

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

DERİN ÖĞRENME TEMELLİ SÜRÜLEBİLİR ALAN TESPİTİ
VE GÖMÜLÜ UYGULAMASI

ONUR ACUN

KOCAELİ 2021

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

DERİN ÖĞRENME TEMELLİ SÜRÜLEBİLİR ALAN TESPİTİ
VE GÖMÜLÜ UYGULAMASI

ONUR ACUN

Prof.Dr. Oğuzhan URHAN

Danışman, Kocaeli Üniversitesi

.....

Prof.Dr Raşit KÖKER

Jüri Üyesi, Sakarya Üniversitesi

.....

Dr.Öğr.Üyesi Ayhan KÜÇÜKMANİSA

Jüri Üyesi, Kocaeli Üniversitesi

.....

Tezin Savunulduğu Tarih: 21.06.2021

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında, derin öğrenme temelli bir sürülebilir alan tespiti yöntemi bilgisayar ve gömülü platform üzerinde gerçek zamalı olarak başarılı bir şekilde uygulanmıştır. Yapılan çalışmanın bu alanda çalışanlara faydalı olmasını temenni ederim.

Tez çalışması süresince verdiği fikirler ve yaptığı yardımlar ile desteğini esirgemeyen tez danışmanım ve değerli hocam Prof. Dr. Oğuzhan URHAN'a teşekkür ederim.

Yine bu süreçte takıldığım ve eksik kaldığım bütün noktalarda bana destek olan ve beni her zaman motive eden Dr. Ayhan KÜÇÜKMANİSA'ya teşekkür ederim.

Doğduğum günden bu zamana kadar beni koşulsuz destekleyen ve her zaman yanımda olan aileme çok teşekkür ederim.

Haziran - 2021

Onur ACUN

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iii
TABLOLAR DİZİNİ	v
SİMGELER VE KISALTMALAR DİZİNİ	vi
ÖZET.....	vii
ABSTRACT	viii
GİRİŞ	1
1. SÜRÜLEBİLİR ALAN TESPİTİ	3
2. LİTERATÜRDEKİ ÇALIŞMALAR	5
2.1. Lidar Temelli Çalışmalar	5
2.2. Lidar ve Görü Temelli Hibrit Çalışmalar.....	7
2.3. Görü Temelli Çalışmalar.....	9
2.3.1. Görüntü işleme temelli yöntemler.....	9
2.3.2. Makine öğrenmesi temelli yöntemler.....	11
2.3.3. Derin öğrenme temelli yöntemler	13
3. TEZ KAPSAMINDA GERÇEKLEŞTİRİLEN SÜRÜLEBİLİR ALANLARIN TESPİTİ ÇALIŞMASI.....	18
3.1. Sürülebilir Alan Tespiti Performansının Değerlendirilmesi	18
3.2. Regresif Çıkışlı Konvolüsyonel Sinir Ağı ile Gerçek Zamanlı Sürülebilir Alan Tespiti	19
3.2.1. Ön işlem	20
3.2.2. Konvolüsyonel sinir ağı modeli	22
3.2.3. Son işlem.....	23
4. DENEYSEL SONUÇLAR.....	25
5. SONUÇLAR VE ÖNERİLER	33
KAYNAKLAR	35
KİŞİSEL YAYINLAR VE ESERLER	39
ÖZGEÇMİŞ	40

ŞEKİLLER DİZİNİ

Şekil 1.1.	Trafikte bulunabilecek örnek sürülebilir alanlar (a) Açık sürülebilir alan (b) Sürülebilir alanın dönüş sırasında yol sınırları (c) Sürülebilir alanda araç ve engel bulunması (d) Sürülebilir alanlardan yaya geçişi (e) Sürülebilir alan farklı zemin tipi ve yaya geçişi (f) Sürülebilir alan farklı zemin tipi	3
Şekil 1.1.	(Devam) Trafikte bulunabilecek örnek sürülebilir alanlar (a) Açık sürülebilir alan (b) Sürülebilir alanın dönüş sırasında yol sınırları (c) Sürülebilir alanda araç ve engel bulunması (d) Sürülebilir alanlardan yaya geçişi (e) Sürülebilir alan farklı zemin tipi ve yaya geçişi (f) Sürülebilir alan farklı zemin tipi	4
Şekil 2.1.	Sürülebilir alan tespiti sistemlerinin sınıflandırılması	5
Şekil 2.2.	Görü temelli çalışmaların genel gösterimi	9
Şekil 3.1.	Veri seti örnek görüntüsü (a) Giriş görüntüsü (b) Yer doğrusu haritası	18
Şekil 3.2.	Önerilen yöntemin akış diyagramı	20
Şekil 3.3.	Ön işlem çıkışında oluşan ince dikey şeritler	21
Şekil 3.4.	Sürülebilir alanların tespiti için oluşturulan KSA modeli	23
Şekil 3.5.	Önerilen yöntemin sürülebilir alan tespiti sonuçları (a) Orijinal görüntü (b) Derin öğrenme modeli tarafından her bir ince şerit dilimi görüntüsünde tespit edilen noktalar (c) Orijinal görüntü üzerinde tespit edilen noktalar ile çıkartılmış sürülebilir alan sonucu	24
Şekil 4.1.	Örnek bir görüntü üzerinde önerilen yöntemin derin öğrenme modelinin farklı giriş boyutlardaki sonuçları (a) 64×1024 piksel (b) 32×1024 piksel (c) 16×1024 piksel (d) 64×512 piksel (e) 32×512 piksel (f) 16×512 piksel (g) 64×256 piksel (h) 32×256 piksel (ı) 16×256 piksel	25
Şekil 4.1.	(Devam) Örnek bir görüntü üzerinde önerilen yöntemin derin öğrenme modelinin farklı giriş boyutlardaki sonuçları (a) 64×1024 piksel (b) 32×1024 piksel (c) 16×1024 piksel (d) 64×512 piksel (e) 32×512 piksel (f) 16×512 piksel (g) 64×256 piksel (h) 32×256 piksel (ı) 16×256 piksel	26
Şekil 4.2.	Örnek bir görüntü üzerinde önerilen yöntemin derin öğrenme modelinin farklı giriş boyutlardaki sonuçları (a) 64×1024 piksel (b) 32×1024 piksel (c) 16×1024 piksel (d) 64×512 piksel (e) 32×512 piksel (f) 16×512 piksel (g) 64×256 piksel (h) 32×256 piksel (ı) 16×256 piksel	26
Şekil 4.3.	Örnek bir görüntü üzerinde önerilen yöntemin derin öğrenme modelinin farklı giriş boyutlardaki sonuçları (a) 64×1024 piksel (b) 32×1024 piksel (c) 16×1024 piksel (d) 64×512 piksel (e) 32×512 piksel (f) 16×512 piksel (g) 64×256 piksel (h) 32×256 piksel (ı) 16×256 piksel	27

Şekil 4.4. Örnek bir görüntü üzerinde önerilen yöntemin derin öğrenme modelinin farklı giriş boyutlardaki sonuçları (a) 64×1024 piksel (b) 32×1024 piksel (c) 16×1024 piksel (d) 64×512 piksel (e) 32×512 piksel (f) 16×512 piksel (g) 64×256 piksel (h) 32×256 piksel (ı) 16×256 piksel.....	27
Şekil 4.5. Örnek bir görüntü üzerinde önerilen yöntemin derin öğrenme modelinin farklı giriş boyutlardaki sonuçları (a) 64×1024 piksel (b) 32×1024 piksel (c) 16×1024 piksel (d) 64×512 piksel (e) 32×512 piksel (f) 16×512 piksel (g) 64×256 piksel (h) 32×256 piksel (ı) 16×256 piksel.....	28
Şekil 4.6. Karşılaştırılan yöntemlerin FPS-Başarım Grafiği.....	30
Şekil 4.7. Karşılaştırılan yöntemlerin Parametre-Başarım Grafiği	31



TABLolar DİZİNİ

Tablo 2.1. Yakın zamanda önerilen lidar temelli sürülebilir alan tespiti yöntemlerinin karşılaştırılması.....	6
Tablo 2.1. (Devam) Yakın zamanda önerilen lidar temelli sürülebilir alan tespiti yöntemlerinin karşılaştırılması.....	7
Tablo 2.2. Yakın zamanda önerilen lidar ve görü temelli sürülebilir alan tespiti yöntemlerinin karşılaştırılması.....	8
Tablo 2.2. (Devam) Yakın zamanda önerilen lidar ve görü temelli sürülebilir alan tespiti yöntemlerinin karşılaştırılması.....	9
Tablo 2.3. Önerilen görüntü işleme temelli sürülebilir alan tespiti yöntemlerinin karşılaştırılması.....	11
Tablo 2.4. Yakın zamanda önerilen makine öğrenmesi temelli sürülebilir alanların tespiti yöntemlerinin karşılaştırılması.....	12
Tablo 2.4. (Devam) Yakın zamanda önerilen makine öğrenmesi temelli sürülebilir alanların tespiti yöntemlerinin karşılaştırılması.....	13
Tablo 2.5. Yakın zamanda önerilen derin öğrenme temelli şerit tespit yöntemlerinin karşılaştırılması.....	16
Tablo 2.5. (Devam) Yakın zamanda önerilen derin öğrenme temelli şerit tespit yöntemlerinin karşılaştırılması.....	17
Tablo 3.1. Orijinal ve ön işlem sonrası oluşan görüntü adetleri ve boyutları.....	22
Tablo 4.1. Farklı giriş boyutundaki modellerin cityScapes test seti üzerindeki başarımları ve hız değerleri.....	28
Tablo 4.2. Farklı giriş boyutundaki modellerin tensorRT dönüşümü sonrası cityScapes test seti üzerindeki başarımları ve hız değerleri.....	29
Tablo 4.3. Sürülebilir alan tespit yöntemlerinin CityScapes test veri seti üzerinde performans değerlendirmesi.....	29
Tablo 4.4. Önerilen yöntem ve karşılaştırılan yöntemlerin parametre-başarımları tablosu.....	30
Tablo 4.5. Gömülü platform özellikleri.....	31
Tablo 4.6. Jetson Nano üzerinde farklı giriş boyutundaki modellerin CityScapes test seti üzerindeki başarımları ve hız değerleri.....	32
Tablo 4.7. Jetson Xavier NX üzerinde farklı giriş boyutundaki modellerin tensorRT dönüşümü sonrası CityScapes test seti üzerindeki başarımları ve hız değerleri.....	32
Tablo 4.8. Jetson Xavier NX üzerinde önerilen yöntemin ve farklı segmentasyon yöntemlerinin CityScapes test seti üzerindeki başarımları ve hız değerleri.....	32

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

ACC	: Adaptive Cruise Control (Uyarlanabilir Hız Kontrolü)
ADAS	: Advanced Driver Assistance Systems (İleri Sürücü Yardım Sistemleri)
ASPP	: Atrous Spatial Pyramid Pooling (Aşırı Uzaysal Piramit Havuzu)
CPU	: Central Processing Unit (Merkezi İşlem Birimi)
DAD	: Drivable Area Detection (Sürülebilir Alan Tespiti)
DLSTM	: Distributed Long-Short Term Memory (Dağıtık Uzun-Kısa Süreli Bellek)
ESP	: Efficient Spatial Pyramid (Verimli Mekansak Piramit)
FP32	: Floating Point 32 (Kayan Nokta 32)
FP16	: Floating Point 16 (Kayan Nokta 16)
FPS	: Frame Per Second (Saniyelik Görüntü Sayısı)
GPU	: Graphical Processing Unit (Grafik İşlem Birimi)
ITS	: Intelligent Transportation Systems (Akıllı Ulaşım Sistemleri)
iGUM	: Improved Guided Upsampling Module (Geliştirilmiş Güdümlü Üst Örnekkele Modülü)
LIDAR	: Light Detection and Ranging (Işık Tespiti ve Ölçümü)
NAS	: Neural Architecture Search (Sinir Mimarisi Araması)
OS	: Operating System (İşletim Sistemi)
RAM	: Random Access Memory (Rastgele Erişim Belleği)
ReLU	: Rectified Linear Unit (Doğrultulmuş Doğrusal Birim)
RPNet	: Feature Residual Pyramid Network (Özellik Artıklığı Piramit Ağı)

DERİN ÖĞRENME TEMELLİ SÜRÜLEBİLİR ALAN TESPİTİ VE GÖMÜLÜ UYGULAMASI

ÖZET

Gelişen teknoloji ve teknolojik cihazların yaygınlaşması ile birlikte trafikte akıllı sürüş sistemleri ve otonom araç teknolojileri trafik ve sürücü güvenliğini artırmak için sıklıkla kullanılmaya başlanmıştır. Bu gibi sistemler sürücüleri ve aracı seyir halindeyken güvende tutmayı amaçlamaktadır. Bu tez çalışması kapsamında akıllı sürüş sistemleri ve otonom araçlarda sıklıkla kullanılan bir sürülebilir alan tespiti yöntemi önerilmiştir. Önerilen yöntem derin öğrenme temelli bir yaklaşım ile bilgisayar ve gömülü sistem platformlarında gerçek zamanlı çalışabilme kapasitesine sahiptir.

Sürülebilir alan tespiti yöntemlerinde aracın önündeki güvenli sürüş yapılabileceği her bir noktanın doğru bir şekilde tespit edilmesi amaçlanmaktadır. Bu çalışma kapsamında geleneksel segmentasyon modellerinin aksine regresif çıkışlı basit bir konvolüsyonel sinir ağı kullanılarak gerçek zamanlı sürülebilir alanların tespiti yöntemi uygulanmıştır.

Deneysel sonuçlar, tez kapsamında önerilen yöntemin farklı koşullarda yüksek hızda ve başarılı bir şekilde istenilen sonuçları verebildiğini göstermiştir.

Anahtar Kelimeler: Derin Öğrenme, Konvolüsyonel Sinir Ağları, Regresyon, Sürülebilir Alan.

DEEP LEARNING BASED DRIVABLE AREA DETECTION AND ITS EMBEDDED APPLICATION

ABSTRACT

With the widespread use of developing technology and technological devices, smart driving systems and autonomous vehicle technologies have been used frequently to increase traffic and driver safety. These type of systems aim to keep drivers and the vehicle safe while driving. Within the scope of this thesis, a drivable area detection method, which frequently used in smart driving systems and autonomous vehicles, has been proposed. The proposed method has the capacity to work in real-time on computer and embedded system platform with deep learning based approach.

In the drivable area detection methods, it is aimed to accurately determine each point in front of the vehicle where safe driving can made. In this study, unlike traditional segmentation models, real time drivable area detection method was applied using a simple convolutional neural network with regressive output.

Experimental results have shown that the method proposed in the thesis can successfully yield the desired results in different conditions at high speed.

Keywords: Deep Learning, Convolutional Neural Networks, Regression, Drivable Area.

GİRİŞ

Günümüzde artan nüfus ile beraber yollarda kullanılan araç sayısı da gün geçtikçe artmaktadır. Artan araç sayısı ile birlikte yollarda meydana gelen trafik kazalarının sayısında da artış gözükmemektedir. Meydana gelen kazaların ise büyük bir çoğunluğu sürücü kusurlarından oluşmaktadır.

Teknolojinin de gelişmesi ile beraber sürücü kusurlarını azaltmak ve trafik kazalarını engellemek için yeni yöntemler geliştirilmeye başlanmıştır. Otonom araçlar ve ileri sürüş destek sistemleri de bu sorunların üstesinden gelinmesi için geliştirilen yöntemlerin kullanılması ile ortaya çıkmıştır. Temel olarak bu sistemler araç üzerinde bulunan sensörlerden çeşitli verilerin toplanması ve bu verilerin yorumlanarak seyir halindeki sürücünün uyarılması ve/veya aracın hareketinin otonom bir şekilde yapılarak aracın ve sürücünün güvenliğini sağlamayı amaçlamaktadır.

Otonom araç ve ileri sürüş destek sistemlerine; sürülebilir alan tespiti, şerit tespiti, adaptif hız sabitleyici (Adaptive Cruise Control - ACC) ve trafik işareti tespiti örnek olarak verilebilir.

Otonom araçlarda sıklıkla kullanılan sürülebilir alan tespiti yöntemi ise temelde aracın önünde ya da etrafında serbestçe güvenli bir şekilde hareket edebileceği alanların tespit edilmesini amaçlamaktadır. Hareket halindeyken çevresindeki yolun sınırlarını, var ise önündeki araç, yaya ya da engellere takılmadan sürebileceği güvenli yolu tespit etmeye çalışır. Yol sınırları tespit edilerek araç yol üzerinde güvenli konumlarda seyahat edebilir, belirli bir yakınlıkta araç ya da bir engel var ise bunu anlayıp aracı yavaşlatıp ve/veya durdurabilir.

