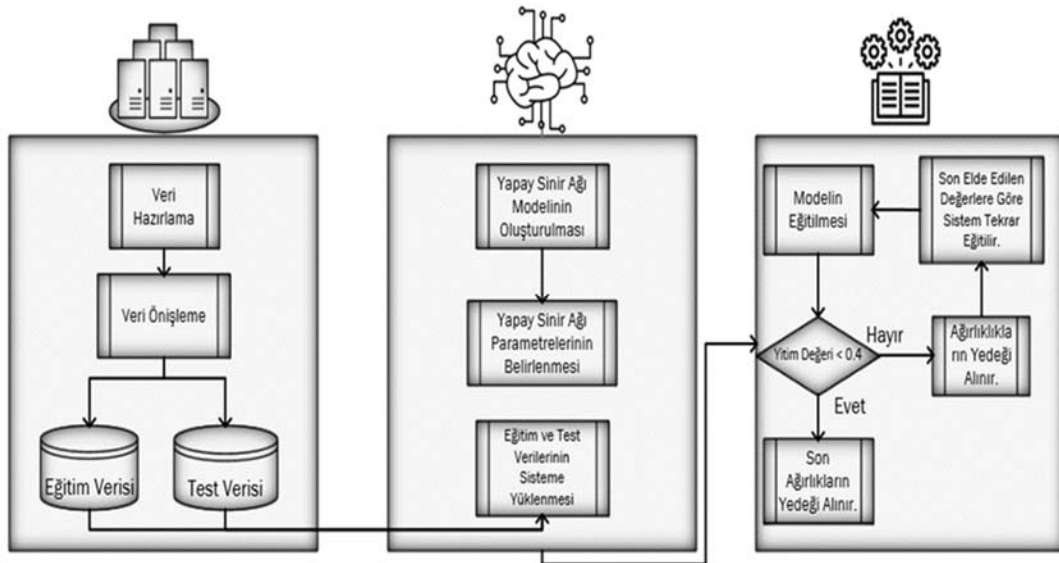
kullanılacak olan özneteliklerin ve sınıflama algoritmalarının sonucu etkilemede önemli bir rol oynadığı tespit edilmiştir.

Oktay Yıldız [26] tarafından yapılan çalışmada derin öğrenme yöntemleri ile dermoskopi görüntülerinden melanom tespiti yapılmıştır. Melanom tanısı için tasarlanan C4Net derin sinir ağ modelinin yanında literatürde bulunan Alexnet, GoogLeNet, ResNet, VGGNet derin öğrenme modelleri ve yapay sinir ağları, en yakın komşu algoritması ve destek vektör makinesi gibi geleneksel makine öğrenme yöntemleri de kullanılarak kapsamlı bir çalışma yapılmıştır. Deneysel çalışmalar International Skin Imaging Collaboration tarafından sunulan 13.000'den fazla dermoskopi görüntüleri içeren veri seti üzerinde gerçekleştirilmiştir. %96.94 doğrulukla C4Net derin sinir ağ modelinin diğer modellerden daha başarılı olduğu sonucu elde edilmiştir.

Emre Dandıl ve Rukiye Polattimur [27] tarafından yapılan çalışmada daha hızlı bölgesel evrişimsel sinir ağları(Faster R-CNN) ile köpek davranışlarının tanınması ve takibi gerçekleştirilmiştir. Araştırmacılar tarafından oluşturulan veri seti üzerinde deneysel çalışmalar yapılmıştır. Yapılan deneysel çalışmalar sonucunda köpeklere ait ağız açma, dil çıkarma, koklama, kulak dikme, kuyruk sallama ve oyun oynama gibi davranışları sırasıyla %100, %99,99, %99,28, %99,99, %95,99, %99,64 doğruluk değerleri ile sınıflandırma işlemi yapılmıştır.

2. ÖNERİLEN YÖNTEM (PROPOSED METHOD)

Yapılan çalışma Şekil 1' de görüldüğü gibi temel olarak 3 adımdan oluşmaktadır. Birinci adımda veri setinin hazırlanması ile ilgili alt işlemler, ikinci adımda derin öğrenme için kullanılacak evrişimsel sinir ağının oluşturulması ve konfigürasyonu, üçüncü adımda sistemin



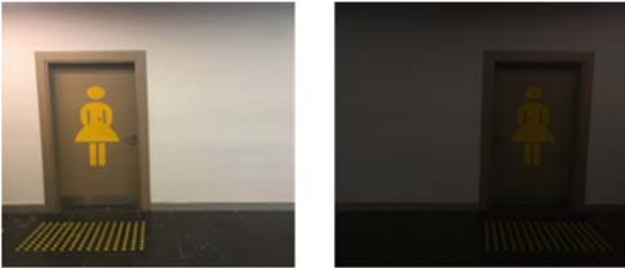
Şekil 1. Sistem Mimarisi (System Architecture)

eğitilmesi ve diğer sistemlerde de kullanılmak üzere evrimsel sinir ağı ağırlıklarının elde edilmesi gerçekleştirilmiştir.

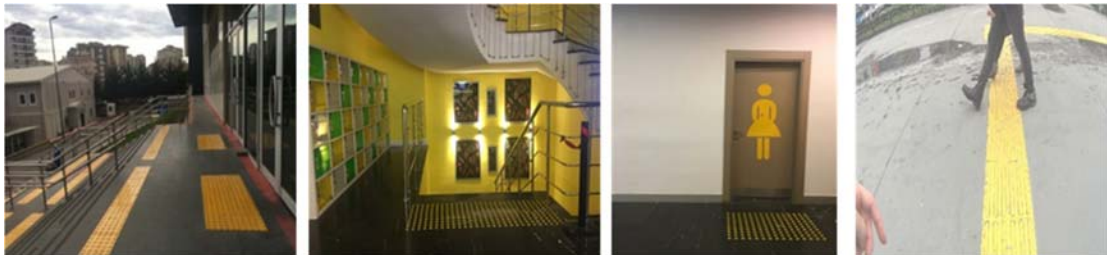
2.1. Veri Hazırlama ve Önışleme (Data preparation and preprocessing)

Yapılan çalışmada arařtırmacılar tarafından oluşturulan veri seti kullanılmıştır. Veri hazırlama ve önışleme evresinde, iç ve dış ortamlarda çekilen görüntü ve video gibi görsel kaynaklar belirli işlemlerden geçirilerek etiketli bir veri seti haline getirilmiştir. Görüntüler İstanbul'un çeşitli semt ve bölgelerinde bulunan kütüphane, metro, parklar, yürüyüş yolları ve caddeler gibi gündelik hayatta çokça kullanılan yerlerden temin edilmiştir. Görüntüler temin edilirken GoPro Hero 7 ve Sony HDR-X3000R-4K aksiyon kamerası kullanılmıştır. Görüntü temini sırasında herhangi bir kişinin özel hayatı ihlal edilmemiştir. Elde edilen görüntülere ait örnek görselleri Şekil 2'de görmek mümkündür.

Elde edilen görüntüleri çok daha hızlı ve sisteme uygun bir şekilde kullanmak için yeniden boyutlandırılma işlemi yapılmış ve 1024x768 boyutuna getirilmiştir. Evrimsel sinir ağları ile danışmanlı öğrenmede, toplanması ve hazırlanması oldukça maliyetli olan çok sayıda etiketli veriye ihtiyaç duyulmaktadır[28]. İhtiyaç duyulan veriyi elde etmek amacıyla python programlama dili ve OpenCV kütüphanesi kullanılarak veri artırma işlemi gerçekleştirilmiştir. Başlangıçta 1200 adet görüntü bulunan veri setine veri artırma işlemleri uygulanarak görüntü sayısı 4580'e çıkarılmıştır. Veri artırma yöntemlerinden aynalama, beyazlatma, döndürme, vb. yöntemler kullanılmıştır[29]. Veri artırma işleminden sonra elde edilen örnek bir görüntüyü Şekil 3'te görmek mümkündür. Görüntülerin büyük bir kısmında birden fazla DPY olması, etiketli veri sayısını daha da artırmaktadır.



