



**DERİN SİNİR AĞLARI KULLANILARAK STEREO TERMAL VE
ÇAPRAZ MODALİTE İLE DERİNLİK TAHMİNİ**

Ahmet Faruk AKYÜZ

**YÜKSEK LİSANS TEZİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

EYLÜL 2021

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Ahmet Faruk AKYÜZ

09/09/2021

DERİN SİNİR AĞLARI KULLANILARAK STEREO TERMAL VE ÇAPRAZ MODALİTE İLE DERİNLİK TAHMİNİ

(Yüksek Lisans Tezi)

Ahmet Faruk AKYÜZ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Eylül 2021

ÖZET

Stereo görüntüler kullanılarak derinlik tahmini, bilgisayarlı görme alanında zorlu bir problemdir. Bu problemi çözmek için birçok farklı çalışma yapılmıştır. Gelişen makine öğrenimi ile bu sorunun üstesinden gelmek genellikle sinir ağı tabanlı çözümlerle yapılır. FADNet ve PSMNet sinir ağları, başarıları kanıtlanmış, günümüzde kullanılmaya devam edilen bu çözümlerin önde gelenlerinden ikisidir. Bu tezde, bu iki çalışma, yapılarında uygun değişiklikler yapılarak kullanılmıştır. Öte yandan, bu çözümlerde kullanılan görüntüler çoğunlukla görünür spektrumdadır. Ancak, bazı koşullarda görünür spektrumlardan daha faydalı bilgiler vermesi nedeniyle derinlik tahmini için Kızılötesi (IR) spektrumunun da kullanılma ihtiyacı ortaya çıkmıştır. Bu kapsamda, kullanılabilir veri kümesi olarak CATS veri kümesi bulunmuştur. Fakat bu veri kümesi, kendisinde mevcut olan doğrultulmuş görüntülerin uyumsuzluğu sebebiyle olduğu gibi kullanılamamıştır. Termal-termal görüntü çiftleri için CATS veri kümesinde bulunan ham görüntüler ile LiDAR cihazından elde edilmiş, sahnenin derinliğini ifade eden, uzaysal nokta kümeleri çeşitli işlemlerden geçirilerek sinir ağlarını eğitmek için uygun hale getirilmiştir. Diğer yandan, görünür-termal görüntü çiftleri içinde, CATS veri kümesinde bulunan mevcut görüntüler, görünür kamera ve termal kameraların bakış alanlarının (FOV) farklı olması sebebiyle kullanılamamış, bunun yerine, yeni bir veri kümesi oluşturulmuştur. Oluşturulan bu veri kümesinde, LiDAR cihazından elde edilen ayırık örnekleme sayısının yetersizliği sebebiyle, örnek artırma işlemi yapılmıştır. Bu çalışmada, FADNet ve PSMNet sinir ağları, termal-termal ve görünür-termal görüntü çiftleriyle ayrı ayrı eğitilerek sonuçları karşılaştırılmıştır.

Bilim Kodu : 90521

Anahtar Kelimeler : CNN, Termal stereo eşleştirme, Derinlik tahmini, Derin sinir ağları

Sayfa Adedi : 78

Danışman : Prof. Dr. Hasan Şakir BİLGE

DEPTH ESTIMATION WITH STEREO THERMAL AND CROSS MODALITY USING DEEP NEURAL NETWORKS

(M. Sc. Thesis)

Ahmet Faruk AKYÜZ

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

September 2021

ABSTRACT

Depth estimation using stereo images is a challenging problem in computer vision. Many different studies have been carried out to solve this problem. With developing machine learning, tackling this problem is often done with neural network-based solutions. FADNet and PSMNet neural networks are two of the leading solutions that have proven their success and continue to be used today. In this thesis, these two studies were used with appropriate modifications in their structures. On the other hand, the images used in these solutions are mostly in the visible spectrum. However, the need to use the Infrared (IR) spectrum for depth estimation has emerged, as it gives beneficial information than the visible spectra in some conditions. In this context, the CATS dataset was found as a usable dataset. However, this dataset could not be used as it is due to the incompatibility of the rectified images available in it. For thermal-thermal image pairs, the raw images in the CATS dataset and spatial point cloud, which are obtained by LiDAR and express the depth of the scene, are made suitable for the training of networks by undergoing various processes. On the other hand, for the visible-thermal image pairs, the existing images in the CATS dataset could not be used due to the difference in the field of view of the visible and thermal cameras, instead, a new dataset was created. In this created dataset, due to the insufficient number of discrete samples obtained from the LiDAR device, sample augmentation was performed. In this study, FADNet and PSMNet neural networks were trained separately with thermal-thermal and visible-thermal image pairs and their results were compared.