Bu tez çalışması kapsamında bilgisayar ve gömülü platformlar üzerinde gerçek zamanlı çalışabilen derin öğrenme temelli bir sürülebilir alan tespiti yöntemi üzerine çalışılmıştır. Önerilen yöntem kameradan alınan görüntülerde sürülebilir yol bölgelerini tespit etme yetisine sahiptir. Yöntemin bu özelliği ile elde edilen bilgiler yorumlanarak, ileri sürücü destek sistemlerinde sürücü uyarılabilir veya otonom sürüş

gerçekleştirebilen araçlarda aracın hareket rotasının planlaması ve/veya yol üzerindeki olası engel konumlarının tespit edilmesi gibi geri bildirimler ile otonom sürüş, geliştirilen yöntem tarafından desteklenebilir. Bu tez çalışmasının literatüre katkıları şunlardır:

- Bilgisayar ortamında 4ms, gömülü platform üzerinde 12ms gibi düşük bir çalışma süresine sahip derin öğrenme temelli bir yöntem geliştirilmiştir.
- Sürülebilir alan tespiti yöntemlerinin genelde kullandığı segmentasyon modelleri yerine gömülü sistemler üzerinde de gerçek zamanlı çalışabilen daha basit ve az parametre içeren regresif çıkışlı konvolüsyonel bir sinir ağı uygulanmıştır.

Tez çalışmasının birinci kısmında sürülebilir alan tespiti ve yöntemleri hakkında genel bilgiler verilmiştir.

Tez çalışmasının ikinci kısmında literatürde hali hazırda bulunan farklı sensör ve yöntemler ile gerçekleştirilmiş sürülebilir alan tespiti yöntemleri detaylı olarak incelenmiştir.

Tez çalışmasının üçüncü kısmında sürülebilir alan tespiti için önerilen derin öğrenme temelli yöntem ve yöntemin gelişimi ayrıntılı olarak anlatılmıştır.

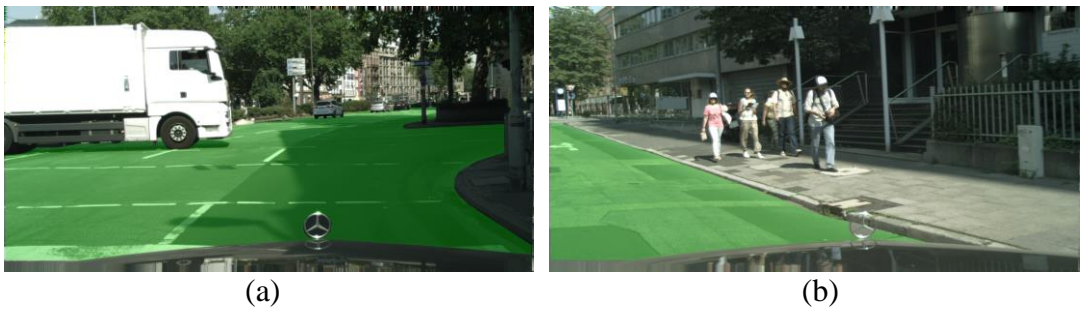
Önerilen yöntem bu gibi yöntemlerin üzerinde çokça test edildiği bir veri seti üzerinde çalıştırılarak deneysel sonuçlar elde edilmiş ve yönteme ait bulgular tez çalışmasının dördüncü kısmında ayrıntılı olarak analiz edilmiştir.

Tez çalışmasının son bölümünde ise önerilen yöntemin bilimsel katkıları ve ileride yapılabilecek çalışmalar için öneriler sunulmaktadır.

1. SÜRÜLEBİLİR ALAN TESPİTİ

Günümüz teknolojilerine ayak uyduran otomobil üreticileri ile birlikte otomobillerde Akıllı Ulaşım Sistemlerinin (Intelligent Transportation Systems - ITS) kullanımı da yaygınlaşmaktadır. Bu gibi akıllı sistemler trafikte sürücüler için seyri kolaylaştırma ve acil durumlarda sürücüyü uyararak ve/veya seyre müdahale ederek sürüş güvenliğini artırmayı amaçlamaktadır. Bu gibi sistemler literatürde İleri Sürücü Destek Sistemleri (Advanced Driver Assistance Systems - ADAS) olarak adlandırılmaktadır. Bu sistemlerden biri olan Sürülebilir Alan Tespiti (Drivable Area Detection - DAD) sürüş sırasında aracın önündeki yol bölgelerini tespit ederek güvenli seyir yapabileceği hız ve alanları hesaplamayı amaçlamaktadır. Sürülebilir alan tespiti algoritmaları genel olarak görüntü üzerinde yol olan her bir pikselin tespit edilmesi ve sonrasında bu piksellerin birleştirilip yorumlanması ile tespit işlemi gerçekleştirilmektedir.

Trafikte sürülebilir alanlar yol kenarlarındaki sınırların, kaldırımların, aracın önünde bulunan başka araçların, engellerin ve/veya yolu sınırlandıracak nesnelere bulunmasına göre değişiklik göstermektedir. Şekil 1.1’de CityScapes veri setinden [1] alınmış örnek bir kaç yol görüntüsü üzerinde sürülebilir alanlar gösterilmektedir. Görüntülerde yeşil gözükten yol bölgeleri sürülebilir alanları temsil etmektedir.



Şekil 1.1. Trafikte bulunabilecek örnek sürülebilir alanlar (a) Açık sürülebilir alan (b) Sürülebilir alanın dönüş sırasında yol sınırları (c) Sürülebilir alanda araç ve engel bulunması (d) Sürülebilir alanlardan yaya geçişi (e) Sürülebilir alan farklı zemin tipi ve yaya geçişi (f) Sürülebilir alan farklı zemin tipi [1]



(c)



(d)



(e)



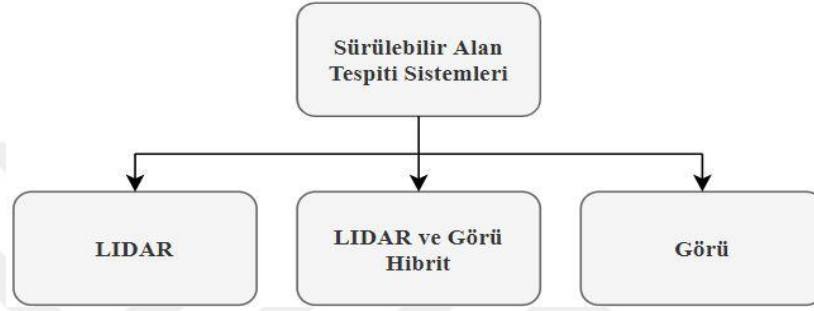
(f)

Şekil 1.1. (Devam) Trafikte bulunabilecek örnek sürülebilir alanlar (a) Açık sürülebilir alan (b) Sürülebilir alanın dönüş sırasında yol sınırları (c) Sürülebilir alanda araç ve engel bulunması (d) Sürülebilir alanlardan yaya geçişi (e) Sürülebilir alan farklı zemin tipi ve yaya geçişi (f) Sürülebilir alan farklı zemin tipi [1]

Şekil 1.1'deki görsellerde olduğu gibi aracın önündeki sürülebilir alanlar farklı nedenler ile değişebilmektedir. Yol sınırları, yol üzerindeki diğer araçlar, yol üzerinde duran engeller ve yoldan geçen yayalar sürülebilir alanların değişmesine neden olan etkenlerdendir. Sürülebilir alanların tespiti yöntemlerinde de geliştirilen yöntemlerin farklı durumlara karşı gürbüz, değişimlere hızlı tepki verebilmesi ve yüksek kare-saniye (frame-rate) değerlerinde çalışması beklenmektedir.

2. LİTERATÜRDEKİ ÇALIŞMALAR

Sürülebilir alan tespiti yöntemlerinde literatürde lidar ve görü sensörleri gibi algılayıcılar sıklıkla kullanılmaktadır. Şekil 2.1’de sürülebilir alan tespiti sistemlerinin sınıflandırılması gösterilmektedir.



Şekil 2.1. Sürülebilir alan tespiti sistemlerinin sınıflandırılması

2.1. Lidar Temelli Çalışmalar

Otonom araçlar ve sürücü destek sistemlerinde LIDAR (Light Detection and Ranging) aracın çevresini sezinlemesi için kullanılan en popüler algılayıcılardan biridir [2]. LIDAR lazer atımları kullanarak bir nesne veya yüzey hakkında mesafe ve yüzey yansıtıcılığı gibi bilgileri edinmemizi sağlar. LIDAR’ın temel çalışma prensibi, kaynaktan çıkan lazer ışınının bir nesneye çarpıp alıcı sensöre geri gelmesi ve arasında geçen süreye ve gelen lazerin genliğine bakılarak, uzaklık ve yüzey yansıtıcılığı hakkında bilgi vermesidir. Bu özellikleri nedeniyle LIDAR, 3-boyutlu modelleme, haritalama, doğal kaynak arama, otonom araçlar gibi farklı kullanım alanlarına sahiptir. Yakın zamanda yayınlanan LIDAR temelli yöntemlerin karşılaştırılması yapılmış ve Tablo 2.1’de karşılaştırmalı olarak gösterilmiştir.

[3]’deki çalışmada LIDAR ile toplanan nokta kümeleri kullanılarak derin öğrenme temelli bir sürülebilir yol bölgelerini segmente eden ve yol üzerindeki engelleri tespit eden bir yöntem önerilmiştir. Önerilen yöntemde SSADNet isimli hem segmentasyon hemde nesne tespiti çıkışı veren tek bir derin öğrenme modeli geliştirilmiştir. LIDAR’dan alınan nokta kümeleri derin öğrenme modeline girdi olarak verilebilmesi için 3 kanallı kuş bakışı bir görsele dönüştürülmüştür. Bu kanallar sırasıyla uzaklık,

yoğunluk ve yükseklik değerlerini içermektedir. Oluşturulan bu veriler kullanılarak model KITTI [4] veri seti üzerinde eğitilmiş ve başarımlar çıkartılmıştır. Önerilen yöntem 33 fps çalışma hızına sahip olup, sürülebilir alan tespitinde %83.57'lik bir başarımlar elde etmiştir.

[5]'de FPGA üzerinde gerçek zamanlı çalışabilen derin öğrenme temelli bir sürülebilir alan segmentasyonu yöntemi önerilmiştir. LIDAR ile toplanan nokta kümeleri ön işleme ile örneklenerek sonrasında tek kanallı bir görüntü oluşturacak biçimde dönüştürülmüştür. Dönüştürülen bu veriler önerilen yöntemde geliştirilen ChipNet isimli bir derin öğrenme algoritmasına girdi olarak verilmiştir. Ağın çıkışında elde edilen segmentasyon sonucu da son işlem olarak kuş bakışı formata getirilip sürülebilir alanlar gösterilmiştir. Önerilen yöntem 17ms çalışma hızına sahip ve KITTI [4] veri seti üzerinde %94 başarımlar ile çalıştığı belirtilmiştir.

[6]'de 3-boyutlu LIDAR verileri kullanılarak özgün bir yarı denetimli derin öğrenme temelli "off-road" sürülebilir alan tespiti yöntemi önerilmiştir. Araç üzerine yerleştirilmiş bir LIDAR kamera ile off-road senaryolarında veriler toplanarak özgün bir veri seti oluşturulmuştur. Elde edilen veriler kuş bakışı görünümüne dönüştürülüp 3 sınıf olarak segmente edilmiştir. Bu sınıflar, sürülebilir alan, tampon alan ve engel içeren alan olarak isimlendirilmiştir. Elde edilen veriler ile model eğitilerek off-road senaryoları için sürülebilir alanlar tespit edilmiştir. Çalışmada yöntemin off-road senaryolarında iyi bir başarımla çalıştığı belirtilmektedir.

[7]'deki çalışmada benzer şekilde FPGA üzerinde gerçek zamanlı LIDAR verileri işlenerek sürülebilir alanların tespiti gerçekleştirilmiştir. LIDAR verilerinden elde edilen 3-boyutlu geometri kullanılarak konvolüsyonel sinir ağı ile yol bölgelerini segmente edilmiştir. Çalışmada FPGA üzerinde yüksek hız ve başarımlar elde edildiği belirtilmektedir.

Tablo 2.1. Yakın zamanda önerilen lidar temelli sürülebilir alan tespiti yöntemlerinin karşılaştırılması

Yöntem	Ön İşlem	Özellik Çıkartımı	Son İşlem	Yaklaşım	Platform / Hız (fps)
(Lee ve diğ., 2020) [3]	3D lidar nokta bulutu kuş bakışı dönüşümü	SSADNet (segmentasyon, 9 konv., 4 dekonv., Sınıflandırıcı, 9 konv., 4 regresif FCL)	—	Nesne tespiti ve Sınıflandırma	Tensorflow / 33

Tablo 2.1. (Devam) Yakın zamanda önerilen lidar temelli sürülebilir alan tespiti yöntemlerinin karşılaştırılması

Yöntem	Ön İşlem	Özellik Çıkartımı	Son İşlem	Yaklaşım	Platform / Hız (fps)
(Lyu ve diğ., 2019) [5]	3D lidar nokta bulutu küresel görünüm dönüşümü	4 katmanlı ağ (1 konv., 3 dekonv.),	—	Segmentasyon	FPGA Tensorflow Keras / 59
(Gao ve diğ., 2019) [6]	3D lidar nokta bulutu kuş bakışı dönüşümü	7 konv. ve 3 dekonv. Katmanı, Softmax sınıflandırıcı	—	Segmentasyon	— / —
(Lyu ve diğ., 2018) [7]	3D lidar nokta bulutu küresel görünüm dönüşümü	11 katmanlı ağ (2 konv. ve 9 tekrarlı konv.)	—	Segmentasyon	FPGA Caffe / 58

2.2. Lidar ve Görü Temelli Hibrit Çalışmalar

Sürülebilir alanların tespitinde LIDAR ve görünür bölge kameraları sıkça kullanılan sensörlerdir. Literatürdeki yöntemlerin bir kısmı da LIDAR'dan alınan haritalama verileri ve kameradan alınan görüntüler birleştirilerek sürülebilir alanların tespitinde başarıyı artırmayı amaçlamaktadır. Yöntemler genel olarak kameradan aldığı verilerdeki renk bilgisini ve LIDAR'dan aldığı verilerdeki yön ve nesnelere olan mesafe bilgileri kullanılmaktadır. Bu kısımda LIDAR ve görü temelli hibrit yöntemler incelenmiş ve Tablo 2.2'de karşılaştırmaları yapılmıştır.

[8]'de araç önüne takılı LIDAR sensöründen alınan uzamsal bilgiler ile monoküler kameradan alınan piksel renk bilgileri birleştirilerek sürülebilir alanlar tespit edilmektedir. Önerilen yöntemde LIDAR ile elde edilen veriler ile monoküler kamera ile elde edilen veriler ortak nokta haritalama kullanılarak birleştirilmiştir. Görüntüdeki her bir piksel için uzamsal bilgilerden öznitelikler çıkartılarak ilk aşamada görüntü üzerindeki engeller tespit edilmiştir. İkinci aşamada ise görüntüdeki piksel değerlerinin renk özellikleri kullanılarak süperpiksel yöntemi ile kümeleme gerçekleştirilmiş. Kümeleme ve engel tespiti sonuçları bir bayes sınıflandırıcıya verilerek nihai sürülebilir alanlar tespit edilmiştir. Çalışmada yöntemin özgün bir yöntem olduğu ve deneysel sonuçlara bakıldığında yüksek başarımla elde edildiği rapor edilmiştir.

[9]'daki çalışmada benzer bir şekilde 3-boyutlu LIDAR ve bir adet monoküler kamera kullanılarak hem uzamsal hemde piksel renk bilgileri elde edilmiştir. LIDAR

sensöründen alınan yön ve ışın bilgileri analiz edilerek ilk aşamada sürülebilir alan olabilecek adaylar bulunmuştur. Bu adaylar ve ilgili piksel renk bilgileri de kullanılarak denetimsiz öğrenme ile nihai sürülebilir alanlar tespit edilmiştir. Önerilen yöntemin deneysel sonuçlarda yüksek başarımlı ve farklı koşullarda dayanıklı bir şekilde çalıştığı rapor edilmiştir.

[10]'daki çalışmada araç üzerine yerleştirilmiş 3 boyutlu bir LIDAR kamerası önüne yerleştirilmiş görünür bölge kamerasından alınan veriler kullanılarak sürülebilir alanlar tespit edilmiştir. LIDAR sensöründen elde edilen uzamsal bilgiler kullanılarak aracın çevresindeki engeller ve sınır bölgeleri tespit edilmiştir. Yol üzerindeki sürülebilir alanı daha dayanıklı tespit edebilmek için ise görünür bölge kamerasından alınan görüntüler üzerinde kenar tespiti ve renk segmentasyonu yapılarak ikili bir imge oluşturulmuş ve bu imge üzerinde yol şeritleri tespit edilmiştir. Uzamsal ve şerit bilgileri kullanılarak aracın serbestçe hareket edebileceği alanlar belirlenmiştir. Önerilen yöntem iki farklı veri seti üzerinde denenmiş ve yüksek başarımlı ve performans ile çalıştığı belirtilmiştir.