Şekil 3. Veri Çoğaltma Örnek Görüntüler
(Data augmentation sample images)



Şekil 2. Veri Seti Örnek Görüntüler (Dataset Sample Images)

Görüntüleri etiketlemek için python programlama dili kullanılarak etiketleme aracı olarak kullanılabilen bir sistem tasarlanmıştır. Sisteme verilen görüntüler etiketlenerek, ilgili görüntünün adı ile bir text dosyası oluşturulmuş ve görüntü içerisinde bulunan DPY'nin koordinatları bu dosya içerisine kaydedilmiştir. Etiketlenmiş örnek görselleri Şekil 4'te görmek mümkündür.

Etiketlenen görüntülerin uzunluk(dH) ve genişliği(dW), DPY nesnesinin görüntünün tam olarak hangi konumunda olduğu bilgisini veren (X_0, Y_0, X_1, Y_1) noktaları ve görüntü üzerinde kaç adet DPY olduğu bilgisi text dosyası içerisine kayıt edilmektedir. Tüm görüntüler üzerinde bu işlem uygulanarak etiketli veri seti oluşturulmuştur. Oluşturulan veri setini, sistemde kullanılan YOLO [16] mimarisine uygun hale getirmek için normalizasyon işlemi yapılmaktadır. Normalizasyon işlemi ile (X_0, Y_0, X_1, Y_1) koordinatları 0 ve 1 aralığına indirgenmiştir. Bu işlem ile görüntü içerisindeki etiketli DPY'nin, merkez nokta koordinatları(X,Y), yükseklik(H) ve genişlik(W) bilgileri elde edilmektedir. (Şekil 5)

$$X = \frac{X_1 + X_0}{2} \times \frac{1}{dW} \quad (1)$$

X : Etiketli verinin orta noktasına ait X koordinatı.

$$Y = \frac{Y_1 + Y_0}{2} \times \frac{1}{dH} \quad (2)$$

Y : Etiketli verinin orta noktasına ait Y koordinatı.

$$W = (X_1 - X_0) \times \frac{1}{dW} \quad (3)$$

W : Etiketli verinin genişliği

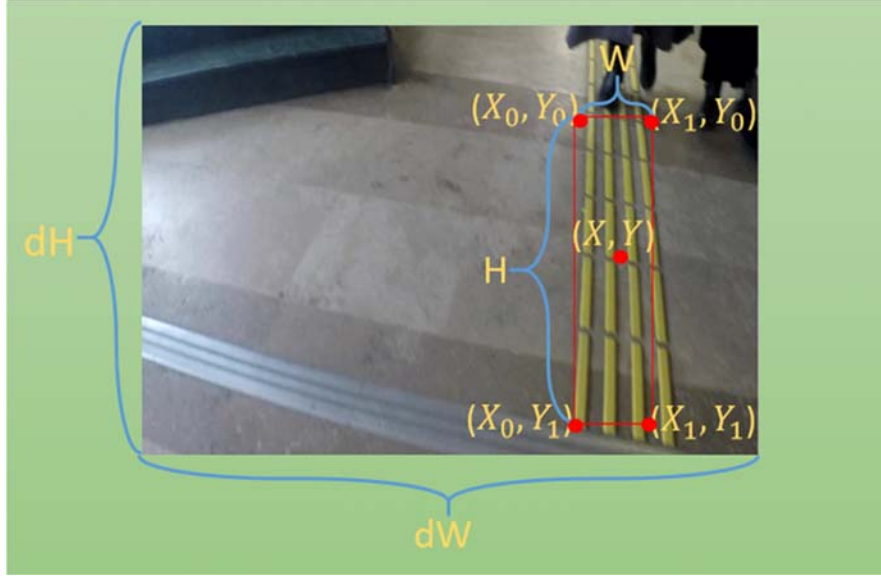
$$H = (Y_1 - Y_0) \times \frac{1}{dH} \quad (4)$$

H : Etiketli verinin yüksekliği

Eş. 1 ve Eş. 2'de yapılan işlemler ile nesne tespitinde kullanılan sınırlayıcı kutuya ait (X_0, Y_0, X_1, Y_1) koordinat değerleri kullanılarak (X,Y) sınırlayıcı kutu merkez nokta koordinat değerleri hesaplanmaktadır. Bu sayede işlem yapılan parametre sayısı düşürülmüş ve nesnenin merkez noktası tespit edilmiştir. Eş. 3 ve 4'te yapılan işlemler ile yine aynı sınırlayıcı kutunun (H,W) yükseklik ve genişlik



Şekil 4. Etiketlenmiş Örnek Görseller (Labeled sample images)



Şekil 5. Etiketlenen nesnenin koordinatları (Coordinates of the labeled object)

değerleri elde edilmiştir. Görüntüye ait dH ve dW (Yükseklik ve Genişlik) değerleri eşitliklerde kullanılırken 1'e bölünerek değerler 0 ve 1 aralığına normalize edilmiştir. Bu işlem sayesinde çok büyük değerler yerine küçük değerler ile işlemler yapılarak sistem çalışma hızı arttırılmıştır. Hesaplanan yeni koordinat, yükseklik ve genişlik değerleri (X, Y, W, H) görüntünün adı ile aynı olan yeni bir text dosyasına kaydedilmektedir. Ön işlemler tamamlandıktan sonra veri seti, %65 eğitim ve %35 test verileri olmak üzere rastgele bir şekilde ikiye ayrılmıştır.

2.2. Sistem Konfigürasyonu (System configuration)

Gerçekleştirilen sistem, YOLOV3-Dense, YOLO-V3 ve YOLO-V2 modelleri olmak üzere üç farklı model kullanılarak çalıştırılabilir. YOLOV3-Dense modeli oluşturulurken YOLO-V3 modeli ve yoğun bağlantılı sinir ağı olan DenseNet ağından yararlanılmıştır. Sistemin konfigürasyonu, kullanılan model ve ağlar, önerilen yapay sinir ağı modeli ve algoritma alt başlıklarda anlatılmıştır.

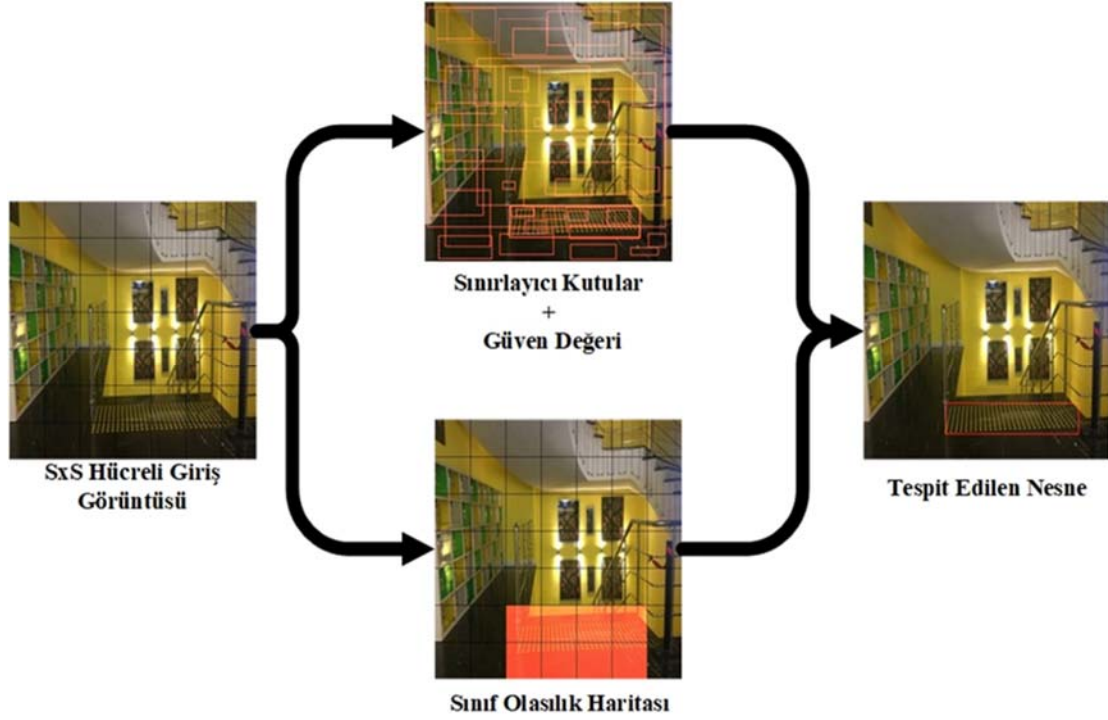
2.2.1. YOLO modeli (YOLO model)

YOLO-V3 [30] ağı YOLO[16] ve YOLO-V2 [31] ağları kullanılarak geliştirilmiştir. Çalışma prensibi olarak YOLO ağı, tespit etme problemini regresyon problemine dönüştürmektedir. YOLO geleneksel yöntemlerden olan Hızlı R-CNN[14] in aksine nesne tespit etmede nesnenin

bulunabileceği muhtemel alanlara ihtiyaç duymamaktadır. Regresyon sayesinde görüntüde bulunan her sınıf için sınırlayıcı kutu (Bounding Box) koordinatlarını ve olasılıklarını oluşturmaktadır. Her bir sınıf için görüntüde ayrı ayrı işlem yapmak yerine görüntüye yalnızca bir kere bakılır ve tüm sınıflara ait sınırlayıcı kutu koordinatları ve olasılıklar oluşturulur.

Bu sayede görüntü üzerinde hangi nesnelerin olduğunu ve nesnelerin görüntünün tam olarak neresinde olduğunu öğrenmek oldukça hızlı bir şekilde gerçekleştirilmektedir. Bu durum geleneksel nesne tespiti yapan sistemlerle karşılaştırıldığında tespit süresini oldukça kısaltmaktadır.

YOLO tespit modeli, eğitim setinde bulunan her bir görüntüyü Şekil 6'da gösterildiği gibi $S \times S$ ($S=7$) kare hücrelere (grid) bölmektedir. Tespit edilmesi istenen nesnelere herhangi birinin merkez konumu bölünen hücrelerden herhangi birinin içerisinde ise, merkezin bulunduğu hücre o nesneyi tespit etmekte sorumludur. Her bir hücre B sınırlayıcı kutularını üretmekte ve bu sınırlayıcı kutular için güven skorunu tahmin etmektedir. Güven skoru ile modelin ürettiği sınırlayıcı kutunun bir nesne içerdiğinden ne kadar emin olduğu ve üretilen sınırlayıcı kutunun üretilme olasılığının ne kadar doğru olduğu bilgileri yansıtılmaktadır. Güven skoru = $P(\text{Object}) * IOU_{pred}^{truth}$, $P(\text{Object}) \in (0,1)$ olarak hesaplanmaktadır. Bölünen hücreler içerisinde herhangi bir nesne yok ise o hücrelerin



Şekil 6. YOLO Tespit Şeması (Yolo detection scheme)

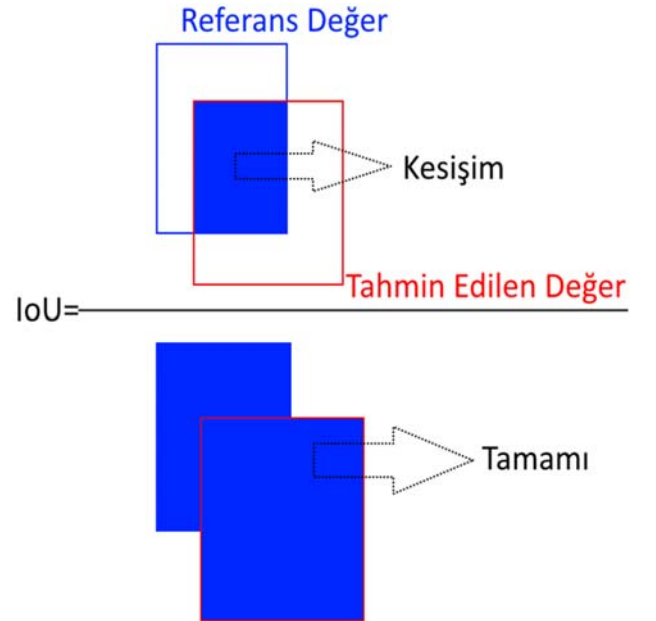
güven skoru 0, var ise 1 olmaktadır. Her bir hücre aynı zamanda Eş. 5'te görüldüğü gibi C sınıf koşullu olasılıklarını (conditional class probabilities) $P(Class_i | Object)$, tahmin etmektedir. Bu olasılıklar, nesne içeren hücreler ile bağlantılıdır. Üretilen sınırlayıcı kutu sayısına B bakılmaksızın her bir hücre başına sınıf olasılıklarının yalnızca biri tahmin edilmektedir. Test sırasında koşullu sınıf olasılıkları ve bireysel güven skoru tahminleri çarpılarak her bir kutu için sınıf bazında güven skorları hesaplanmaktadır[16]. Eş. 5'te kullanılan IOU_{pred}^{truth} değeri iki sınırlayıcı kutu arasındaki örtüşmenin ölçümü için kullanılan bir metrik sistemdir. Tahmin edilen değer ile olması gereken referans değeri arasında kalan bölgenin tüm bölgeye bölümü ile elde edilir. Şekil 7'de IoU'nun çalışma mantığını görmek mümkündür.

$$\frac{P(Class_i | Object) * P(Object) * IOU_{pred}^{truth}}{P(Object) * IOU_{pred}^{truth}} = P(Class_i) * IOU_{pred}^{truth} \quad (5)$$