[11]'de araç üzerinde takılı olan 3 adet LIDAR ve 2 adet görünür bölge kamera ile iki aşamalı gerçek zamanlı sürülebilir alanların ve yol şeritlerinin tespiti gerçekleştirilmiştir. Her bir sensör birbiri ile uyum içinde çalışması için kalibre edilmiştir. İlk aşamada LIDAR sensöründen alınan uzamsal bilgiler kullanılarak optimum yol sınırları bir karar algoritması ile belirlenmiş. İkinci aşamada eğer tespit edilebilir ise aracın stabil bir şekilde sürülebilir alanın içinde kalmasını sağlamak için görüntü işleme yöntemleri ile şerit tespiti yapılmıştır. Önerilen yöntem gerçek zamanlı olarak düzenli ve bozuk yollarda dayanıklı bir şekilde çalıştığı belirtilmiştir.

Tablo 2.2. Yakın zamanda önerilen lidar ve görü temelli sürülebilir alan tespiti yöntemlerinin karşılaştırılması

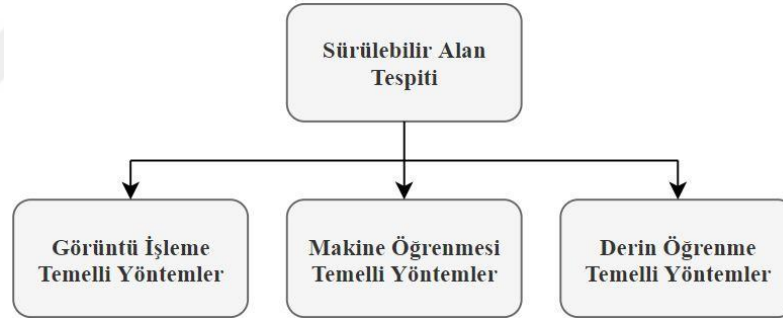
Yöntem	Ön İşlem	Özellik Çıkartımı	Son İşlem	Platform / Hız (FPS)
(Liu ve diğ., 2018) [8]	Lidar ve görünür kamera ortak nokta haritalama ile veri füzyonu	Süper piksel temelli kümeleme, Bayes Sınıflandırıcı	—	PC / —
(Liu ve Oh, 2017) [9]	Görünür bölge renk uzayı dönüşümü, lidar nokta bulutu engel tahmini, yön ışın haritası	Süper piksel temelli kümeleme, markov ağları	—	PC / —
(Raguraman ve diğ., 2020) [10]	3D lidar nokta bulutu ile olası engel noktaları tespiti, renkli kamera kenar ve renk filtreleri	RANSAC, lidar nokta bulutu ile zemin dışı algılama	—	PC / —

Tablo 2.2. (Devam) Yakın zamanda önerilen lidar ve görü temelli sürülebilir alan tespiti yöntemlerinin karşılaştırılması

Yöntem	Ön İşlem	Özellik Çıkartımı	Son İşlem	Platform / Hız (FPS)
(Li ve diğ., 2014) [11]	3 adet lidar ile 3D nokta bulutu füzyonu, renkli görüntü Top-Hat dönüşümü, adaptif eşikleme	Öngün özellik düzeyinde füzyon tabanlı optimum sürülebilir bölge tespiti algoritması	—	PC / —

2.3. Görü Temelli Çalışmalar

Görü temelli sistemlerde sürülebilir alanların tespiti için görünür bölge kameraları sıklıkla kullanılmaktadır. Sürülebilir alanların tespitinde ise aracın önünde serbestçe hareket edebileceği yol bölgelerinin tespiti amaçlanmaktadır. Kameradan alınan verilerde ise uzamsal ve renk bilgileri görüntüdeki yol bölgesi vb diğer bir çok bölgeyi ayırt edebilmeyi sağlamaktadır. Bu tez çalışması kapsamında sürülebilir alanların tespit edilmesine yönelik görü temelli çalışmalar Şekil 2.2’de gösterildiği sınıflandırılmıştır.



Şekil 2.2. Görü temelli çalışmaların genel gösterimi

2.3.1. Görüntü işleme temelli yöntemler

Görüntü işleme temelli yöntemlerde görüntü üzerindeki kenar, alan ve renk vb. gibi öznitelikler geleneksel görüntü işleme algoritmaları kullanılarak elde edilmektedir. Görüntü işleme temelli sürülebilir alanların tespiti yöntemlerinde ise geleneksel görüntü işleme yöntemleri ile tespit edilen yol görsellerindeki öznitelikler kullanılarak görüntü üzerindeki sürülebilir alan içeren bölgelerin tespiti amaçlanmaktadır. Görüntü işleme temelli yöntemler bu kısımda incelenmiş ve Tablo 2.3’te karşılaştırılmıştır.

[12]’de iki aşamalı bir görüntü işleme algoritması ile sürülebilir yol alanlarının tespiti için bir yöntem önerilmiştir. Önerilen yöntemde ilk aşamada kuş bakışı görünümüne

dönüştürülmüş görüntülerde sağ ve sol tarafta bulunan yol sınır adaylarını tespit etmek için görüntüye kenar bulma filtresi uygulanmıştır. Kenarlar tespit edildikten sonra hangi kenar bölgelerinin yol sınırları olabileceğini oylamak için önceden kuş bakışı yol görüntülerinde sıklıkla karşılaşılan 9 adet tanımlanmış yol kıvrımı kullanılmıştır. Sağ ve sol tarafta en yüksek oya sahip kıvrımlar yol sınırları kabul edilmiştir. İkinci aşamada ise sürülebilir yol bölgesini segmente etmek için renk bilgisi kullanılmıştır. Yol sınırları olarak kabul edilen bölgenin içindeki renk bileşenlerinin çok değişkenli bir gauss dağılımını sağladığı varsayılarak bu dağılımın sonucunda bulunan değerleri içeren pikseller yol bölgesi kabul edilmiştir. Bulunan yol sınırları içerisinde bu renk değerine sahip pikseller tespit edilerek sürülebilir alanlar tespit edilmiştir. Önerilen yöntem şehir içi yol tespitinde başarı gösterdiği belirtilmiştir.

[13]'de gerçek zamanlı çalışan bir görüntü işleme algoritması önerilmiştir. Önerilen yöntemde kameradan alınan görüntüler üzerinde ilk olarak ufuk çizgisi üzerinde bulunan gökyüzü bölgesi kırılmıştır. Kırılan görüntü üzerinde orta alt kısımda bir dikdörtgen alanın yol bölgesi içerdiği varsayılarak bu bölgedeki renk değerlerinin histogramına bağlı olarak yöntemde önerilen dinamik eşikleme algoritması ile yol bölgeleri tespit edilerek ikili formata dönüştürülmüştür. Önerilen yöntem bilgisayar üzerinde 10ms gibi bir çalışma hızına sahip ve farklı yol tipleri üzerinde de başarılı bir şekilde çalıştığı belirtilmiştir.

[14]'de araç üzerine takılı binocular renk stereo kamera kullanılarak sürülebilir yol bölgelerinin tespiti amaçlanmaktadır. Önerilen yöntemde ilk olarak stereo görseller üzerinde oluşturulan 3 boyutlu bilgi kullanılarak görüntü üzerinde yer düzlemi uydurma (ground plane fitting) yöntemi ile bir trapezoid alan öğrenme bölgesi olarak seçilmiştir. Bu bölgedeki renk bilgileri kullanılarak sağ kameradan alınan tek görüntü üzerinde renk eşiklemesi kullanılarak yol bölgeleri tespit edilmiştir. Önerilen yöntemin farklı ışıklılık ve ortam koşullarında karşı dayanıklı olduğu ve başarılı çalıştığı belirtilmiştir.

[15]'teki çalışmada yol görüntüleri üzerindeki dokuların baskın bir yönelime sahip olduğu ve bu özelliklerin tespit edilerek sürülebilir yol bölgelerinin bulunması amaçlanmaktadır. Yol görüntüleri gri formata dönüştürüldükten sonra gri değerlerin değişikliklerinin kovaryans matrisine dayalı olarak dokuların yönelimlerini tespit

etmektedir. Son aşamada tespit edilen yönelimler kümelenecek yol bölgeleri tespit edilmektedir.

Tablo 2.3. Önerilen görüntü işleme temelli sürülebilir alan tespiti yöntemlerinin karşılaştırılması

Yöntem	Ön İşlem	Özellik Çıkartımı	Son İşlem	Platform / Hız (FPS)
(He ve diğ., 2004) [12]	Kuş bakışı dönüşümü, kenar bulma filtresi, yol yüzeyi renk tespiti	Olası yol sınır çizgileri oylama, yol sınırları içerisinde renk segmentasyonu	—	PC / —
(Neto ve diğ., 2013) [13]	Ufuk çizgisi tespiti, olası yol bölgesi renk histogramı	Özgün dinamik eşikleme algoritması ile yol bölge tespiti	—	PC / 100
(Tue-Cuog ve diğ., 2008) [14]	Stereo kamera ile yer düzlemi uydurma ve olası yol bölgelerinin seçimi	Seçilen bölge üzerinde renk segmentasyonu	—	PC / 33,3
(Zhang ve diğ., 1994) [15]	Gri seviye dönüşüm, kovaryans matrisine dayalı doku yönelimleri	Doku yönelimleri kümeleme algoritması	—	PC / —

2.3.2. Makine öğrenmesi temelli yöntemler

Makine öğrenmesi temelli sürülebilir alanların tespiti yöntemlerinde görüntü üzerindeki uygun öznitelikler çıkartılarak tercih edilen sınıflandırıcı eğitilmektedir. Eğitim sonucunda oluşan fonksiyona yeni bir görsel giriş olarak verildiğinde bu girdi eğitilen fonksiyon tarafından yorumlanarak istenilen çıkış değeri elde edilmektedir. Bu tür sistemlerde seçilen sınıflandırıcı yöntem ve bu yöntemin öğrenmesi için girdi olarak verilen öznitelikler performansı belirleyen önemli unsurlardandır. Literatürdeki makine öğrenmesi temelli yöntemler incelenmiş ve Tablo 2.4’te karşılaştırılmıştır.

[16]’da görsel üzerindeki sürülebilir alanlar bir denetimli öğrenme yöntemi olan Gauss karışım modeli (gaussian mixture model) sınıflandırıcısı kullanılarak piksel seviyesinde gruplandırılmış ve segmente edilmiştir. Modelin parametreleri beklenti enbüyükleme (expected maximization) algoritması ile eğitilmektedir. Önerilen yöntemde Gauss karışım modelinin başarımını artırmak ve sadece güçlü yol işaretlerine sahip piksellerin tespiti için reddetme kriterini de (rejection criteria) geliştirmektedir. Önerilen yöntem 9 fps ile başarılı bir şekilde çalıştığı belirtilmektedir.

[17]’deki çalışmada rastgele orman (random forest) sınıflandırıcı ile gerçek zamanlı çalışan bir sürülebilir alan segmentasyonu yöntemi önerilmektedir. Önerilen yöntemde

rastgele orman algoritmalarında kullanılan standart önceden belirlenmiş ağırlık ve şekil filtreleri yerine, ağırlıkları öğrenerek optimum özellikleri kendi kendine çıkartabilen esnek filtreli bir rastgele orman algoritması geliştirilmiştir. Önerilen yöntem CityScapes [1] veri seti üzerinde farklı giriş boyutları ve parametreler ile test edilmiş ve yüksek başarımla çalıştığı belirtilmiştir.

[18]'de stereo görüş temelli yol ve yol dışı tespit ve takibini amaçlayan bir yöntem önerilmiştir. Önerilen yöntemde ilk aşamada stereo kameralar kalibre edilerek aralarındaki perspektif matrisleri oluşturulmuş ve kameralar kalibre edilmiştir. Bu kalibrasyon değerleri başarımın düşmesini engellemek için her bir görüntü için dinamik olarak güncellenmektedir. Kalibre edilmiş görseller ikinci aşamada makine öğrenmesi temelli yol ve yol dışı olarak ikili sınıflandırma yapabilen bir sınıflandırıcıya giriş olarak verilmiştir. Sınıflandırma algoritmasında en uygun parametrelerin öğrenilmesi ve başarılı bir şekilde yol ve yol değil ayrımını yapabilmesi için markov rastgele alanı (Markov Random Field - MRF) optimizasyonu kullanılmıştır. Son aşamada ise stereo görme sistemlerinde sıklıkla sorun yaratan dokusuz bölgeler elenerek nihai sürülebilir alanlar tespit edilmiştir. Önerilen yöntem veri toplamak için geliştirilen bir akıllı araç üzerinden alınan görüntüler üzerinde eğitilmiş ve test edilmiştir. Test sonuçlarında yöntemin gölgeli, bozuk vb. zorlu yol görüntülerinde başarılı bir şekilde çalıştığı belirtilmiştir.

[19]'da araç üzerine bulunan kameradan görüntüler üzerinde dikdörtgenel öznitelikler ve histogram geri yansıtma tabanlı bir makine öğrenmesi algoritması önerilmektedir. Önerilen yöntem sürülebilir alanlarının segmentasyonunu amaçlamakta ve yapı olarak AdaBoost sınıflandırıcı yöntemine dayanmaktadır. Önerilen yöntem veri toplamak için geliştirilen otonom bir traktör üzerinden toplanan görüntüler ile eğitilmiş ve test edilmiştir. Yöntemim test veri seti üzerinde %88 oranında bir başarımla elde edildiği belirtilmiştir.

Tablo 2.4. Yakın zamanda önerilen makine öğrenmesi temelli sürülebilir alanların tespiti yöntemlerinin karşılaştırılması

Yöntem	Ön İşlem	Özellik Çıkartımı	Son İşlem	Platform / Hız (FPS)
(Savha ve diğ., 2010) [16]	Görüntü piksel yoğunluğu haritası	Gauss karışım modeli, beklenti maksimizasyonu	Reddetme kriteri	PC / 9
(Kang ve diğ., 2019) [17]	Bilateral filtre	Rastgele orman sınıflandırıcı	—	PC/ 500

Tablo 2.4. (Devam) Yakın zamanda önerilen makine öğrenmesi temelli sürülebilir alanların tespiti yöntemlerinin karşılaştırılması

Yöntem	Ön İşlem	Özellik Çıkartımı	Son İşlem	Platform / Hız (FPS)
(Guo ve diğ., 2012) [18]	Stereo kamera homografi matrisi	Markov rastgele alanları algoritması,	Ortalama kare yatay yoğunluk gradyanı eşikleme	PC / —
(Guo ve diğ., 2006) [19]	Görüntü renk histogramı	AdaBoost, histogram geri yayılımı	—	PC / —

2.3.3. Derin öğrenme temelli yöntemler

Sürülebilir alan tespiti yöntemlerinde sıklıkla karşılaşılan, görüntülerdeki farklı ışıklılık ve ortam koşulları tespit başarımını büyük ölçüde etkilemektedir. Her bir koşul ve farklı durum için tespit edilecek sürülebilir alanların özellikleri değişiklik göstermektedir. Derin öğrenme temelli algoritmalarda sistem verilen giriş görüntüleri üzerindeki öznitelikleri ve filtreleri kendi kendine çıkartabilmektedir. Bu özelliği ile derin öğrenme yöntemleri sürülebilir alanların tespitinde farklı koşullarda yüksek başarımlı ve dayanıklılıkla çalışabilmektedir. Literatürdeki derin öğrenme temelli yöntemler bu kısımda incelenmiş ve karşılaştırmaları Tablo 2.5’te verilmiştir.

[20]’de fast segmentation convolutional neural network (Fast-SCNN) isimli yüksek çözünürlüklü görseller üzerinde gerçek zamanlı çalışabilen, çok sınıflı bir segmentasyon modeli önerilmiştir. Önerilen ağ modelinde “learning to downsample” isimli 1 adet konvolüsyonel ve 2 adet derinlemesine ayrılabilir konvolüsyon içeren bir yapı geliştirilmiştir. Bu yapı aynı anda birden fazla çözünürlük seviyesi için düşük seviye öznitelikleri hesaplayabilmektedir. Modelin devamında çıkışında öznitelik birleştirme özelliği olan bir encoder-decoder mimarisi bulunmakta ve çıktı olarak bir sınıflandırıcı ile segmente edilmiş sonuçlar elde edilmektedir. Önerilen yöntem CityScapes [1] veri seti üzerinde farklı çözünürlüklerde test edilmiş ve yüksek hız ve başarımlı ile çalıştığı belirtilmiştir.