2.2.2. Yoğun bağlantılı sinir ağları (Densely connected neural network)

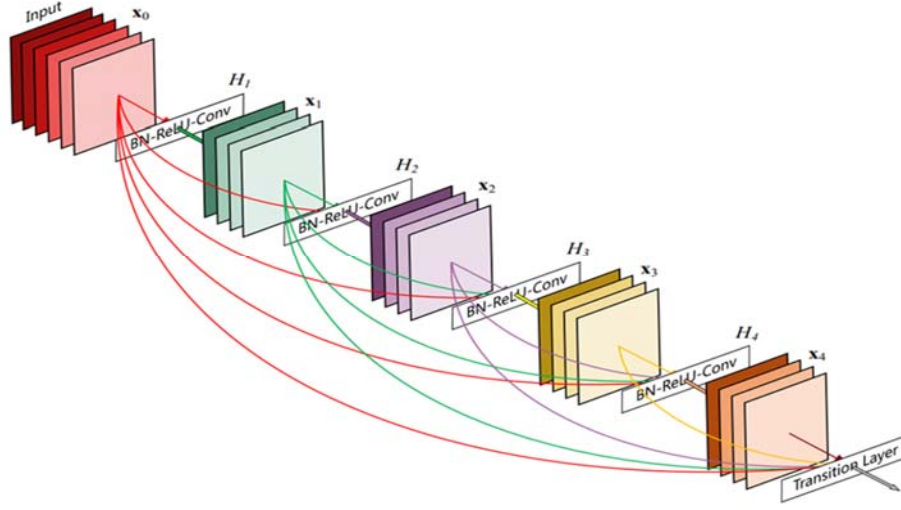
Sinir ağları eğilirken konvolüsyon ve alt-örnekleme işlemleri nedeniyle özellik haritalarında azalma meydana gelmektedir. Aynı zamanda katmanlar arası geçişlerde görüntü özelliğinde kayıplar olmaktadır. Görüntü özellik bilgilerinin daha etkili kullanılması için Huang [32] tarafından DenseNet sistemi geliştirilmiştir. Yapılan sistemde her bir katman diğer katmanlara ileri beslemeli bir şekilde bağlanmıştır. Bu sayede herhangi bir l katmanı kendinden önceki bütün katmanlara ait özellik bilgisine

erişebilmektedir. Yoğun bağlantılı sinir ağları genel ağ mimarisini Şekil 8'de görmek mümkündür.



Şekil 7. Birlik üzerinde kesişim (Intersection over Union)

$[X_1, X_2, X_3, \dots, X_{l-1}]$ Katmanları, özellik bilgilerinin birleşimidir. H_l , özellik bilgilerini işlemek için kullanılan transfer fonksiyonudur. Bu sayede özelliklerin ağa yayılım oranı artmakta, tekrar kullanımı kolaylaşmakta ve parametre



Şekil 8. Yoğun bağlantılı sinir ağı[32] (Densely connected neural network)

sayısında büyük ölçüde azalma olmaktadır. Yoğun bağlantılı sinir ağı eşitliğini Eş. 6'da görmek mümkündür.

$$X_l = H_l[X_0, X_1, X_2, \dots, X_{l-1}] \quad (6)$$

2.2.3. Önerilen algoritma (Proposed algorithm)

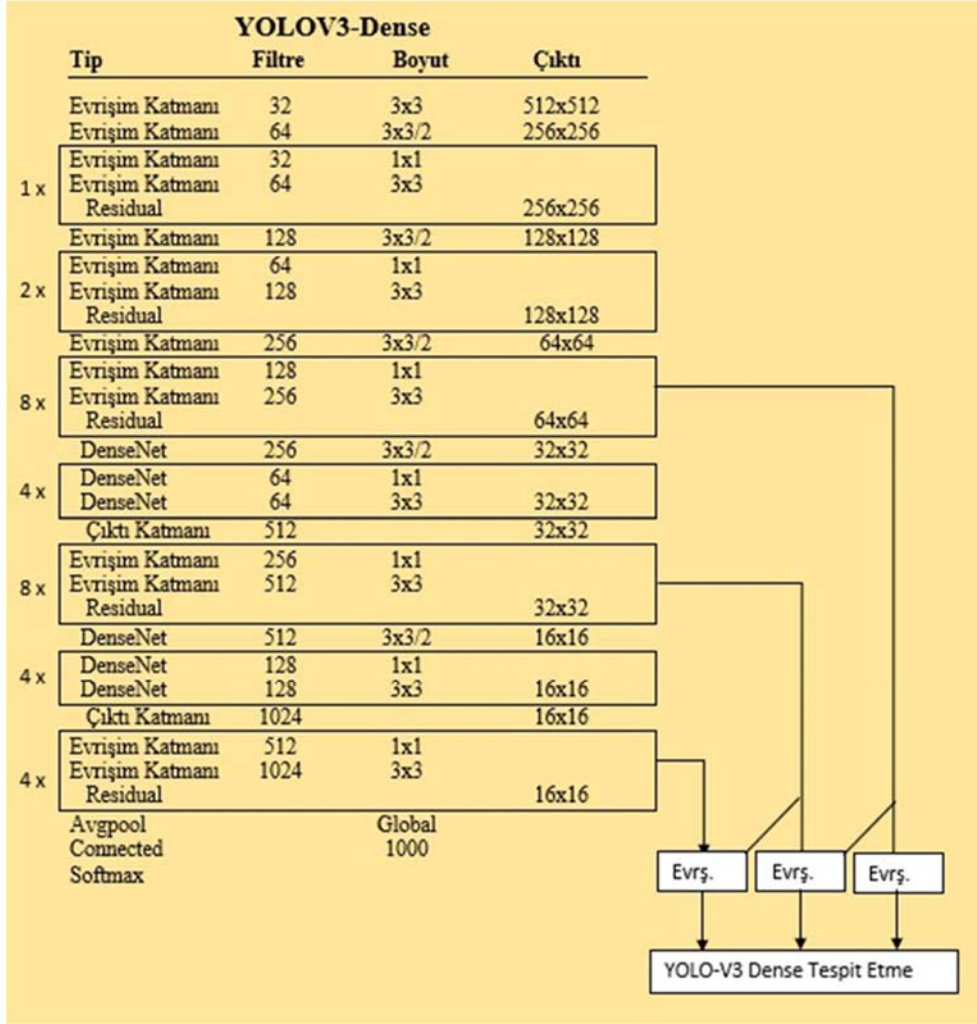
Yapılan çalışmada kullanılan YOLOV3-Dense ağ parametreleri Şekil 9'da gösterildiği gibidir. Yüksek çözünürlüklü görüntüleri daha iyi işlemek için öncelikle görüntüler 512x512 olarak yeniden boyutlandırılmıştır. İlerleyen katmanlarda görüntüler 32x32 ve 16x16 alt örnekleme filtrelerinden geçirilerek özneteliklerin çıkarılması sağlanmıştır. Yapılan çalışmada öznetelik bilgilerini işlemek için kullanılan transfer fonksiyonu H_l 'de BN-LReLU-Conv(1X1) fonksiyonunu kullanılmaktadır. BN (Batch Normalization) yığın normalizasyonu, LReLU(Leaky Rectified Linear Units) ve Conv, konvolüsyon özelliklerinin birleşimidir. H_l fonksiyonu ile doğrusal olmayan $[X_0, X_1, X_2, \dots, X_{l-1}]$ katmanları üretilmektedir. Bu katmanların her biri 32x32 çözünürlük değerine sahip 64 özellik katmanından oluşmaktadır. H_1 ile x_0 üzerine BN-LReLU-Conv(1X1) fonksiyonu uygulanmaktadır. Çıkan sonuç üzerine BN-LReLU-Conv(3X3) fonksiyonu uygulanarak görüntünün öznetelik haritası çıkarılmaktadır. Yukarıda H_1 tarafından yapılan işlemler, H_2 tarafından x_0 ve x_1 üzerinde uygulanarak x_2 sonucu elde edilmektedir. Elde edilen $[x_0, x_1, x_2]$ sonuçları H_3 fonksiyonuna girdi olarak verilerek $[x_0, x_1, x_2, x_3]$ sonucu elde edilmektedir. Yapılan bu işlem Şekil 9'da görüldüğü gibi 16x16x1024 öznetelik katmanına kadar devam ettirilerek ilerlenmektedir. Bu sayede görüntü öznetelikleri bir katmandan diğerine aktararak düşük çözünürlüğe sahip görüntüler üzerinde çalışan katmanlarda da görüntünün bozulmadan önce alınan öznetelikleri kullanılmıştır. Öznetelik kaybı azaltılmış, öznetelik kullanım oranı artırılmış, görüntüye ait üç farklı boyutta tespit işlemi yapılmış ve daha iyi bir sonuç alınması sağlanmıştır.

3. DENEYSEL ÇALIŞMALAR (EXPERIMENTAL STUDIES)

Çalışmada tasarlanan yapay sinir ağı modelinin eğitim ve test süreçleri NVIDIA GeForce GTX1080 Ti 11GB ekran kartı üzerinde gerçekleştirilmiştir. Sistemin düzgün bir şekilde çalışması için gerekli olan momentum, başlangıç öğrenme oranı(initial learning rate) ve diğer parametreler orijinal YOLO-V3 modelindeki ile aynı alınmıştır. Model yaklaşık olarak 160 saat eğitilmiştir. Eğitim işlemi 70.000 dönem(epoch) sürmüştür. 70.000 döneme kadar yapılmasının nedeni test hata değerindeki azalmanın devam etmesidir. Bu değerden sonra yapılan eğitim işlemlerinde test hata değerinin giderek arttığı gözlemlenmiştir. Test hata değerinin artması, yapay sinir ağı'nın eğitim verilerini ezberleyerek bu veriler dışında herhangi bir veri geldiğinde nesne tespit işlemini yapamaması anlamına gelmektedir. Ezberlemenin(overfitting) önüne geçmek için erken durdurma (early stopping) yapılarak eğitim işlemine son verilmiştir. Her 10.000 dönemde modelin ağırlıkları yedeklenmiştir. Yedeklenen bu ağırlıklar daha sonra test verileri üzerinde çalıştırılarak ağırlığa ait ortalama hassasiyet (average precision) değerlerinin ortalamaları(mean Average Precision) karşılaştırılmıştır. Eğitilen modelin performansını değerlendirmek için sisteme ait yitim(loss), F1 skoru, ortalama hassasiyet (mAP) ve intersection over union (IoU) değerleri metrik olarak kullanılmıştır.

3.1. F1 skor ve Ortalama Hassasiyet (F1-score and mean Average Precision)

Olasılık modellerinde bulunan tüm sistemler, örnekleri dört ayrı başlık altında toplamaktadır [33]. Bunlar doğru pozitif(TP), yanlış pozitif(FP), doğru negatif(TN) ve yanlış negatif(FN) değerleridir. Bu değerler, kullanıcı tarafından etiketlenmiş verinin etiket değeri ve eğitilmiş sistem tarafından yapılan tahmin sonucunun uyuyup uyumadığına göre oluşmaktadır. Etiketlenmiş verinin pozitif olduğu



Şekil 9. YoloV3-Dense Ağ yapısı (YoloV3-Dense Network structure)

varsayılırsa, tahmin sonucu eğer pozitif ise TP, negatif ise FN değeri oluşmaktadır. Yine aynı şekilde etiketlenmiş verinin negatif olduğu varsayılırsa, tahmin sonucu eğer pozitif ise FP, negatif ise TN değeri oluşmaktadır. Duyarlılık tüm pozitif örnekler için hesaplanmaktadır. Hassaslık ise pozitif sınıflardan gelen tüm örnekler için hesaplanmaktadır. Bu iki metrik değerini oluşturduğu hassaslık/duyarlılık eğrisini tek bir değer ile ifade edebilmek için ortalama hassasiyet metrik değeri oluşturulmuştur. [34] Ortalama hassasiyet değeri hesaplanırken belirli bir IoU eşik değerinin üstündeki sınırlayıcı kutular ile işlem yapılmıştır. Bu sayede mAP metrik değeri hem tespit edilen nesnenin sınıflandırması ile ilgili hem de tespit edilen nesnenin konumu ile ilgili değerler kullanılarak hesaplanmıştır.

F1-skoru: Hassaslık (Precision) ve Duyarlılık (Recall) değerlerinin harmonik ortalaması olarak hesaplanmaktadır. F-1 skorunun formülü Eş. 7'de görülmektedir.

$$F_1 \text{ Skoru} = 2x \frac{P \times R}{P+R} \quad (7)$$

Ortalama Hassasiyet(mAP): Kesinlik, duyarlılık, F1-skoru ve IoU gibi değerlerin tek bir noktadan değerlendirilmesi için tasarlanmış metrik bir sistemdir. mAP değerinin formülü Eş. 8'de görülmektedir.

$$mAP = \int_0^1 P(R) dR \quad (8)$$

3.2. Kayıp Fonksiyonu (Loss Function)

YOLO modelinde her bir hücre(grid) birden fazla sınırlayıcı kutu üretmektedir. Doğru pozitif olan durumlarda yitim değerini hesaplamak için üretilen sınırlayıcı kutular içerisinde en yüksek IoU değerine sahip olan kutu seçilmelidir. Bu sayede gelecek jenerasyonlarda düşük IoU değerine sahip sınırlayıcı kutular elenerek daha yüksek IoU değerine sahip sınırlayıcı kutular üretilmektedir. YOLO algoritmasında yitim değerini hesaplamak için, sınıf, yerini saptama ve güven değerleri hesaplanmaktadır. Hesaplama işleminde tahmin edilen ve gerçek referans değer olan sınırlayıcı kutulara ait bu değerlerin kare hatalarının toplamı alınmaktadır. Bu değerlere; sınıflandırma yitimi, yerini saptama yitimi ve güven yitimi denilmektedir[16].

3.2.1. Sınıflandırma yitimi (Loss of classification)

Sınıflandırma yitiminde nesne tespit edildiğinde her bir sınıf için koşullu sınıf olasılıklarının kare hatası hesaplanmaktadır. Sınıflandırma yitimi formülü Eş. 9'da görülmektedir.

$$\sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in \text{sınıflar}} (P_i(C) - \hat{p}_i(C))^2 \quad (9)$$

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

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

2.1.1. Yerini saptama yitimi (Localization loss)

Yerini saptama yitiminde tahmin edilen sınırlayıcı kutunun konumu ve boyutu ile ilgili yitim değeri hesaplanmaktadır. Bu değer yalnızca nesneyi tespit eden kutu için hesaplanmaktadır.

Yerini saptama yitimi formülü Eş. 10'da görülmektedir.

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \quad (10)$$

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

λ_{coord} : Sınırlayıcı kutu koordinatlarındaki yitim değerinin ağırlığını artırır.

2.1.2. Güven yitimi (Confidence loss)

Güven yitiminde nesne tespit edildiğinde, tespit eden kutunun güven değerini hesaplanmaktadır. Güven yitimi formülü Eş. 11 ve Eş. 12'de görülmektedir.

$$\sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 \quad (11)$$

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

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

Eğer kutuda nesne tespit edilmezse güven yitimi:

$$\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \quad (12)$$

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

1_{ij}^{noobj} : 1_{ij}^{obj} değerinin tümleyenidir.

λ_{noobj} : Arkaplan tespit edildiğinde yitimi azaltmak için kullanılır.

Son yitim değeri; sınıflandırma, yerini saptama ve güven yitimi değerlerinin toplamıdır. Yitim değeri formülü Eş. 13'te görülmektedir.