[21]’deki çalışmada FasterSeg isimli gerçek zamanlı çalışabilen bir çok sınıflı segmentasyon modeli geliştirilmiştir. Önerilen model sinir mimari araması (Neural architecture search - NAS) yapısını kullanan ve segmentasyon modellerinde hayati önem taşıyan çok çözünürlüklü katmanlarda paralel işlem yapabilen yeni bir yöntem kullanmaktadır. Önerilen yöntem CityScapes [1] veri seti üzerinde test edilmiştir. Görüntüler üzerinde yol bölgelerinin tespitinde %98’lik bir başarımlı ile 168 fps hızında çalışmaktadır.

LiteSeg [22], hızlı çalışması ve az işlem yükü gerektirmesi için hafif bir mimarisi olan anlamsal görüntü segmentasyonu yapan derin öğrenme temelli bir çalışmadır. Önerilen model MobileNetV2 [23] üzerine geliştirilmiş ve modeli daha verimli hale getirmek için Atrous uzamsal piramit havuzlama (Atrous spatial pyramid pooling - ASPP) modülünün yeni geliştirilmiş versiyonu, kısa ve uzun artıklık bağlantıları ve derinlemesine ayrılabilir konvolüsyon katmanları modele eklenmiştir. Önerilen yöntem CityScapes [1] veri seti üzerinde görseller 640×480 piksel çözünürlüğünde test edilmiş ve 161 fps ile yüksek başarımda tespit yaptığı belirtilmiştir.

[24]'de öznitelik kalıntılı piramit ağı (Feature residual pyramid network - RPN) olarak adlandırılan encoder-decoder mimarisine sahip bir tek atımlı (single-shot) segmentasyon ağı önerilmiştir. Önerilen yöntem spesifik olarak kenarları ve ayrıntıları öğrenmek için katmanlardaki artıklık bilgisini, genel hatların öğrenilmesi için ise konvolüsyonlar sonucu oluşan karakteristik öznitelikleri kullanmaktadır. Model test aşamasında tahmin ettiği artıklık verilerini de tahmin sırasında ayrıntıları geliştirmek için kullanmaktadır. Önerilen yöntemin başarımının hesaplanması sırasında CamVid [25] ve CityScapes [1] veri setleri üzerinde test edilmiş ve gerçek zamanlı ve yüksek başarımla çalıştığı belirtilmiştir.

Segmentasyon ağlarında bulunan encoder-decoder yapılarını verimli hale getirmek ve gerçek zamanlı çalıştırabilmek için [26]'daki çalışmada yeni geliştirilmiş katmanlar ile gerçekleştirilmiş çok sınıflı bir segmentasyon ağı mimarisi önerilmiştir. Önerilen yöntem encoder kısmında ERFNet [27] mimarisinden esinlenilmiştir. Decoder kısmında ise iGUM olarak adlandırılan (improved Guided Upsampling Module) bir üst örnekleme katmanı ve anlamsal örnek segmentasyonunu bir difüzyon süreci olarak modelleyen türevlenebilir bir örnekleme katmanı kullanılmıştır. Geliştirilen bu katmanlar ile daha düşük hesaplama maliyeti ve daha az parametre ile nesne sınırlarının tespiti sağlanmıştır. Önerilen yöntem CityScapes [1], CamVid [25] ve PASCAL VOC 2012 [28] veri setleri üzerinde eğitilmiş ve test edilmiştir. Yöntem bilgisayar üzerinde 113 fps, NVIDIA TEGRA TX1 (Jetson) platformu üzerinde 5 fps ile %97,89 sürülebilir alan tespiti başarımına sahiptir.

ESPNet (efficient spatial pyramid) [29], yüksek çözünürlüklü görüntüler üzerinde hızlı ve verimli çalışabilecek çok sınıflı bir segmentasyon çalışmasıdır. Önerilen modelde

hesaplama bellek ve güç açısından verimi artırmak için verimli uzamsal piramit (efficient spatial pyramid pooling – ESP) adı verilen bir modül geliştirilmiştir. Önerilen modül, standart konvolüsyonel katmanlar yerine noktasal konvolüsyon ve ardından uzamsal genişlemiş konvolüsyon piramidi olan 2 adımdan oluşan katmanlar içermektedir. Kullanılan noktasal konvolüsyon işlemi işlemsel yükü azaltırken, uzamsal genişlemiş konvolüsyon işlemi ise özellik haritalarını yeniden örnekleme yapmaktadır. Önerilen yöntem CityScapes [1] ve PASCAL VOC 2012 [28] veri setleri üzerinde eğitilmiş ve test edilmiştir. Yöntem bilgisayar üzerinde sürülebilir yol bölgelerini 112 fps hızında ve %95,68 başarımla segmente etmektedir.

[30]'da EDANet (efficient dense modules of asymmetric convolution) adında gerçek zamanlı çalışabilen çok sınıflı bir derin öğrenme segmentasyon modeli önerilmiştir. Önerilen yöntemde EDA block isimli modüller geliştirilmiştir ve ek olarak bir decoder yapısına sahip değildir. Geliştirilen bu modüller farklı sayıda ve yapıda birbirine bağlı asimetric konvolüsyon katmanları içermektedir ve her modülün girişi çıkışına birleştirilmiştir. Önerilen yöntem CityScapes [1] ve CamVid [28] veri setleri üzerinde test edilmiş ve bilgisayar ortamında 108 fps ile %97,8 yol bölgesi tespiti başarımla çalıştığı belirtilmiştir.

[31]'de yol tespit başarımlarını artırmak ve yol sınırlarının daha doğru tespit edilebilmesi için tam konvolüsyonel siyam ağı (siamese fully convolutional network) isimli bir derin öğrenme ağı önerilmiştir. Önerilen yöntem renkli giriş görüntüsünü ve aynı görüntünün sınır haritasını paralel olarak girdi alan ve hesapladığı parametreleri birbiri ile paylaşan iki adet encoder ve decoder mimarisine sahip konvolüsyonel sinir ağı içermektedir. Nihai tespit performansını geliştirmek için ağı girdi olarak verilen önceki görüntülerin konum haritası da modele modele girdi olarak verilmektedir. Yöntemde sınır ve konum haritalarından elde edilen bilgiler, renkli görüntü ile birleştirilerek yol bölgelerinin ve sınırlarının daha doğru hesaplanmasını sağlamaktadır. Önerilen yöntem KITTI [4] veri seti üzerinde denenmiş ve yol segmentasyonunu % 98,56 başarımla gerçekleştirmiştir.

[32]'de 112 katmanlı piramit havuzlamaya (Pyramid pooling) sahip konvolüsyonel artıklık ağı önerilmiştir. Önerilen yöntem sürülebilir alanları tespit etmek için encoder-decoder mimarisine sahip sürülebilir yol, yol sınırları ve sürülemez yol sınıflarını

içeren bir segmentasyon ağı kullanmaktadır. Encoder kısmında bulunan 4 adet artıklık katmanı ve 1 adet piramit havuzlama katmanı sayesinde küçük veri setlerinde bile gerekli özneliklerin model tarafından öğrenilebilmesini sağlamıştır. Önerilen yöntem KITTI [4] veri seti üzerinde test edilmiş ve %96,23 başarımla çalıştığı belirtilmiştir.

[33]'deki çalışmada konvolüsyon katmanları ve dağıtık uzun-kısa süreli bellek (Distributed long-short term memory - DLSTM) adında yöntem tarafından geliştirilen katmanları içeren ve yol segmentasyonu yapabilen bir derin öğrenme modeli önerilmiştir. Geliştirilen bu katmanlar konvolüsyonel katmanların çıkışındaki öznelik haritalarından sonra uygulanarak, yol görüntülerinde yan yana birbirine yakın yol sınırlarını doğru bir biçimde tespit edebilmek için kullanılmıştır. Önerilen yöntem KITTI [4] veri seti üzerinde denenmiş ve %89,08 F1 skor başarımla çalıştığı ve bilgisayar ortamında 16 fps'te çalıştığı belirtilmiştir.

[34]'de önerilen yöntemde yol üzerindeki engellerin ve yol bölgelerinin tespit edilebilmesi için konvolüsyonel bir derin öğrenme modeli ve bir kayıp fonksiyonu önerilmiştir. Yöntemde tez kapsamında gerçekleştirilen yöntemle benzer bir şekilde görüntüler 5 piksel genişliğinde ince dikey şerit dilimlere halinde kesilmiş ve her bir görseldeki yolun bittiği 5 piksel değeri tespit edilmeye çalışılmıştır. Model çıkışında ise tahminleri tutarlı hale getirmek için koşullu rastgele alan (Conditional Random Field - CRF) yöntemi, ardından sürülebilir alanları oluşturmak ve başarımla artırmak için ise grafik kesimli segmentasyon (Graph-cut segmentation) yöntemi ile sürülebilir alanlar tespit edilmiştir. Önerilen yöntem KITTI [4] veri seti üzerinde eğitilmiş ve test edilmiştir. Test sonucunda %89 F1 skoru ile çalıştığı belirtilmiştir.

Tablo 2.5. Yakın zamanda önerilen derin öğrenme temelli şerit tespit yöntemlerinin karşılaştırılması

Yöntem	Ön İşlem	Özellik Çıkartımı	Son İşlem	Yaklaşım	Platform / Hız (fps)
(Poudel ve diğ., 2019) [20]	Fast-SCNN (2 konv., 3 Dskonv., 3 bottleneck, 1 PPM, 1 FFM), Softmax		—	Segmentasyon	Tensorflow / 485
(Chen ve diğ., 2020) [21]	FasterSeg (8 konv., 10 zoomed konv.), Softmax		—	Segmentasyon	Tensorflow / 163
(Emara ve diğ., 2019) [22]	LiteSeg, Encoder (MobileNetV2, DASPP Module, 1 konv., 1 upsample), Decoder (3 konv., 1 upsample)		—	Segmentasyon	PyTorch / 161
(Chen ve diğ., 2020) [24]	RPNNet, Encoder (13 bottleneck), Decoder (3 upsample)		—	Segmentasyon	Tensorflow / 125

Tablo 2.5. (Devam) Yakın zamanda önerilen derin öğrenme temelli şerit tespit yöntemlerinin karşılaştırılması

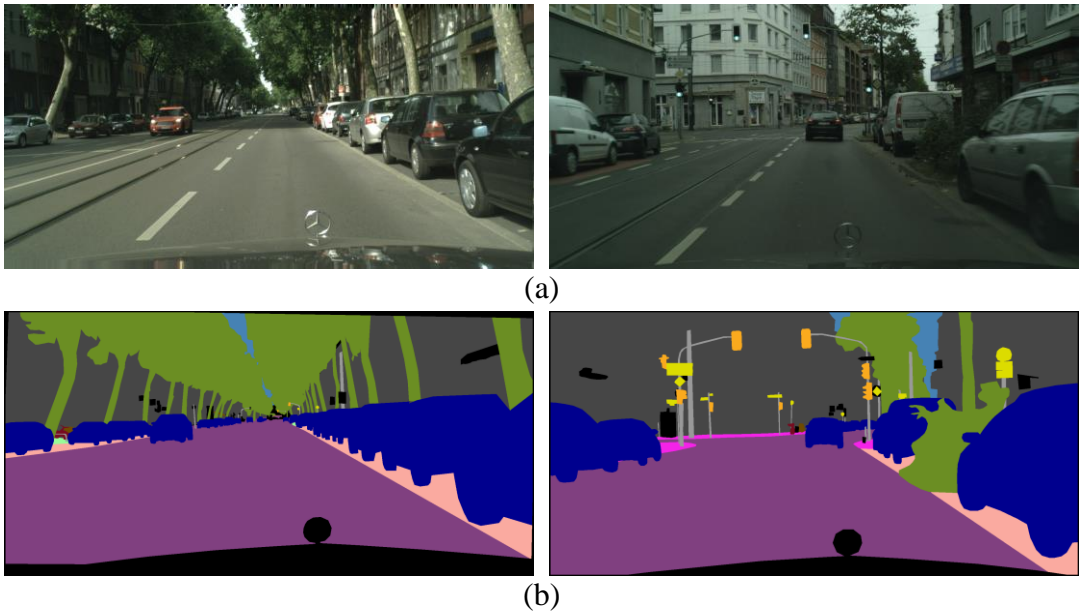
Yöntem	Ön İşlem	Özellik Çıkartımı	Son İşlem	Yaklaşım	Platform / Hız (fps)
(Mazzini ve diğ., 2019) [26]	Encoder (ERFNet, 3 downsampler, 13 bottleneck-1D), Decoder (iGUM upsampling)		—	Segmentasyon	PyTorch / 113
(Mehta ve diğ., 2017) [29]	ESPNet, Encoder (1 konv, 4 ESP), Decoder (3 dekonv, 1 ESP), softmax		—	Segmentasyon	PyTorch / 112
(Lo ve diğ., 2019) [30]	EDANet, Encoder (3 downsampling, 2 EDA Blok), Decoder (1 konv.)		Bilateral interpolation	Segmentasyon	PyTorch / 108
(Wang ve diğ., 2017) [31]	Fast contour detection	Decoder (RGB görsel için 7 konv., contour için 7 konv.), decoder (2 konv., 2 dekonv.)	CRF, Graph-cut	Segmentasyon	Caffe / 2,5
(Liu ve diğ., 2018) [32]	Rastgele görüntü kırpma	112 katmanlı	—	Segmentasyon	Caffe / —
(Lyu ve diğ., 2019) [33]	13 katmanlı (10 konv., 2 D-LSTM, 1 upsample)		—	Segmentasyon	Tensorflow, Keras / 16
(Levi ve diğ., 2015) [34]	Görüntü kırpma	5 katmanlı (2 konv., 3 FCL)	—	Regresyon	Caffe / —

3. TEZ KAPSAMINDA GERÇEKLEŞTİRİLEN SÜRÜLEBİLİR ALANLARIN TESPİTİ ÇALIŞMASI

Bu bölümde tez kapsamında geliştirilen sürülebilir alanların regresif çıkışlı konvolüsyonel sinir ağı ile tespiti yöntemi detaylı bir şekilde anlatılacaktır. Önerilen yöntemin başarılı bir şekilde, hem bilgisayar hem de gömülü ortam üzerinde yüksek hızlarla çalışması amaçlanmıştır.

3.1. Sürülebilir Alan Tespiti Performansının Değerlendirilmesi

Önerilen yöntem kapsamında sürülebilir alanların tespit performansının değerlendirilmesi için CityScapes [1] veri setinden yararlanılmıştır. CityScapes veri setinde 2975 train, 500 validation ve 1525 test ile toplamda 5000 adet görüntü yer almaktadır. Görüntüler 50 farklı şehirde, farklı hava ve mevsim koşullarında toplanmıştır. Toplanan görüntüler 30 farklı sınıf etiketi ile piksel bazlı etiketlenmiştir. Önerilen yöntemde sadece sürülebilir alanların tespiti amaçlandığından etiketli piksellerden sadece yol etiketine sahip pikseller dikkate alınmıştır. CityScapes veri setinden örnek bir görüntü ve piksel bazlı yer doğrusu etiketleri Şekil 3.1’de gösterilmiştir.



Şekil 3.1. Veri seti örnek görüntüsü (a) Giriş görüntüsü (b) Yer doğrusu haritası [1]

Sürülebilir alan tespit yöntemlerinin başarımı yer doğrusunda yol olarak etiketlenen her bir pikselin yöntem tarafından da yol olarak tespit edilip edilmediğine bakılarak ölçülmektedir. CityScapes [1] veri setinde de performansı değerlendirmek için bu yapıyı gerçekleyen PASCAL VOC birleşim üzerinden kesişme (Intersection over union - IoU) olarak bilinen standart Jaccard indeksi kullanılmaktadır [35]. Birleşim üzerinden kesişme değeri Eşitlik (3.1)'de verilen formül ile hesaplanmaktadır. Formülde TP, FP ve FN değerleri sırasıyla yol olarak etiketlenen pikselin yol olarak tespit edilmesi durumunu (True Positive), yol olarak etiketlenmeyen pikselin yol olarak tespit edilmesi durumunu (False Positive), yol olarak etiketlenmiş pikselin yol değil olarak tespit edilmesi durumunu (False Negative) göstermekte ve test setindeki toplam sayılarını temsil etmektedir.

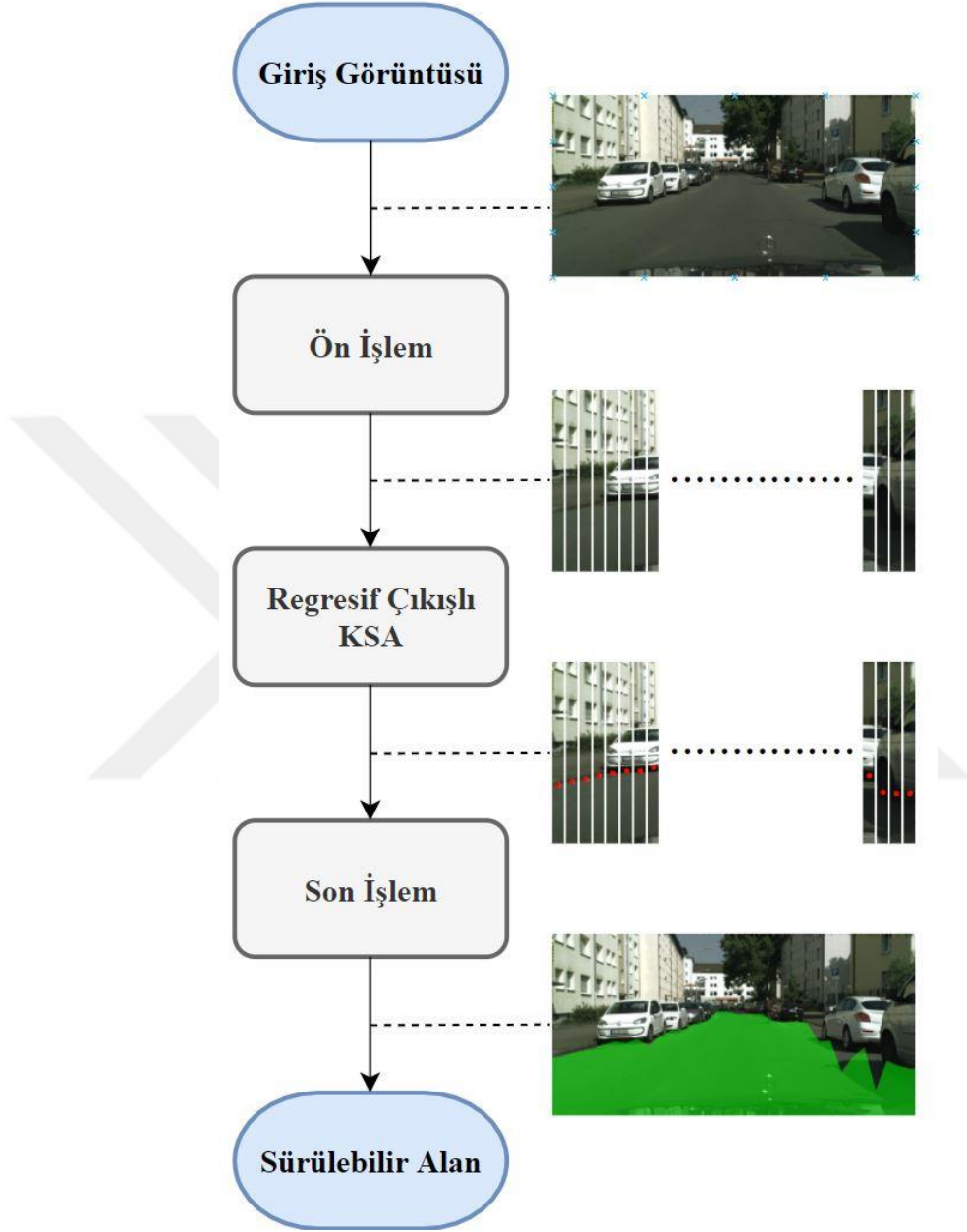
$$IoU = \frac{TP}{TP + FP + FN} \quad (3.1)$$

Tez kapsamında önerilen yöntemin performans karşılaştırmasını CityScapes veri setinin internet sitesinde yer alan ve herkese açık olan kıyaslama (benchmark) tablosunda ve literatürde yer alan en hızlı algoritmalar kullanılarak yapılmıştır.

3.2. Regresif Çıkışlı Konvolüsyonel Sinir Ağı ile Gerçek Zamanlı Sürülebilir Alan Tespiti

Bu çalışmada gerçekleştirilen sürülebilir alan tespiti yönteminde konvolüsyonel sinir ağı temelli bir derin öğrenme modeli kullanılmaktadır. Önerilen yöntem literatürde sıklıkla kullanılan encoder-decoder mimarisine sahip segmentasyon temelli modeller yerine regresif bir çıkış veren ve göreceli olarak daha basit bir mimariye sahip konvolüsyonel bir sinir ağı içermektedir. Görüntüler yöntemin gerçekleştirilebilmesi ve regresif çıkışlı derin öğrenme modeline girdi olarak verilebilmesi için bir ön işleme tabi tutulmuştur. Ön işlemde orijinal renkli görüntüler ince dikey şeritler halinde kesilmiştir. Kesilen her bir dikey şerit görseli derin öğrenme modeline girdi olarak verilmiş ve görüntünün alt noktasından başlayarak yolun sonlandığı piksel değerinin tespit edilmesi amaçlanmıştır. Son işlem olarak tespit edilen bütün noktalar birleştirilerek sürülebilir alanlar çıkartılmıştır. Geliştirilen yöntem farklı orijinal görüntü ve dikey şerit boyutlarında denenmiş ve gömülü platformlar üzerinde

çalıştırılmıştır. Önerilen yöntemin aşamaları ve akış diyagramı Şekil 3.2’de gösterilmektedir.

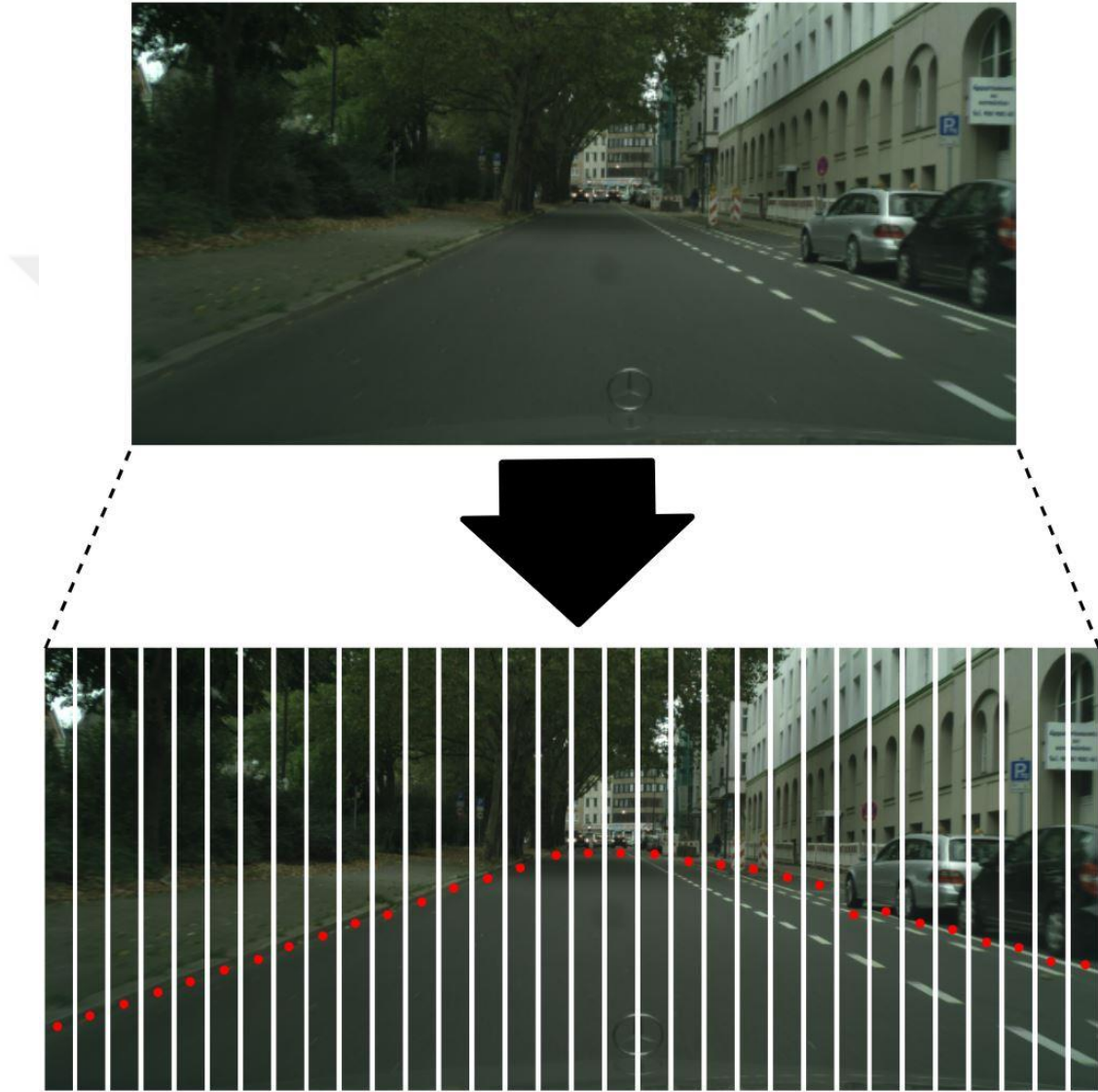


Şekil 3.2. Önerilen yöntemin akış diyagramı

3.2.1. Ön işlem

Önerilen yöntemde sürülebilir alanların tespiti için CityScapes [1] veri setinde bulunan görüntüler ince dikey şeritler halinde kesilerek kullanılmaktadır. Çıkartılan her bir ince dikey şerit görselinde ise görüntünün alt kısmından başlayarak yolun bittiği piksel değeri, görüntünün yükseklik değerine normalize edilerek o görsel için yer doğrusu

yani yolun bittiği sınıf piksel değeri elde edilmekte ve eğitim için bu veriler kullanılmaktadır. Modelin görsel üzerinden sonuç üretmesi aşamasında ise ön işlem kısmında görseller ince dikey şeritler halinde kesilmekte ve ard arda dizilerek modele giriş verisi olarak hazırlanmaktadır. Örnek bir yol görüntüsünde uygulanan ön işlem Şekil 3.3'te gösterilmektedir.



Şekil 3.3. Ön işlem çıkışında oluşan ince dikey şeritler

Hız ve başarımda oluşacak değişimi incelemek için önerilen yöntemde kullanılan derin öğrenme modeli 9 farklı giriş boyutu için eğitilmiştir. Bu giriş boyutları CityScapes [1] veri setinin orijinal görüntü boyutlarının yanında 1/2 ve 1/4 kat olacak şekilde alt örneklenerek ve orijinal ve alt örneklenmiş her bir boyut için ince dikey şeritlerin genişlikleri 64, 32 ve 16 piksel olacak şekilde yeni veri setleri oluşturulmuştur.

Yöntem sonuç çıkarma aşamada da ön işlem gerçekleştirirken orijinal boyuttaki giriş görüntülerini derin öğrenme modeline girecek boyutlara uyacak şekilde alt örnekleme yapılmıştır. Oluşturulan veri setlerinin görüntü boyutları ve görüntü sayıları Tablo 3.1’de detaylı olarak verilmiştir.

Tablo 3.1. Orijinal ve ön işlem sonrası oluşan görüntü adetleri ve boyutları

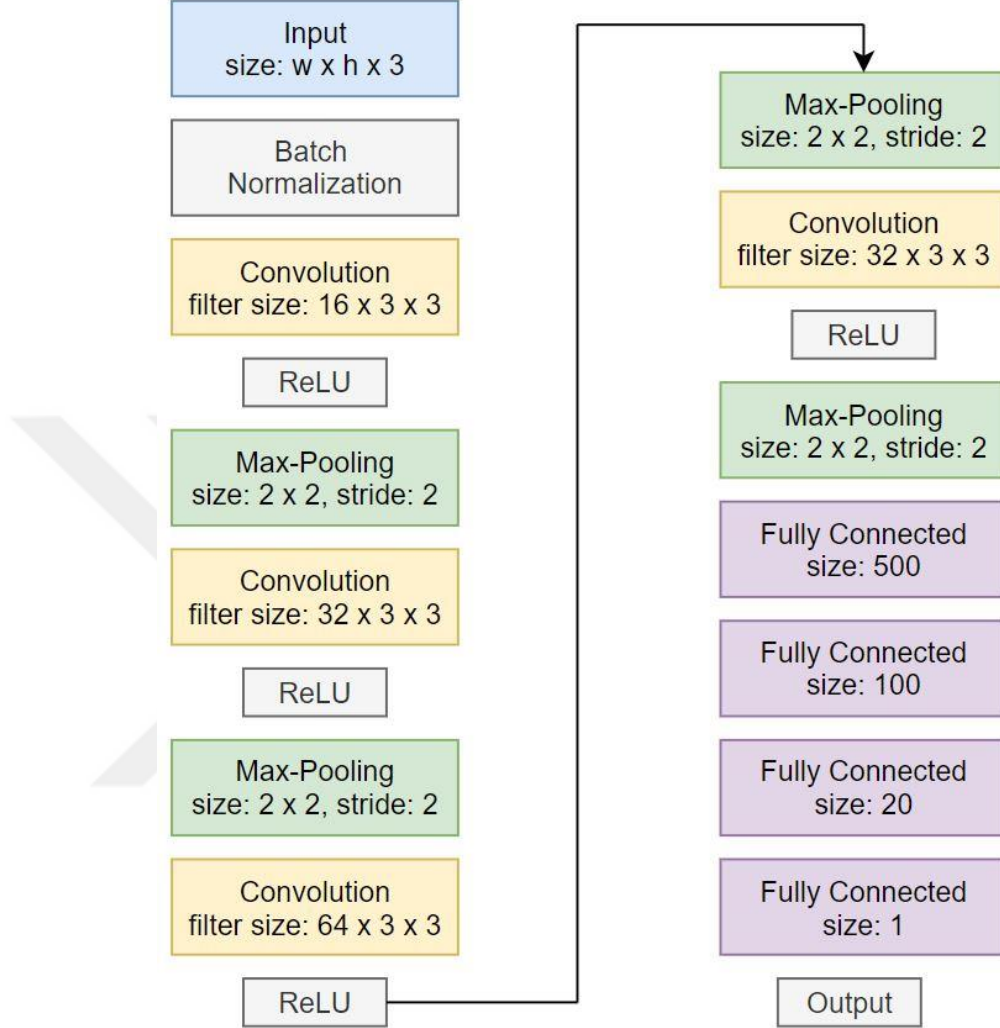
CityScapes [1]		Ön İşlem Sonrası Veri Seti	
Görüntü Boyutları (W×H)	Görüntü Sayısı (Adet)	Görüntü Boyutları (W×H)	Görüntü Sayısı (Adet)
2048 × 1024	5000	64 × 1024	160000
		32 × 1024	320000
		16 × 1024	640000
2048 × 1024 (Ölçekleme Oranı: 1/2)	5000	64 × 512	80000
		32 × 512	160000
		16 × 512	320000
2048 × 1024 (Ölçekleme Oranı: 1/4)	5000	64 × 256	40000
		32 × 256	80000
		16 × 256	160000

3.2.2. Konvolüsyonel sinir ağı modeli

Önerilen yöntemde sürülebilir alanların tespiti için geliştirilen konvolüsyonel sinir ağı temelli derin öğrenme modeli Şekil 3.4’te gösterilmiştir. Modele giriş görüntüsü olarak orijinal görüntülerden kesilmiş renkli ince dikey şeritler verilmektedir. Model ile çözülmeye çalışılan problem ise her bir ince dikey şerit görüntüsünde en alt noktadan başladığında yol sınırının bittiği ilk pikselin tespit edilmesidir. Model son katman olarak ise lineer regresif bir çıkış veren katmana sahiptir. Bu çıkış katmanında elde edilen değer, görüntünün yükseklik piksel değeri ile çarpılarak ilgili görsel için tespit edilen yol sınırının piksel değerini vermektedir.

Geliştirilen derin öğrenme modelinin gömülü ortamlarda gerçek zamanlı çalıştırılabilmesi, az parametre içermesi ve düşük karmaşıklık içermesi hedeflenmiştir. Bu nedenle model ilk 4 katmanı konvolüsyonel ve son 4 katmanı tam bağlı katman olmak üzere 8 katmandan oluşmaktadır. Çıkış katmanı hariç her bir katman aktivasyon fonksiyonu olarak doğrultulmuş doğrusal birim (Rectified Linear Unit - ReLU) ve her konvolüsyonel katmanın sonrasında da maksimum-havuzlama (Max-pooling) katmanı kullanılmaktadır. Tam bağlı katmanlarda bulunan nöron sayıları ise sırasıyla 500, 100

ve 20 adettir. Son çıkış katmanında ise tek nöronlu lineer regresif bir çıkış veren bir yapı ile sonuç elde edilmektedir.