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 + \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in \text{sınıflar}} (P_i(C) - \hat{p}_i(C))^2 \quad (13)$$

Yapılan çalışmaya ait yitim değerleri çizelgesini Şekil 10'da görmek mümkündür. Dokunsal parke yüzey tespitinde hangi modelin daha iyi olduğunu göstermek için, YOLO-V2, YOLO-V3 ve YOLOV3-Dense modelleri oluşturulan veri seti üzerinde çalıştırılmıştır. Eğitim işlemi sırasında kullanılan yitim değerleri kayıt edilerek aşağıda görülen Şekil 10. Yitim Değerleri Çizelgesinde birbirleri ile kıyaslanmıştır. Son yitim değerleri yaklaşık olarak YOLO-V2 modeli için 1.15 YOLO-V3 modeli için 0,82 ve YOLOV3-Dense için 0,36'dır. Orijinal YOLO-V3 ve YOLOV3-Dense arasında yaklaşık olarak 0,46 fark bulunmaktadır. Güven skoru eşik değeri 0,25 olarak alınıp ortalama hassasiyet değerleri karşılaştırıldığında YOLOV3-Dense modelinin %81,2 değeri ile orijinal YOLO-V3



Şekil 10. Modellerin yitim değerleri (Loss values of models)

modelinden yaklaşık olarak %10 daha iyi olduğu Şekil 11 ortalama hassasiyet değerleri çizelgesinde görülmektedir. YOLO-V2, YOLO-V3, YOLOV3-Dense modellerine ait F1-Skor, IoU, mAP ve Ortalama tespit zamanı değerleri Tablo 1’de görülmektedir. Çizelge ve tablolara bakıldığında YOLOV3-Dense modelinin sınırlayıcı kutu tespiti işleminde diğer iki modelden daha iyi olduğu görülmüştür.

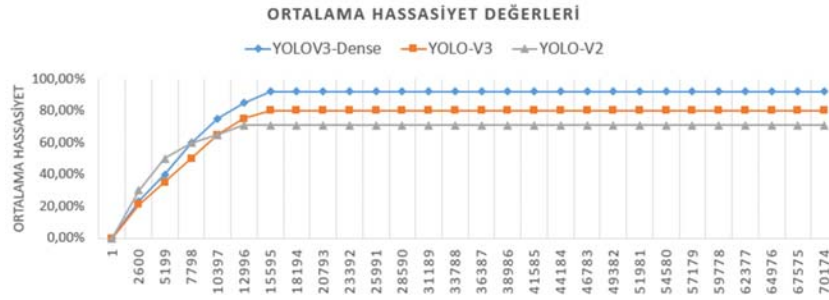
Güven skoru eşik değeri 0,50 olarak belirlenerek bu değer altında kalan sınırlayıcı kutular göz ardı edilmiş ve mAP, F1-skor ve IoU değerleri hesaplanırken bu kutular işleme alınmamıştır. Güven skoru eşik değerinde yapılan bu değişim ile YOLO-V2, YOLO-V3 ve YOLOV3-Dense modellerinin tümünde mAP, F1-skor ve IoU değerlerinin yaklaşık olarak %10 daha iyi performans gösterdiği Tablo 2’te görülmektedir. DPY tespit edilmiş 3 adet örnek görüntü Şekil 13’te görülmektedir. Her bir görüntü 3 model ile ayrı ayrı test edilerek çıkan sonuçlar Tablo 3’te toplanmıştır. a, d, g görselleri YOLO-V2 ile b, e, h görüntüleri YOLO-V3 ile c, f, i görüntüleri YOLOV3-Dense ile tespit edilmiştir.

Tablo 4’te görüldüğü gibi YOLOV3-Dense modeli, geleneksel görüntü işleme yöntemleri ile yapılan sistemlere göre daha iyi sonuç vermektedir. Geleneksel yöntemler ile

yapılan sistemlerin en büyük dezavantajının düşük fps değeri olduğu görülmektedir. Bu dezavantaj nedeni ile yapılan sistemler gerçek zamanlı olarak kullanılamamaktadır. YOLOV3-Dense modelinde yalnızca doğruluk değeri büyük değil, aynı zamanda fps değeri de geleneksel yöntemlere göre oldukça yüksektir.

Tablo 3. Şekil 13’de bulunan görüntülere ait güven skoru değerleri. (The confidence score values of the images in Figure 13.)

Görüntü	Kullanılan Model	Güven Skoru
a	YOLO-V2	0,79
b	YOLO-V3	0,81
c	YOLO-V3-Dense	0,85
d	YOLO-V2	0,78-0-0
e	YOLO-V3	0,85-0,53-0,35
f	YOLO-V3-Dense	0,85-0,70-0,43
g	YOLO-V2	0,82-0,80
h	YOLO-V3	0,90-0,70
i	YOLO-V3-Dense	0,89-0,84



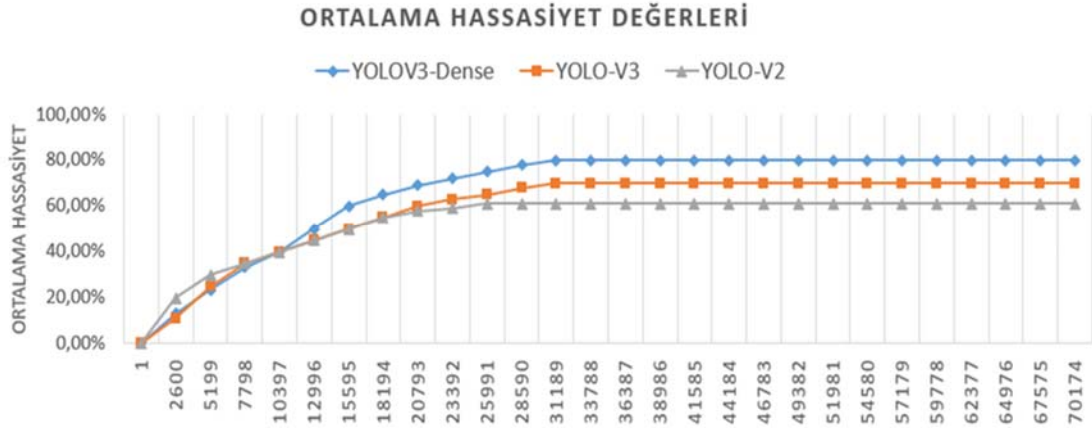
Şekil 11. Ortalama Hassasiyet Değerleri Çizelgesi, Güven Skoru Eşik Değeri=0,50 (mAP values chart, confidence score threshold = 0,50)

Tablo 1. Modellere ait F1-Skor, mAP, IoU ve tespit zamanı değerleri. (F1-Score, mAP, IoU and Detection Time values of the models)