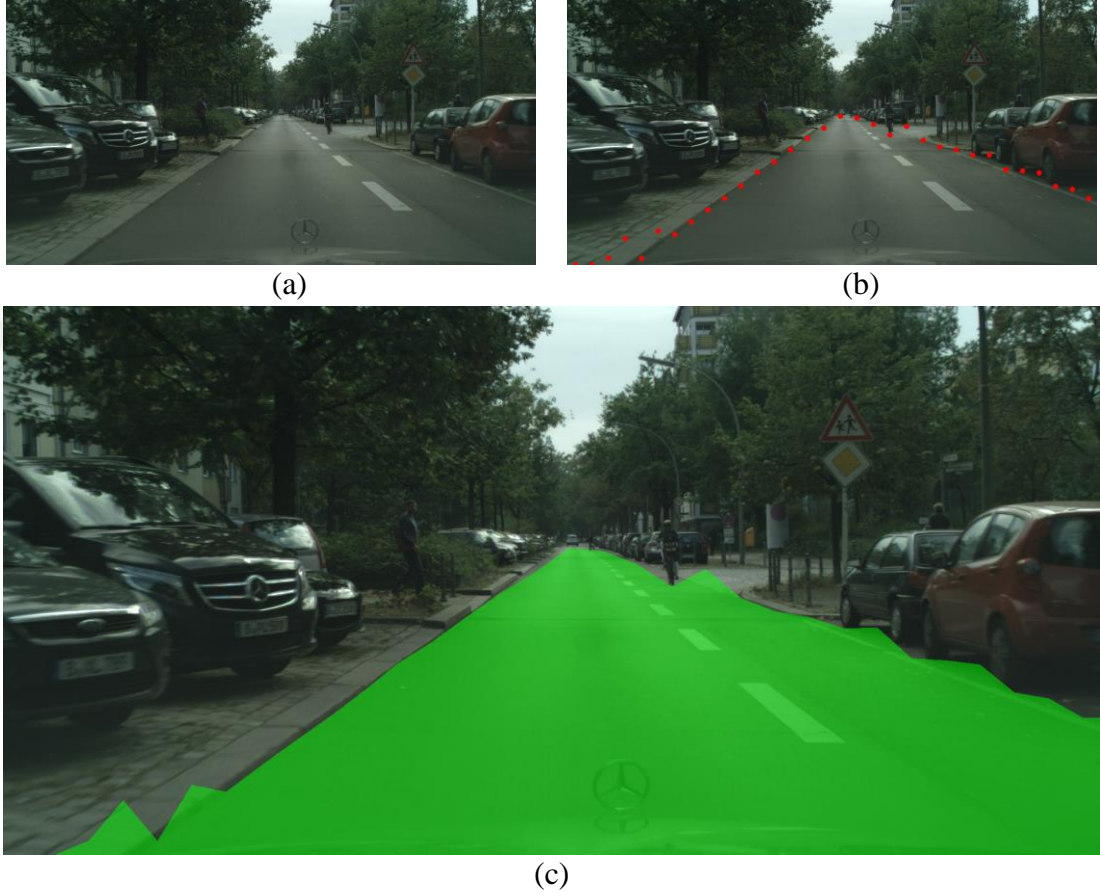


Şekil 3.4. Sürülebilir alanların tespiti için oluşturulan KSA modeli

3.2.3. Son işlem

Bir önceki aşamada orijinal görüntüden kesilmiş her bir ince dikey şerit görseli için yolun bittiği sınır noktaları derin öğrenme modeli tarafından tespit edilmektedir. Son işlem aşamasında ise orijinal görüntü üzerinde sürülebilir alanların çizdirilebilmesi için her bir ince dikey şerit görselinin konumu ve ilgili görsel için tespit edilen yol sınır noktası orijinal görüntü üzerinde işaretlenmektedir. Modelin çıkışında normalize edilmiş bir değer ile yolun bittiği piksel elde edildiği için alt örneklenmiş veriler ile eğitilmiş derin öğrenme modellerinden elde edilen değerler de alt örnekleme oranı bilindiği için orijinal boyuttaki görüntüler üzerinde kullanılabilir. Orijinal

görüntü üzerinde noktaların her biri işaretlendikten sonra ise yan yana olan noktalar birleştirilerek oluşan eğrinin altında kalan alan sürülebilir alan olarak etiketlenmiştir. Noktaların orijinal görüntü üzerinde işaretlenmiş ve sonrasında sürülebilir alanların çizdirilmiş sonuçları Şekil 3.5’te gösterilmektedir.

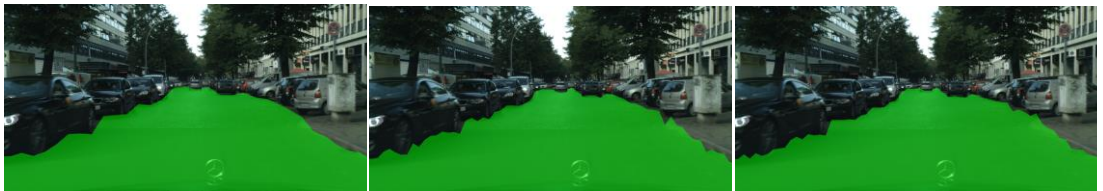


Şekil 3.5. Önerilen yöntemin sürülebilir alan tespiti sonuçları (a) Orijinal görüntü (b) Derin öğrenme modeli tarafından her bir ince şerit dilimi görüntüsünde tespit edilen noktalar (c) Orijinal görüntü üzerinde tespit edilen noktalar ile çıkartılmış sürülebilir alan sonucu

4. DENEYSEL SONUÇLAR

Önerilen yöntemin gerçekleştirilmesi için oluşturulan veri setleri Intel i9-7900X ve Nvidia TITAN Xp ekran kartına sahip bir bilgisayarda, tensorflow, keras platformu kullanılarak eğitilmiştir. Eğitilen modeller CityScapes [1] veri setinin test görüntüleri üzerinde çalıştırılarak sürülebilir alanlar tespit edilmiş ve başarımların hesaplanması için sonuçlar internet üzerindeki CityScapes kıyaslama (benchmark) sunucusuna gönderilmiştir. Kıyaslama (benchmark) sunucusundan geri dönüş yapılan sonuçlarda yol etiketine ait başarımlar dikkate alınmıştır. Kıyaslama sunucusunda sürülebilir alanların tespit performansının değerlendirilmesi ise Bölüm 3.1’de açıklanan birleşim üzerinden kesişme (Intersection over union - IoU) ölçütü kullanılarak gerçekleştirilmektedir..

Şekil 4.1, Şekil 4.2, Şekil 4.3, Şekil 4.4 ve Şekil 4.5’da önerilen yöntemin farklı model giriş boyutlardaki ince dikey şeritler ile eğitilmiş versiyonlarının CityScapes test setinde bulunan farklı görseller üzerinde tespit sonuçları gösterilmektedir. Görüntüler üzerinde yeşil olarak işaretlenen alanlar yöntem tarafından tespit edilen sürülebilir alan sonuçlarını göstermektedir. Görsellerin en alt kısmında yer alan aracın gövdesi CityScapes [1] veri setinin başarımlar hesaplama aşamasında dikkate alınmamıştır. Görseller incelendiğinde önerilen yöntemin farklı giriş boyutlarında başarılı sonuçlar verdiği görülmektedir.

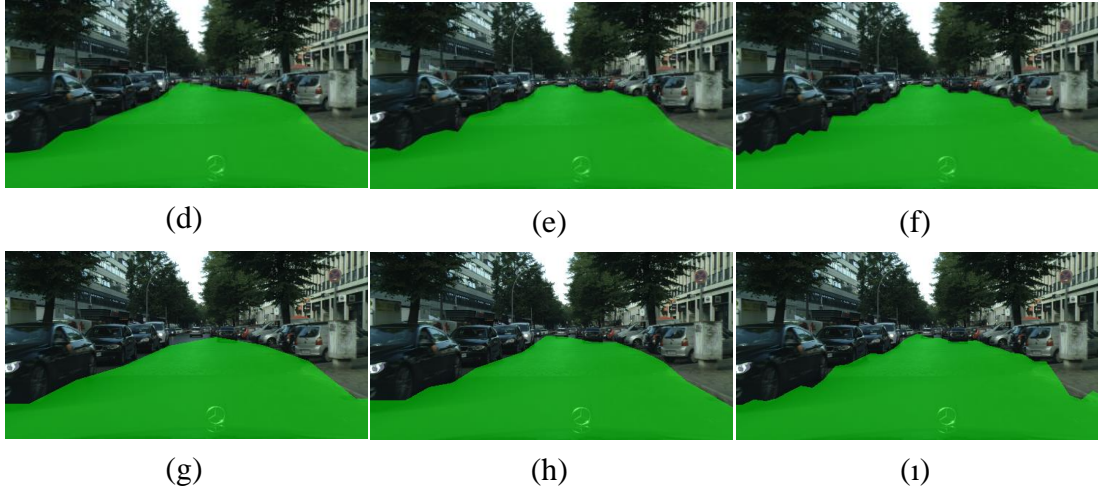


(a)

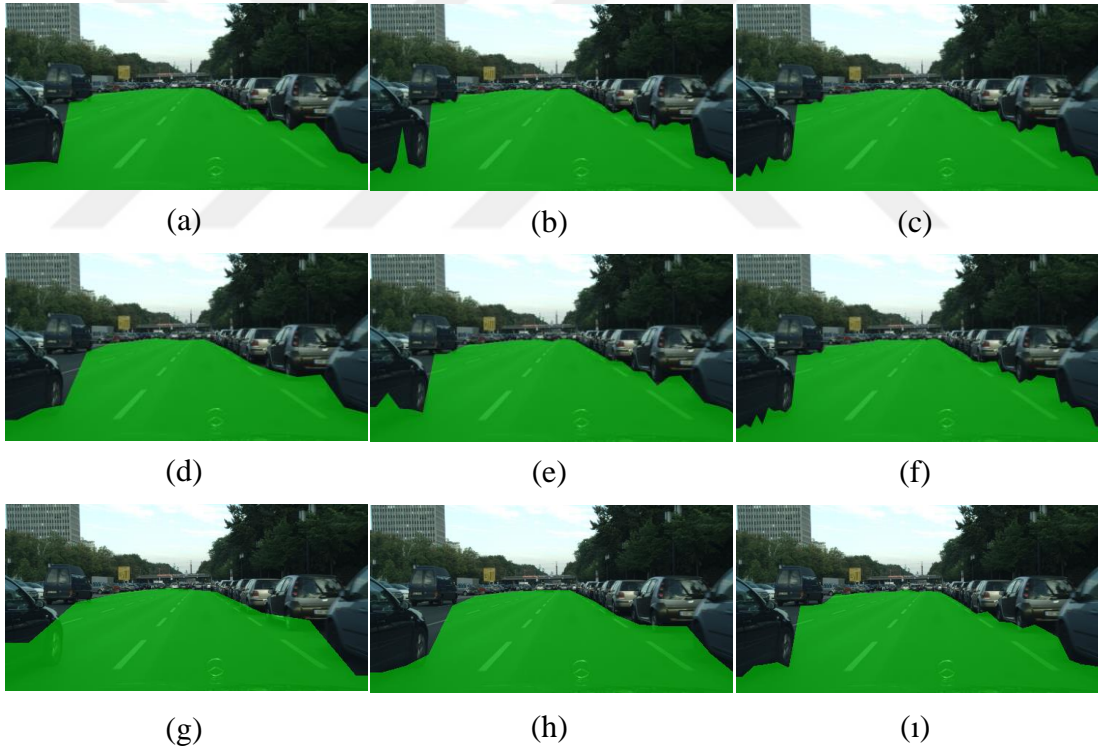
(b)

(c)

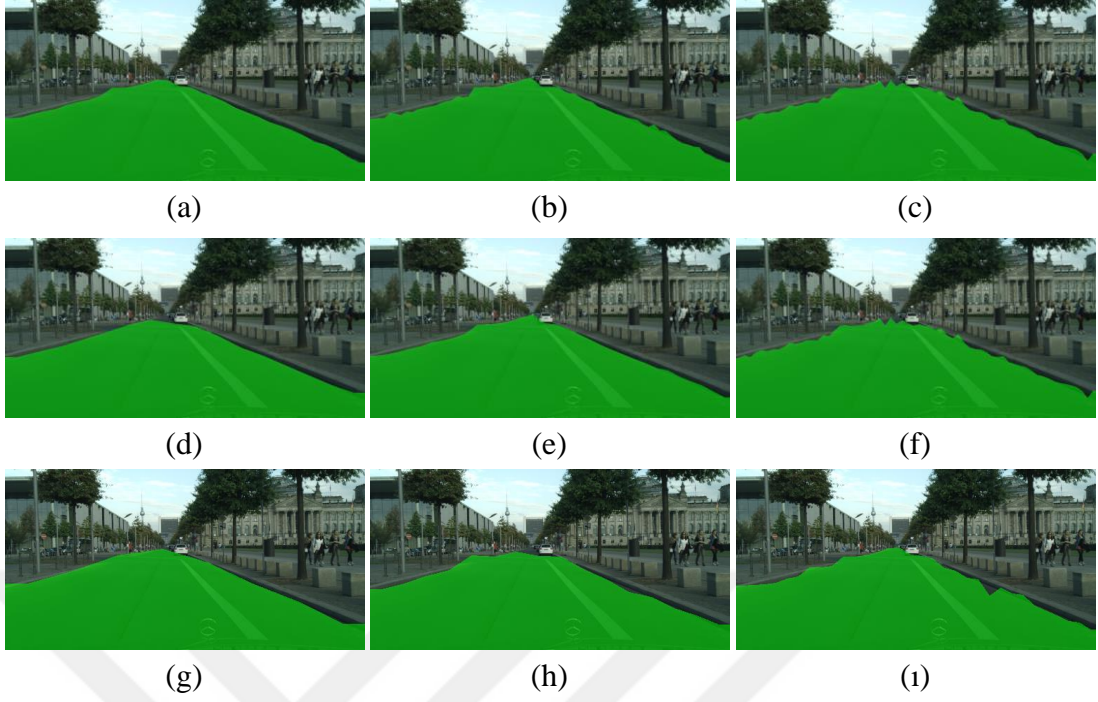
Şekil 4.1. Örnek bir görüntü üzerinde önerilen yöntemin derin öğrenme modelinin farklı giriş boyutlardaki sonuçları (a) 64×1024 piksel (b) 32×1024 piksel (c) 16×1024 piksel (d) 64×512 piksel (e) 32×512 piksel (f) 16×512 piksel (g) 64×256 piksel (h) 32×256 piksel (i) 16×256 piksel



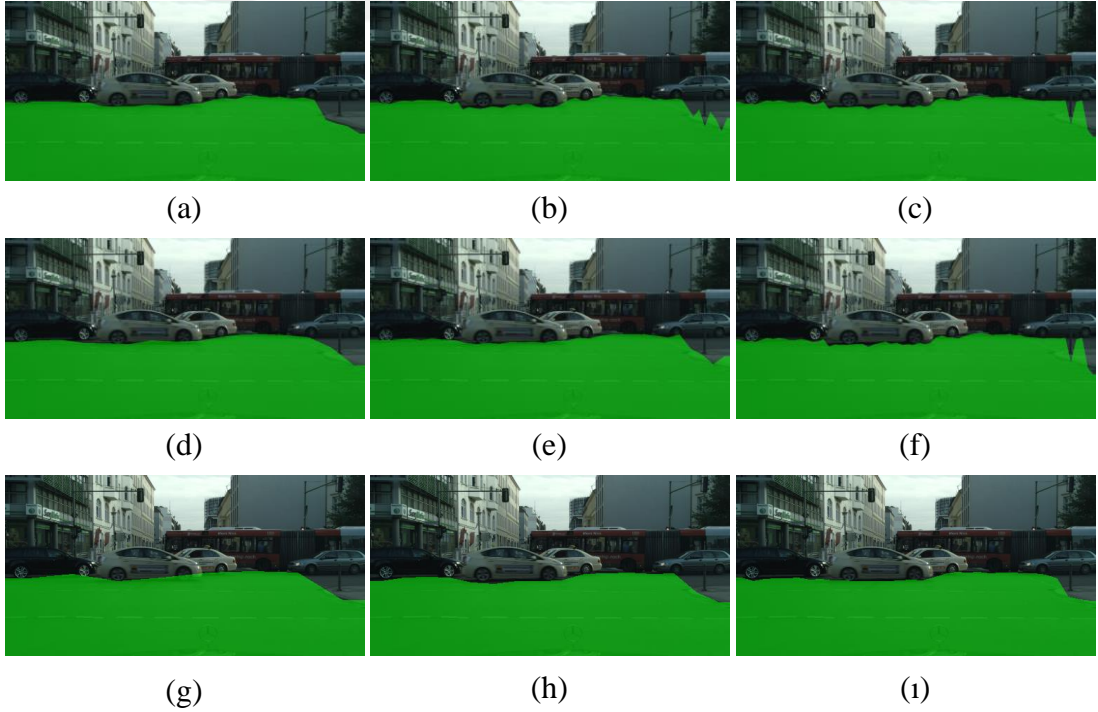
Şekil 4.1. (Devam) Örnek bir görüntü üzerinde önerilen yöntemin derin öğrenme modelinin farklı giriş boyutlardaki sonuçları (a) 64×1024 piksel (b) 32×1024 piksel (c) 16×1024 piksel (d) 64×512 piksel (e) 32×512 piksel (f) 16×512 piksel (g) 64×256 piksel (h) 32×256 piksel (i) 16×256 piksel



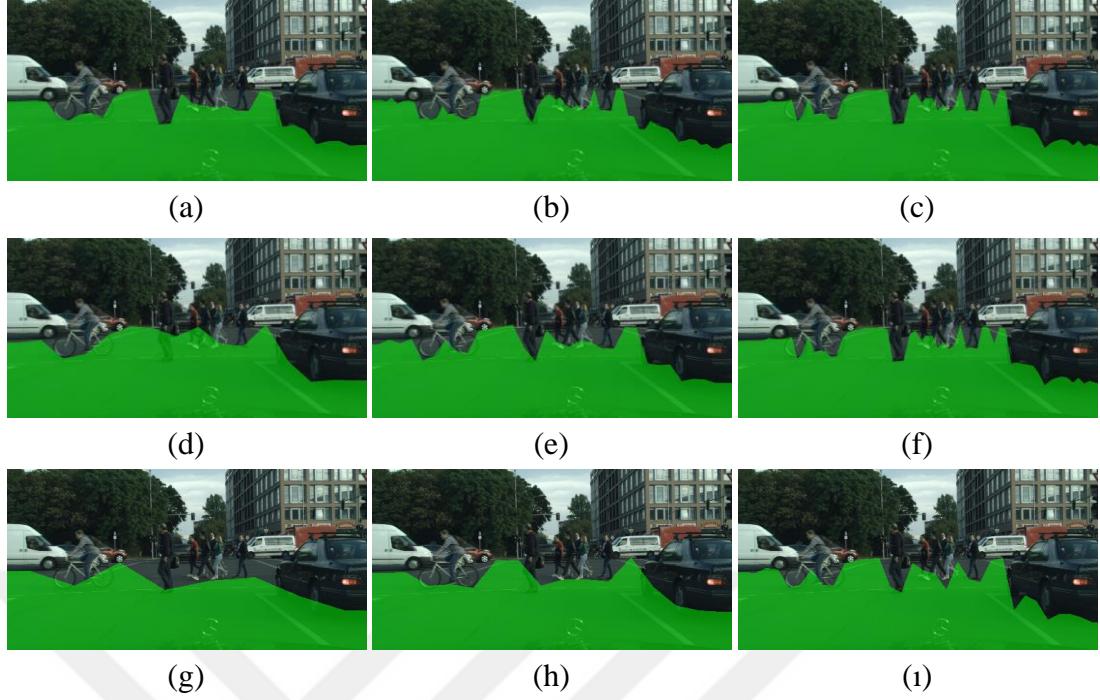
Şekil 4.2. Örnek bir görüntü üzerinde önerilen yöntemin derin öğrenme modelinin farklı giriş boyutlardaki sonuçları (a) 64×1024 piksel (b) 32×1024 piksel (c) 16×1024 piksel (d) 64×512 piksel (e) 32×512 piksel (f) 16×512 piksel (g) 64×256 piksel (h) 32×256 piksel (i) 16×256 piksel



Şekil 4.3. Örnek bir görüntü üzerinde önerilen yöntemin derin öğrenme modelinin farklı giriş boyutlardaki sonuçları (a) 64×1024 piksel (b) 32×1024 piksel (c) 16×1024 piksel (d) 64×512 piksel (e) 32×512 piksel (f) 16×512 piksel (g) 64×256 piksel (h) 32×256 piksel (i) 16×256 piksel



Şekil 4.4. Örnek bir görüntü üzerinde önerilen yöntemin derin öğrenme modelinin farklı giriş boyutlardaki sonuçları (a) 64×1024 piksel (b) 32×1024 piksel (c) 16×1024 piksel (d) 64×512 piksel (e) 32×512 piksel (f) 16×512 piksel (g) 64×256 piksel (h) 32×256 piksel (i) 16×256 piksel



Şekil 4.5. Örnek bir görüntü üzerinde önerilen yöntemin derin öğrenme modelinin farklı giriş boyutlardaki sonuçları (a) 64×1024 piksel (b) 32×1024 piksel (c) 16×1024 piksel (d) 64×512 piksel (e) 32×512 piksel (f) 16×512 piksel (g) 64×256 piksel (h) 32×256 piksel (i) 16×256 piksel

Tablo 4.1’de farklı giriş boyutlarında eğitilmiş modellerin CityScapes veri setinin test görüntüleri üzerinde gösterdiği hız ve başarımlar gösterilmektedir. Önerilen yöntemde eğitilen modellerin çalışma hızına artırmak için model dosyaları üzerine tensorflow tensorRT optimizasyonu uygulanmıştır. Optimizasyon aşamasında model ağırlık dosyalarında floating point 32 (FP32) olarak tutulan değişkenler floating point 16 (FP16) formatında dönüştürülerek kaydedilmiş ve modelin çıkarım aşamasında başarımı etkilemeden işlem yükü azaltılmıştır. TensorRT optimizasyonu sonrasında elde edilen başarımlar ve hız değerleri Tablo 4.2’de gösterilmektedir.

Tablo 4.1. Farklı giriş boyutundaki modellerin cityScapes test seti üzerindeki başarımlar ve hız değerleri

Görüntü Boyutları (W×H)	İnce Şerit Genişliği	IOU (%)	İnce Şerit Sayısı	Zaman (s)	FPS
2048 × 1024	64	92,55	32	0,04452	22
	32	91,99	64	0,05189	19
	16	91,47	128	0,05426	18
2048 × 1024 (Ölçekleme Oranı: 1/2)	64	91,92	16	0,01184	84
	32	91,90	32	0,01179	85
	16	91,33	64	0,01265	79
2048 × 1024 (Ölçekleme Oranı: 1/4)	64	90,50	8	0,00538	186
	32	90,96	16	0,00556	180
	16	91,05	32	0,00585	171

Tablo 4.2. Farklı giriş boyutundaki modellerin tensorRT dönüşümü sonrası cityScapes test seti üzerindeki başarımlar ve hız değerleri

Görüntü Boyutları (W×H)	İnce Şerit Genişliği	IOU (%)	İnce Şerit Sayısı	Zaman (s)	FPS
2048 × 1024	64	92,55	32	0,03827	26
	32	91,99	64	0,04325	23
	16	91,47	128	0,047619	21
2048 × 1024 (Ölçekleme Oranı: 1/2)	64	91,92	16	0,00855	117
	32	91,90	32	0,00874	114
	16	91,33	64	0,00930	107
2048 × 1024 (Ölçekleme Oranı: 1/4)	64	90,50	8	0,00421	237
	32	90,96	16	0,00425	235
	16	91,05	32	0,00432	231

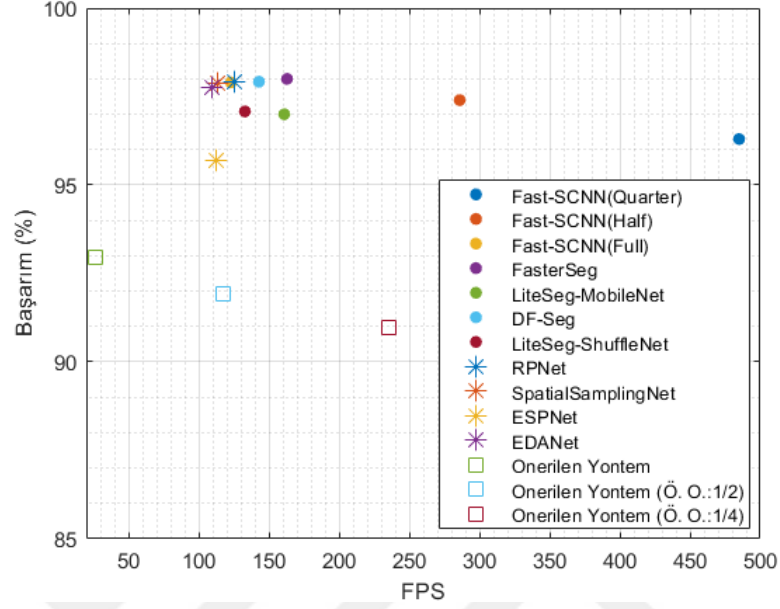
Tablo 4.2 incelendiğinde önerilen yöntemin farklı model giriş çözünürlüklerinde yüksek başarımlar elde ettiği görülmektedir. Düşük çözünürlüklü model giriş görüntülerinde ise yüksek çözünürlükte elde edilen başarımdan fazla ödün vermeden çok yüksek fps değerlerinde çalışabildiği görülmektedir. Tablo 4.3'te önerilen yöntem literatürde bulunan ve CityScapes kıyaslama (benchmark) tablosunda yer alan en yüksek hız ve başarımlara sahip 11 adet yöntem ile karşılaştırılmaktadır.

Tablo 4.3. Sürülebilir alan tespit yöntemlerinin CityScapes test veri seti üzerinde performans değerlendirmesi

Yöntem	Çözünürlük	FPS	Başarımlar (%)	Platform	Yayın Yılı
Fast-SCNN (Quarter) [20]	512 × 256	485	96,3	TITAN Xp	Kasım, 2018
Fast-SCNN (Half) [20]	1024 × 512	286	97,4	TITAN Xp	Kasım, 2018
Fast-SCNN (Full) [20]	2048 × 1024	123	97,9	TITAN Xp	Kasım, 2018
FasterSeg [21]	256 × 128	163	98	1080Ti, TensorRT	Kasım, 2019
LiteSeg-MobileNet [22]	640 × 360	161	97	1080Ti	Ocak, 2019
DF-Seg [36]	256 × 128	143	97,92	1080Ti	Kasım, 2018
LiteSeg-ShuffleNet [22]	1536 × 768	133	97,08	1080Ti	Ocak, 2019
RPNNet [24]	1024 × 512	125	97,9	1080Ti	Aralık, 2018
Spatial Sampling Net [26]	1024 × 512	113	97,89	TITAN Xp	Kasım, 2018
ESPNet [29]	512 × 512	112	95,68	TITAN Xp	Ocak, 2018
EDANet [30]	1024 × 512	109	97,76	1080Ti	Ağustos, 2018
Önerilen Yöntem	2048 × 1024	26	92,55	TITAN Xp, TensorRT	—
Önerilen Yöntem (Örnekleme Oranı: 1/2)	2048 × 1024	117	91,92	TITAN Xp, TensorRT	—
Önerilen Yöntem (Örnekleme Oranı: 1/4)	2048 × 1024	231	90,96	TITAN Xp, TensorRT	—

Tablo 4.3'te yer alan yöntemlerin hız ve başarımlar grafiği halinde gösterimi Şekil 4.6'da gösterilmiştir. Önerilen yöntem, karşılaştırılan yöntemler arasında en hızlı üçüncü yöntem, yüksek çözünürlükte olanlar arasında ise birinci yöntem konumundadır. Yöntem doğası gereği ince dikey şerit görsellerinde yolun bittiği tek bir noktayı tespit etmesi ve son aşamada her bir ince dikey şerit görselinde tespit edilen noktaların birleştirilmesi nedeniyle segmente edilen yol bölgesinin sınırları detaylı bir şekilde tespit edilememektedir. Ayrıca veri setinde sürülebilir yol olmayan yan yollar da yol etiketi ile etiketlenmiştir fakat önerilen yöntem sürülebilir alan olmadığı için bu

bölgeleri tespit edememektedir. Bu iki sebepten ötürü önerilen yöntemin başarımı bir diğer yöntemlere göre bir miktar geride kalmaktadır.

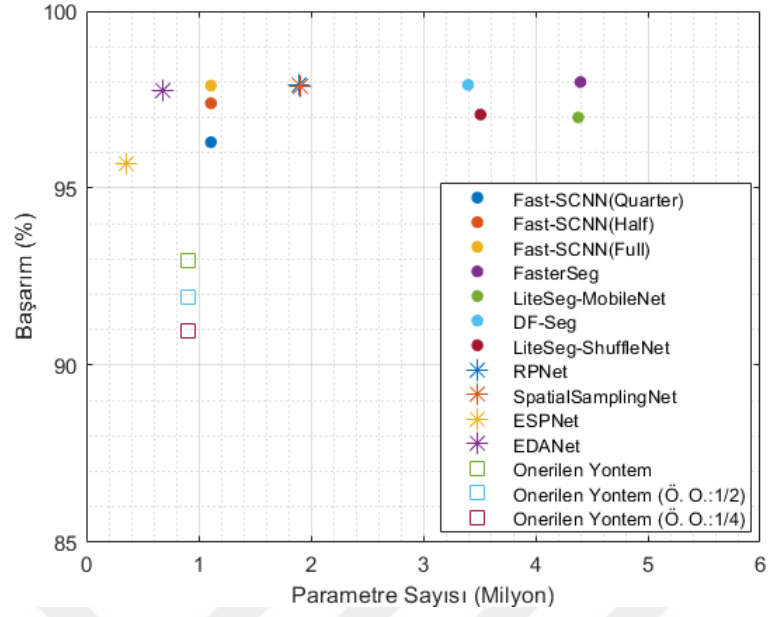


Şekil 4.6. Karşılaştırılan yöntemlerin FPS-Başarım Grafiği

Önerilen yöntem az parametre ve karmaşıklık içeren bir derin öğrenme modeline sahip olması ve düşük çözünürlükte de yüksek başarımla çalışabilmesi nedeniyle gömülü platformlar üzerinde çalışmada uygundur. Tablo 4.4'te önerilen yöntem ve karşılaştırılan diğer yöntemlerin parametre sayısı-başarım değerleri verilmektedir. Şekil 4.7'de ise bu parametre sayısı-başarımlar grafik üzerinde gösterilmiştir.

Tablo 4.4. Önerilen yöntem ve karşılaştırılan yöntemlerin parametre-başarım tablosu

Yöntem	Başarım (%)	Parametre Sayısı (Milyon)
Fast-SCNN (Quarter) [20]	96,3	1,11
Fast-SCNN (Half) [20]	97,4	1,11
Fast-SCNN (Full) [20]	97,9	1,11
FasterSeg [21]	98	4,4
LiteSeg-MobileNet [22]	97	4,38
DF-Seg [37]	97,92	3,4
LiteSeg-ShuffleNet [22]	97,08	3,51
RPNNet [24]	97,9	1,89
Spatial Sampling Net [26]	97,89	1,9
ESPNet [29]	95,68	0,35
EDANet [30]	97,76	0,68
Önerilen Yöntem	92,55	0,9
Önerilen Yöntem (Örnekleme Oranı: 1/2)	91,92	0,9
Önerilen Yöntem (Örnekleme Oranı: 1/4)	90,96	0,9



Şekil 4.7. Karşılaştırılan yöntemlerin Parametre-Başarım Grafiği

Önerilen yöntem Tablo 4.5'te sistem özellikleri verilen NVIDIA Jetson Nano ve NVIDIA Jetson Xavier NX gömülü platformları üzerinde de çalıştırılıp çalışma süreleri ve başarımları elde edilmiştir.

Tablo 4.5. Gömülü platform özellikleri

Özellik	Jetson Nano	Jetson Xavier NX
CPU	Quad-core ARM Cortex-A57	6-core NVIDIA Carmel ARM v8.2
GPU	NVIDIA Maxwell architecture with 128 NVIDIA CUDA Cores	NVIDIA Volta architecture with 384 NVIDIA CUDA Cores
RAM	4GB DDR4	8 GB DDR4
OS	Linux Debian	Linux Debian

Eğitilen her bir model bilgisayar ortamında olduğu gibi Jetson Nano ve Jetson Xavier NX cihazları üzerinde tensorRT optimizasyonu ile optimize edilmiş ve CityScapes test veri seti üzerinde çalıştırılarak sonuçlar elde edilmiştir. Cihazlar üzerinde elde edilen her bir sonuç aynı bilgisayar ortamında olduğu gibi CityScapes karşılaştırma (benchmark) sunucusuna yüklenerek sürülebilir alan tespiti başarımları elde edilmiştir. Tespit başarımları iki cihaz için de bilgisayar ortamından farklı olmamak ile birlikte her bir modelin başarımları ve FPS değerleri Tablo 4.6 ve 4.7'de gösterilmektedir. Tablolarda görüldüğü üzere önerilen sürülebilir alan tespiti yöntemi gömülü sistemler üzerinde de gerçek zamanlı çalışma yetisine sahiptir.