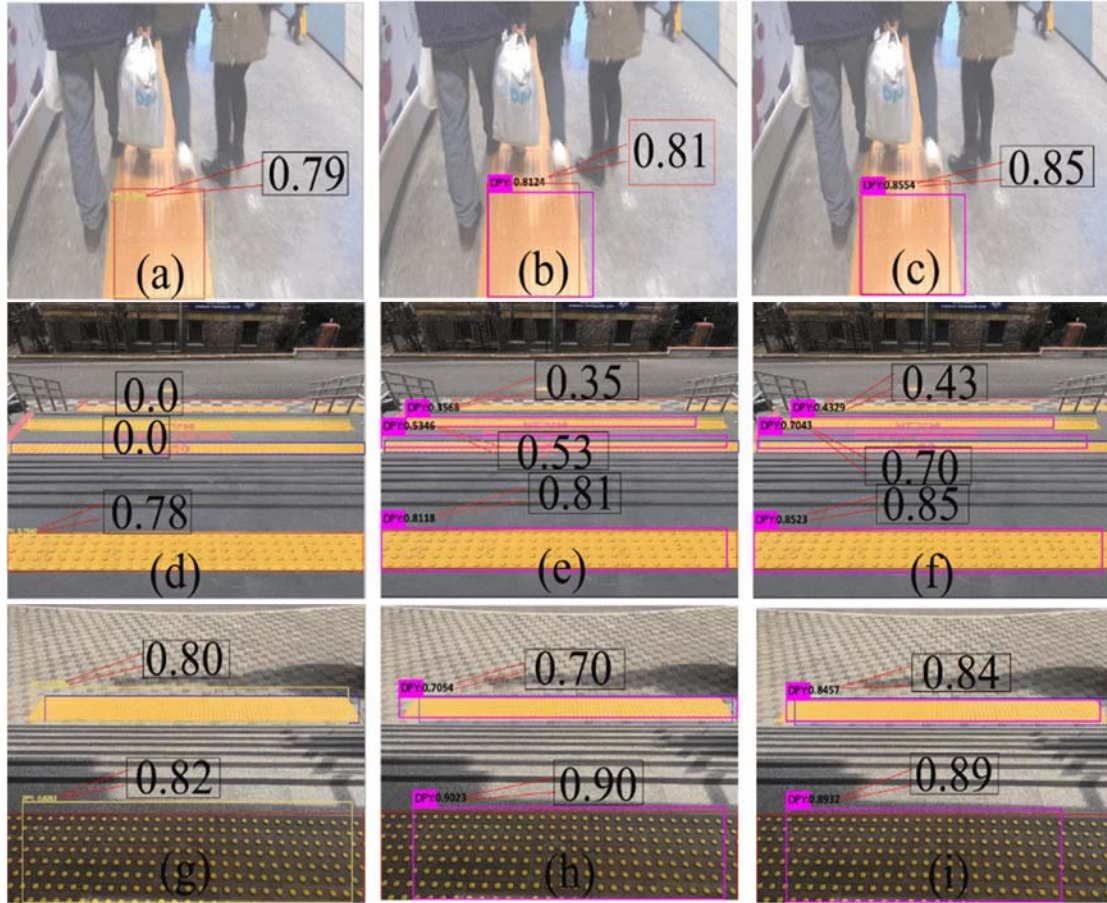
Modeller	F1-Skor	mAP	IoU	Tespit Zamanı(s)	Güven Skoru Eşi Değeri
YOLO-V2	0,701	0,60	0,50	0,201	0,25
YOLO-V3	0,721	0,71	0,61	0,232	0,25
YOLO-V3-Dense	0,814	0,80	0,70	0,254	0,25

Tablo 2. Modellere ait F1-Skor, mAP, IoU ve tespit zamanı değerleri. (F1-Score, mAP, IoU and Detection Time values of the models)

Modeller	F1-Skor	mAP	IoU	Tespit Zamanı(s)	Güven Skoru Eşi Değeri
YOLO-V2	0,69	0,71	0,58	0,201	0,50
YOLO-V3	0,78	0,80	0,68	0,232	0,50
YOLO-V3-Dense	0,89	0,92	0,81	0,254	0,50



Şekil 12. Ortalama Hassasiyet Değerleri Çizelgesi, Güven Skoru Eşik Değeri=0,25
(mAP values chart, confidence score threshold = 0,25)



Şekil 13. DPY Tespit Edilmiş Örnek Görüntüler (The confidence score values of the images in Figure 13.)

3. SİMGELER (SYMBOLS)

- X : Etiketli verinin orta noktasına ait X koordinatı.
- Y : Etiketli verinin orta noktasına ait Y koordinatı.
- W : Etiketli verinin genişliği
- H : Etiketli verinin yüksekliği
- dW : Görüntü genişliği

- dH : Görüntü yüksekliği
- mAP : Ortalama hassasiyet (mean average precision)
- DPY : Dokunsal parke yüzeyi
- MDPY : Marmara dokunsal parke yüzeyi veriseti adı
- IoU : gerçek referans değer ile tahmin edilen değerın keşimlerinin tüm alana oranı (intersection over union)

Tablo 4. Diğer yöntemler ile karşılaştırma (Comparison with the other methods)

Çalışma	Kullanılan Yöntem	Doğruluk	fps	Avantajlar	Dezavantajlar
[4]	Canny, Laplace, Sobel Kenar Belirleme	%88,48	16,27	Geleneksel görüntü işleme algoritmalarına ait örnekler mevcuttur.	Düşük fps değeri. Yüksek maliyet. Kullanım zorluğu.
[18]	Algoritmaları ve Karar Ağaçları Gri seviyeli eş-oluşum matrisleri ve Destek Vektör Makineleri	-	-	Danışmanlı öğrenme yöntemlerine ait örnekler mevcuttur.	Sistemin çıktıları ile ilgili herhangi bir bilgi verilmemiştir.
[19]	Çok Renkli Görüntü Segmentasyonu ve Kirsch Kenar Belirleme Algoritması.	-	2	Geleneksel görüntü işleme algoritmaları ve donanımsal parçaların bir arada kullanılmasına ait örnekler mevcuttur.	Düşük fps değeri. Yüksek maliyet. Kullanım zorluğu.
[31]	YOLO-V2	%71(mAP)	70	Diğer YOLO modellerine kıyasla daha sığ bir ağ yapısı olduğundan fps değeri yüksektir.	Doğruluk ve mAP değerleri düşüktür.
[30]	YOLO-V3	%80(mAP)	60	Çok katmanlı ağ yapılarına ait örnekler mevcuttur.	Küçük nesnelere tespit edilmesinde düşük doğruluk değeri.
	YOLOV3-Dense	%92(mAP)	60	DenseNet sayesinde görüntüde bulunan tüm öznitelikler ağda bulunan tüm katmanlarda tekrar tekrar kullanılmış ve görüntü üzerindeki küçük nesnelere tespiti sağlanmıştır.	Görüntü üzerindeki tüm özniteliklerin kullanılması ağda bulunan parametre sayısını artırmıştır.

4. SONUÇLAR VE TARTIŞMALAR (RESULTS AND DISCUSSIONS)

Bu çalışmada, nesne tespit etme alanında kullanılan YOLO-V3 modeli DenseNet modeli ile birleştirilerek dokunsal parke yüzeylerinin tespiti yapılmıştır. Bu çalışma ile dokunsal parke yüzeylerinin tespiti yapılabilmekte ve tespit edilen nesnenin görüntü içerisinde hangi konumda olduğu bilgisine erişilebilmektedir. YOLOV3-Dense modeli ile YOLO-V3 modelinde bulunan öznitelik katmanlarındaki düşük çözünürlüğe sahip görüntülerin iyileştirilmesi amaçlanmıştır. Bu amaç doğrultusunda görüntüde bulunan özniteliklerin tüm katmanlarda eşit bir şekilde yayılması DenseNet yöntemi ile sağlanmış ve ağ performansı artırılmıştır. Deneysel çalışmalar sonucunda YOLOV3-Dense modelinin %89 F1-skor, %92 ortalama hassasiyet ve %81 IoU değerleri ile diğer YOLO modellerinden çok daha iyi sonuç verdiği görülmektedir.

Gelecek çalışmalarda tarafımızca oluşturulmuş olan Marmara Dokunsal Parke Yüzeyi (MDPY) veri seti genişletilerek daha iyi sonuçlar alınması planlanmaktadır. Bu çalışmada kullanılan veri artırma yöntemleri ve tespit etme algoritmaları optimize edilerek tespit etme doğruluğunun artırılması ve çalışma sonucunda elde edilen evrimsel sinir ağı ağırlıklarının farklı platformlarda

kullanılarak gündelik hayatta kullanılabilen mobil uygulamalar geliştirilmesi planlanmaktadır. Gerçek zamanlı olarak tespit edilen dokunsal parke yüzeylerine ait konum bilgileri elde edilebilmektedir. Bu konum bilgilerine göre ses ile yönlendirme yapan etkili bir yardımcı teknoloji yapılması planlanmaktadır. Gelecek çalışmalarda MDPY veri setinde insanlar, araçlar, ağaçlar, tabelalar gibi birçok nesne etiketlenerek, görme engelli bireylere etrafta bulunan engeller ve nesnelere hakkında bilgilendirme yapılması planlanmaktadır. Yapılması planlanan uygulamalar sayesinde kullanıcı ekstra herhangi bir donanıma ihtiyaç duymadan yalnızca akıllı telefon ile sistemi kullanabilecektir.

KAYNAKLAR (REFERENCES)

1. World Health Organization, Blindness and vision impairment, <https://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment>, Yayın tarihi Ekim 8, 2019, Erişim tarihi Kasım 11, 2019.
2. Lu, J., Siu, K. W. M., & Xu, P., A comparative study of tactile paving design standards in different countries. 9th International Conference on Computer-Aided Industrial Design and Conceptual Design, China, 753-758, 22-25 Nov. 2008.

3. Asami, T., & Ohnishi, K., Crosswalk location, direction and pedestrian signal state extraction system for assisting the expedition of person with impaired vision, 2014 10th France-Japan/8th Europe-Asia Congress on Mechatronics (MECATRONICS), Tokyo, 285-290, 27-29 Nov. 2014
4. Ghilardi, M. C., Macedo, R. C. O., & Manssour, I. H., A New Approach for Automatic Detection of Tactile Paving Surfaces in Sidewalks, *Procedia Computer Science*, 80, 662-672, 2016.
5. Mancini, A., Frontoni, E., & Zingaretti, P., Mechatronic System to Help Visually Impaired Users During Walking and Running. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 19(2), 649-660, 2018.
6. Shoval, S., Borenstein, J., & Koren, Y., The NavBelt-a computerized travel aid for the blind based on mobile robotics technology, *IEEE Transactions on Biomedical Engineering (TBME)*, 45(11), 1376-1386, 1998.
7. Yang, K., Cheng, R., Wang, K., & Zhao, X., A ground and obstacle detection algorithm for the visually impaired, 2015 IET International Conference on Biomedical Image and Signal Processing (ICBISP 2015), Beijing, China. DOI: 10.1049/cp.2015.0777, 2015.
8. Garcia-Garcia, A., Orts, S., Oprea, S., Villena Martinez, V., Martinez-Gonzalez, P., & Rodríguez, J., A Survey on Deep Learning Techniques for Image and Video Semantic Segmentation, *Applied Soft Computing*, 70, 41-65, 2018.
9. Krizhevsky, A., Sutskever, I., & E. Hinton, G., ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems*, 25(2), Jan. 2012.
10. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Li, F. F., & Ieee. ImageNet: A Large-Scale Hierarchical Image Database, *IEEE-Computer-Society Conference on Computer Vision and Pattern Recognition Workshops*, Miami Beach, FL, 248-255, 20-25 Jun. 2009.
11. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Li, F. F., ImageNet Large Scale Visual Recognition Challenge, *International Journal of Computer Vision (IJCV)*, 115, 211-252, 2015.
12. He, K., Zhang, X., Ren, S., & Sun, J., Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, *Proceedings of the IEEE international conference on computer vision*, Massachusetts-USA, 1026-1034, 2015.
13. Tan, M., & Le, Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, 2019 International Conference on Machine Learning(ICML), California-USA, 6105-6114, 2019.
14. Girshick, R., Donahue, J., Darrell, T., & Malik, J., Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, Columbus-OH-USA, 580-587, 25 Sept. 2014.
15. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C., Ssd: Single shot multibox detector, *European conference on computer vision(ECCV2016)*, Amsterdam-Netherlands, 21-37, 11-14 Oct. 2016.
16. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A., You only look once: Unified, real-time object detection, *IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 779-788, 27-30 June 2016.
17. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y., Overfeat: Integrated recognition, localization and detection using convolutional networks, *International Conference on Learning Representations(ICLR2013)*, arXiv:1312.6229 2013.
18. Einloft, D. C., Ghilardi, M. C., & Manssour, I. H., Automatic Detection of Tactile Paving Surfaces in Indoor Environments, *Workshop of Undergraduate Works (WUW) in the 29th Conference on Graphics, Patterns and Images (SIBGRAPI'16)*, 2016.
19. Jie, X., Xiaochi, W., & Zhigang, F., Research and implementation of blind sidewalk detection in portable eta system, *International Forum on Information Technology and Applications (IFITA2010)*, Kunming-China, 431-434, 16-18 July 2010.
20. Kassim, A. M., Yasuno, T., Mohd Aras, M. S., hj shukor, A., Jaafar, H. I., Baharom, F., & Jafar, F., Vision based of tactile paving detection method in navigation system for blind person, *Jurnal Teknologi*, 77, No:20, 11 August 2015.
21. Shen, J., Liu, N., Sun, H., Tao, X., & Li, Q., Vehicle Detection in Aerial Images Based on Hyper Feature Map in Deep Convolutional Network, *KSII Transactions on Internet & Information Systems*, 13(4).No:4, 29 April 2019.
22. Ren, S., He, K., Girshick, R., & Sun, J., Faster r-cnn: Towards real-time object detection with region proposal networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE T PATTERN ANAL)*, 39,No:6, 4 June 2015.
23. Du, Y., Pan, N., Xu, Z., Deng, F., Shen, Y., & Kang, H., Pavement distress detection and classification based on YOLO network, *International Journal of Pavement Engineering*, doi=10.1080/10298436.2020.1714047, 24 Jan. 2020.
24. Wang, J., Wang, N., Li, L., & Ren, Z., Real-time behavior detection and judgment of egg breeders based on YOLO v3, *Neural Computing and Applications (NEURAL COMPUT APPL)*, doi: 10.1007/s00521-019-04645-4, 2019.
25. Kaya, A., Keçeli, A. S., & Can, A. B., Examination of various classification strategies in classification of lung nodule characteristics, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 34(2), 709-725, 2019.
26. Yildiz, O., Melanoma detection from dermoscopy images with deep learning methods: A comprehensive study, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 34(4), 2241-2260, 2019.

27. Dandil, E., & Polattimur, R., Dog Behavior Recognition and Tracking based on Faster R-CNN, Journal of the Faculty of Engineering and Architecture of Gazi University, 35(2), 819-834, 2019.
28. Lv, J.-J., Shao, X.-H., Huang, J.-S., Zhou, X.-D., & Zhou, X., Data augmentation for face recognition. Neurocomputing, 230, 184-196, 2017.
29. Shorten, C., & Khoshgoftaar, T. M., A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1), No: 60, (2019).
30. Redmon, J., & Farhadi, A., Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767, 2018.
31. Redmon, J., & Farhadi, A., YOLO9000: Better, Faster, Stronger, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Hawaii-USA, 6517-6525, 21-26 July 2017.
32. Huang, G., Liu, Z., Maaten, L. v. d., & Weinberger, K. Q., Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Hawaii-USA, 2261-2269, 21-26 July 2017.
33. Goutte, C., & Gaussier, E., A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation, Proceedings of the 27th European conference on Advances in Information Retrieval Research, Spain, 345-359, 2005.
34. Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A., The pascal visual object classes (voc) challenge, International journal of computer vision, 88(2), 303-338, 2010.