Tablo 4.6. Jetson Nano üzerinde farklı giriş boyutundaki modellerin CityScapes test seti üzerindeki başarımlar ve hız değerleri

Görüntü Boyutları (W×H)	İnce Şerit Genişliği	IOU (%)	İnce Şerit Sayısı	Zaman (s)	FPS
2048 × 1024	64	92,55	32	0,36630	3
	32	91,99	64	0,38167	3
	16	91,47	128	0,41152	3
2048 × 1024 (Ölçekleme Oranı: 1/2)	64	91,92	16	0,08695	12
	32	91,90	32	0,08748	11
	16	91,33	64	0,08795	11
2048 × 1024 (Ölçekleme Oranı: 1/4)	64	90,50	8	0,02648	38
	32	90,96	16	0,02712	37
	16	91,05	32	0,02745	36

Tablo 4.7. Jetson Xavier NX üzerinde farklı giriş boyutundaki modellerin tensorRT dönüşümü sonrası CityScapes test seti üzerindeki başarımlar ve hız değerleri

Görüntü Boyutları (W×H)	İnce Şerit Genişliği	IOU (%)	İnce Şerit Sayısı	Zaman (s)	FPS
2048 × 1024	64	92,55	32	0,11086	9
	32	91,99	64	0,13106	8
	16	91,47	128	0,13404	7
2048 × 1024 (Ölçekleme Oranı: 1/2)	64	91,92	16	0,03095	32
	32	91,90	32	0,03079	32
	16	91,33	64	0,03157	32
2048 × 1024 (Ölçekleme Oranı: 1/4)	64	90,50	8	0,01263	79
	32	90,96	16	0,01275	78
	16	91,05	32	0,01305	77

[37]'deki çalışmada CityScapes veri seti üzerinde test edilmiş farklı segmentasyon modellerinin, Jetson Xavier NX platformunda elde edilen hız ve sürülebilir alan tespiti başarımları sonuçları ve önerilen yöntem ile karşılaştırılması Tablo 4.8'de gösterilmektedir. Tabloda yer alan yöntemlerin CityScapes kıyaslama (Benchmark) tablosunda bulunanların yol etiketi tespit başarımları dikkate alınmıştır. Bu tablodan görüleceği üzere bu tez kapsamında önerilen yöntem tespit başarımlarından bir miktar taviz vererek gömülü platformlarda oldukça yüksek hızda çalışma potansiyeline sahiptir.

Tablo 4.8. Jetson Xavier NX üzerinde önerilen yöntemin ve farklı segmentasyon yöntemlerinin CityScapes test seti üzerindeki başarımlar ve hız değerleri

Yöntem	Çözünürlük	FPS	Başarımlar (%)	Platform
CGNet [38]	2048 × 1024	2,91	95,9	Jetson Xavier NX
ContexNet [39]	2048 × 1024	10,49	97,6	Jetson Xavier NX
Fast-SCNN [20]	2048 × 1024	11,49	97,9	Jetson Xavier NX
SwiftNet [40]	2048 × 1024	2,61	98,3	Jetson Xavier NX
CABINet [37]	2048 × 1024	8,21	98,2	Jetson Xavier NX
Önerilen Yöntem	2048 × 1024	7	92,55	Jetson Xavier NX
Önerilen Yöntem (Örnekleme Oranı: 1/2)	2048 × 1024	32	91,92	Jetson Xavier NX
Önerilen Yöntem (Örnekleme Oranı: 1/4)	2048 × 1024	78	90,96	Jetson Xavier NX

5. SONUÇLAR VE ÖNERİLER

Bu çalışması kapsamında sürülebilir alanları tespiti yapan sistemlerde kullanılabilecek gömülü sistemler üzerinde de gerçek zamanlı çalışabilen bir derin öğrenme temelli sürülebilir alan tespiti yöntemi geliştirilmiştir. Geliştirilen yöntemde kameradan alınan ham görüntülerin ince dikey şeritler halinde kesilmesi ve az parametre ve karmaşıklığa sahip bir derin öğrenme modeli ile işlenmesi sayesinde gerçek zamanlı çalışma hızları elde edilmiştir. Önerilen yöntemin gerçek zamanlı çalışma kapasitesini göstermek amacıyla bilgisayar ortamında ve farklı işlem gücüne sahip NVIDIA Jetson platformları üzerinde çalıştırılmıştır. Deneysel sonuçlar, önerilen yöntemin yüksek çalışma hızlarında başarılı sonuçlar verdiğini göstermektedir.

Önerilen yöntem belirli aralıklar ile kesilmiş her bir ince dikey şerit için yolun bittiği tek bir noktayı vermekte ve bu noktalar birleştirilerek ilgili görsel için sürülebilir alan tespit edilmektedir. Her bir dikey şerit görselinden tek bir nokta elde edilmesi de sürülebilir alan sınırlarının çok detaylı bir şekilde çıkartılmasına engel olmaktadır. Bu duruma çözüm olarak literatürde sıklıkla segmentasyon temelli derin öğrenme yöntemleri kullanılmaktadır. Fakat bu yöntemler de daha karmaşık bir mimariye sahip olduğu, fazla parametre içerdiği ve fazla hafızaya ihtiyaç duyduğu için çalışma süreleri yüksek olabilmektedir. Önerilen yöntemde segmentasyon temelli yöntemlerden farklı olarak bütün görüntü yerine kesilmiş parçalar modele paralel olarak beslenebilmekte ve model göreceli olarak daha az sayıda konvolüsyon katmanı içermektedir. Önerilen yöntemin bu özellikleri nedeniyle yöntem bilgisayar ve gömülü sistem platformlarında gerçek zamanlı çalışabilme yeteneğini sürdürülebilmektedir.

İlerleyen aşamalarda önerilen yöntemin başarısını artırmak ve sürülebilir alanların sınırlarını daha detaylı çıkartabilmek için farklı model çıkış tiplerinin ve son işlem aşamasının geliştirilmesi hedeflenmektedir. Model çıkışı için çoklu çıkış verebilecek regresif bir katman ile ilgili ince dikey şerit görselindeki sınırlar için birden fazla nokta ile daha az hatayla ve daha detaylı bir şekilde tespit yapabilecek bir yapı geliştirilmesi hedeflenmektedir. Son işlem aşamasında ise tespit edilen noktaların birbirlerine

doğrudan birleştirilmesi yerine bu noktalar kullanılarak oluşturulacak bir eğri ile sürülebilir alanların daha detaylı tespit edilmesi planlanmaktadır.



KAYNAKLAR

- [1] Cordts M., Omran M., Ramos S., Rehfeld T., Enzweiler M., Benenson R., Franke U., Roth S., Schiele B., The Cityscapes Dataset for Semantic Urban Scene Understanding, in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [2] Thakur R., Scanning LIDAR in Advanced Driver Assistance Systems and Beyond, *IEEE Consumer Electronics Magazine*, 2016, **5**(3), 48-54.
- [3] Lee Y., Park S. A Deep Learning-Based Perception Algorithm Using 3D LiDAR for Autonomous Driving: Simultaneous Segmentation and Detection Network (SSADNet). *Applied Sciences*, 2020, **10**(13), 4486, DOI: 10.3390/app10134486.
- [4] Fritsch J., Kuehnl T., Geiger A., A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms, *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2013.
- [5] Lyu Y., Bai L., Huang X., ChipNet: Real-Time LiDAR Processing for Drivable Region Segmentation on an FPGA, *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2019, **66**, 1769-1779.
- [6] Gao B., Xu A., Pan Y., Zhao X., Yao W., Zhao H., Off-Road Drivable Area Extraction Using 3D LiDAR Data, *IEEE Intelligent Vehicles Symposium (IV)*, 2019, 1505-1511, DOI: 10.1109/IVS.2019.8814143.
- [7] Lyu Y., Bai L., Huang X., Real-Time Road Segmentation Using LiDAR Data Processing on an FPGA, *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Florence, Italy, 2018, 1-5, DOI: 10.1109/ISCAS.2018.8351244.
- [8] Liu Z., Yu S., Zheng N., A Co-Point Mapping-Based Approach to Drivable Area Detection for Self-Driving Cars, *Engineering 4*, 2018, **4**(4), 479-490.
- [9] Liu Z., Yu S., Wang X., Zheng N., Detecting Drivable Area for Self-driving Cars: An Unsupervised Approach, *arXiv preprint arXiv:175.00451*, 2017.
- [10] Ragurman S. J., Park J, Intelligent Drivable Area Detection System Using Camera and Lidar Sensor for Autonomous Vehicle, *2020 IEE International Conference on Electro Information Technology (EIT)*, Chicago, IL, USA, 2020, 429-436, DOI: 10.1109/EIT48999.2020.9208327.
- [11] Li Q., Chen L., Li M., Shaw S., Nüchter A., A Sensor-Fusion Drivable-Region and Lane-Detection System for Autonomous Vehicle Navigation in Challenging Road Scenarios, in *IEEE Transactions on Vehicular Technology*, 2014, **63**(2), 540-555, DOI: 10.1109/TVT.2013.2281199.

- [12] He Y., Wang H., Zhang B., Color-based road detection in urban traffic scenes, *in IEEE Transactions on Intelligent Transportation Systems*, 2004, **5**(4), 309-318, DOI: 10.1109/TITS.2004.838221.
- [13] Neto A. M., Victorino A. C., Fantoni I., Ferreira J.V., Real-Time Estimation of Drivable Image Area based on Monocular Vision, *IEEE Intelligent Vehicles Symposium (IV)*, 2013, 63-68, DOI: 10.1109/IVS.2013.6629448.
- [14] Tue-Cuong D., Dong G., Hwand Y. C., Heng O. S., Robust Extraction of Shady Roads for Vision-based UGV, *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, 3140-3145, DOI: 10.1109/IROS.2008.4650955.
- [15] Zhang J., Nagel H., Texture-based segmentation of road images, *Proceedings of the Intelligent Vehicles '94 Symposium*, 1994, 260-265, DOI: 10.1109/IVS.1994.639516.
- [16] Savitha D. K., Rakshit S., Gaussian Mixture Model based road signature classification for robot navigation, *INTERACT-2010*, 2010, 230-233, DOI: 10.1109/INTERACT.2010.5706145.
- [17] Kang B., Q. Nguyen T., Random Forest With Learned Representations for Semantic Segmentation, *IEEE Transactions on Image Processing*, 2019, **28**(7), 3542-3555, DOI: 10.1109/TIP.2019.2905081.
- [18] Guo C., Mita S., McAllester D., Robust Road Detection and Tracking in Challenging Scenarios Based on Markov Random Fields With Unsupervised Learning, *IEEE Transactions on Intelligent Transportation Systems*, 2012, **13**(3), 1338-1354, DOI: 10.1109/TITS.2012.2187896.
- [19] Guo Y., Gerasimov V., Poulton G., Vision-Based Drivable Surface Detection in Autonomous Ground Vehicles, *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, 3273-3278, DOI: 10.1109/IROS.2006.282437.
- [20] Poudel R. P. K., Liwicki S., Cipolla R., Fast-SCNN: Fast Semantic Segmentation Network, *arXiv preprint arXiv:1902.04502*, 2019.
- [21] Chen W., Gong X., Liu X., Zhang Q., Li Y., Wang Z., FasterSeg: Searching for Faster Real-time Semantic Segmentation, *International Conference on Learning Representations (ICLR)*, 2020.
- [22] Emara T., Munim H. E. A. E., Abbas H. M., LiteSeg: A Novel Lightweight ConvNet for Semantic Segmentation, *2019 Digital Image Computing: Techniques and Applications (DICTA)*, 2019, 1-7, DOI: 10.1109/DICTA47822.2019.8945975.
- [23] Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L., MobileNetV2: Inverted Residuals and Linear Bottlenecks, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, 4510-4520, DOI: 10.1109/CVPR.2018.00474.

- [24] Chen X., Lou X., Bai L., Han J., Residual Pyramid Learning for Single-Shot Semantic Segmentation, in *IEEE Transactions on Intelligent Transportation Systems*, 2020, **21**(7), 2990-3000, DOI: 10.1109/TITS.2019.2922252.
- [25] Brostow G. J., Fauqueur J., and Cipolla R., Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 2009, **30**(2), 88–97, DOI: 10.1016/j.patrec.2008.04.005.
- [26] Mazzini D., Schettini R., Spatial Sampling Network for Fast Scene Understanding, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, 1286-1296, DOI: 10.1109/CVPRW.2019.00168.
- [27] Romera E., Alvarez J. M., Bergasa L. M., Arroyo R., Erfnet: Efficient residual factorized convnet for real-time semantic segmentation, *IEEE Transactions on Intelligent Transportation Systems*, 2018, **19**(1), 263–272, DOI: 10.1109/TITS.2017.2750080.
- [28] Everingham M., Van Gool L., Williams C. K., Winn J., Zisserman A., The PASCAL Visual Object Classes (VOC) Challenge, *International Journal of Computer Vision*, 2010, **88**(2), 303–338, DOI: 10.1007/s11263-009-0275-4.
- [29] Mehta S., Rastegari M., Caspi A., Shapiro L., Hajishirzi H., ESPNet: Efficient spatial pyramid of dilated Convolutions for semantic segmentation, *arXiv preprint arXiv:1803.06815*, 2018.
- [30] Lo S. Y., Hang H. M., Chan S. W., Lin J. J., Efficient Dense Modules of Asymmetric Convolution for Real-Time Semantic Segmentation, In *Proceedings of the ACM Multimedia Asia (MMAsia '19)*, 2019, **1**, 1–6. DOI: 10.1145/3338533.3366558.
- [31] Wang, Q., Gao, J., Yuan, Y., Embedding Structured Contour and Location Prior in Siamesed Fully Convolutional Networks for Road Detection, *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, 219-224, DOI: 10.1109/ICRA.2017.7989027.
- [32] Liu X., Deng Z., Segmentation of Drivable Road Using Deep Fully Convolutional Residual Network with Pyramid Pooling, *Cognitive Computation*, 2018, **10**(3), 272–281, DOI: 10.1007/s12559-017-9524-y.
- [33] Lyu Y., Bai L., Huang X., Road Segmentation Using CNN and Distributed LSTM, *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, 1-5, DOI: 10.1109/ISCAS.2019.8702174.
- [34] Levi D., Garnett N., Fetaya E., StixelNet: A Deep Convolutional Network for Obstacle Detection and Road Segmentation, *British Machine Vision Conference (BMVC)*, 109.1-109.12, DOI: 10.5244/C.29.109.

- [35] Everingham M., Eslami S. M. A., Van Gool L., Williams C. K. I., Winn J., Zisserman A., The Pascal Visual Object Classes Challenge: A Retrospective, *International Journal of Computer Vision*, 2015, **111**(1), 98-136, DOI: 10.1007/s11263-014-0733-5.
- [36] Li X., Zhou Y., Pan Z., Feng J., Partial Order Pruning: for Best Speed/Accuracy Trade-off in Neural Architecture, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 9137-9145, DOI: 10.1109/CVPR.2019.00936.
- [37] Yang M. Y., Kumar S., Lyu Y., Nex F., Real-time Semantic Segmentation with Context Aggregation Network, *arXiv preprint arXiv:2011.00993*, 2021.
- [38] Wu T., Tang S., Zhang R., Zhang Y., Cgnet: Light-weight Context Guided Network for Semantic Segmentation, *arXiv preprint arXiv:1811.08201*, 2019.
- [39] Poudel R. P., Bonde U., Liwicki S., Zach C., Contextnet: Exploring Context and Detail for Semantic Segmentation in Real-time, *British Machine Vision Conference (BMVC)*, 2018.
- [40] Orsic M., Kreso I., Bevandic P., Segvic S., In Defense of Pre-trained Imagenet Architectures for Real-time Semantic Segmentation of Road-driving Images, *2012 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 12607-12607, DOI: 10.1109/CVPR.2019.01289.

KİŞİSEL YAYINLAR VE ESERLER

- [1] **Acun O.**, Küçükmanisa A., Genç Y., Urhan O., Drivable Road Area Detection with Regression Output CNN, 28. *Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU)*, Gaziantep, Türkiye, 5-7 Ekim, 2020.
- [2] Erdoğan K., **Acun O.**, Küçükmanisa A., Duvar R., Bayramoğlu A., Urhan O., KEBOT: An Artificial Intelligence Based Comprehensive Anaysis System for FUE Based Hair Transplantation, *IEEE Access*, 2020, **8**, 200461-200476, DOI: 10.1109/ACCESS.2020.3035637.



ÖZGEÇMİŞ

İlköğretim eğitimini Yahya Kaptan İlköğretim Okulunda, lise eğitimini Köseköy Pirelli Anadolu Lisesi olmak üzere Kocaeli’de tamamlamıştır. 2013 yılında girdiği Kocaeli Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü’nden 2017 yılında Elektronik ve Haberleşme Mühendisi olarak mezun oldu.