Science Code : 90521

Key Words : CNN, Thermal stereo matching, Depth estimation, Deep neural networks

Page Number : 78

Supervisor : Prof. Dr. Hasan Şakir BİLGE

TEŞEKKÜR

Bana sağladığı moral ve motivasyon sebebiyle kıymetli eşim Sinem’e, neşe kaynağım olan kızım Halime’ye, tez süreci boyunca yapmış olduğu yardım ve rehberlik sebebiyle danışman hocam Prof. Dr. Hasan Şakir BİLGE’ye, fikir alış-verişinde bulunduğum Ahmet Anıl DURSUN, Ayhan Sefa YILDIZ ve Osman Çağrı FİDAN’a, çalışmalarım için gerekli her türlü altyapıyı sağlayan ASELSAN, ASELSAN AKADEMİ ve Müdürüm Sn. İsmail ÖZSARAÇ’a, beni yetiştiren babam Sn. İsmail AKYÜZ ve kıymetli Aileme, ve daha sayamadığım üzerimde ve bu tezde emeği geçen herkese çok teşekkür ederim.



İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	x
RESİMLERİN LİSTESİ	xii
SİMGELER VE KISALTMALAR.....	xiv
1. GİRİŞ.....	1
2. METOTLAR	11
2.1. Ağ Mimarileri	11
2.1.1. FADNet ağ yapısı.....	13
2.1.2. PSMNet ağ yapısı.....	15
2.2. Veri Kümeleri	17
2.2.1. CATS veri kümesi.....	17
2.2.2. Oluşturulan veri kümesi	19
2.3. Pinhole Kamera Modeli	22
2.4. KITTI Piksel Hata Hesaplaması	23
2.5. McNemar Test Sinaması	24
3. ÖNERİLEN YÖNTEM	27
3.1. Ağ Yapılarındaki Değişiklikler	27
3.1.1. FADNet yapısındaki değişiklikler	27
3.1.2. PSMNet yapısındaki değişiklikler	28

	Sayfa
3.2. Veri Kümeleri Üzerindeki Değişiklikler.....	28
3.2.1. CATS veri kümesinde yapılan değişiklikler	28
3.2.2. Oluşturulan veri kümesinde yapılan değişiklikler.....	37
4. DENEYSEL ÇALIŞMALAR	51
4.1. Ağ Eğitimleri	51
4.1.1. Oluşturulan veri kümesi ile eğitim.....	51
4.1.2. CATS veri kümesi ile eğitim.....	53
4.2. Ağ Eğitim Sayısal Sonuçları.....	55
4.2.1. Oluşturulan veri kümesi sayısal sonuçları.....	55
4.2.2. CATS veri kümesi sayısal sonuçları	56
4.3. Ağ Eğitim Görsel Sonuçları.....	57
4.3.1. Oluşturulan veri kümesi görsel sonuçları	57
4.3.2. CATS veri kümesi görsel sonuçları	58
5. SONUÇ VE ÖNERİLER.....	61
KAYNAKLAR	63
EKLER.....	67
EK-1. Oluşturulan veri kümesi ile eğitilen FADNet ağının test çıktıları	68
EK-2. Oluşturulan veri kümesi ile eğitilen PSMNET ağının test çıktıları	71
EK-3. CATS veri kümesi ile eğitilen FADNet ağının test çıktıları	74
EK-4. CATS veri kümesi ile eğitilen PSMNET ağının test çıktıları	75
EK-5. Kullanılan kavramlar ve İngilizce karşılıkları.....	76
EK-6. Diğer Görüntü Benzerlik Metrikleri ile Elde Edilmiş Sonuçlar.....	77
ÖZGEÇMİŞ	78

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 4.1. Oluşturulan veri kümesinin ortalama kayıp değerleri	53
Çizelge 4.2. CATS veri kümesinin ortalama kayıp değerleri	55
Çizelge 4.3. Oluşturulan veri kümesindeki eğitim sonuçlarının KITTI piksel hata hesaplamaları	55
Çizelge 4.4. Oluşturulan veri kümesi test çıktıları üzerindeki farklı toleranslarda 2x2'lik kontenjans tablosu	56
Çizelge 4.5. CATS veri kümesindeki eğitim sonuçlarının KITTI piksel hata hesaplamaları	56
Çizelge 4.6. CATS veri kümesi test çıktıları üzerindeki farklı toleranslarda 2x2'lik kontenjans tablosu.....	57
Çizelge 5.1. Kullanılan kavramlar ve İngilizce karşılıkları	76
Çizelge 6.1. Diğer Görüntü Benzerlik Metrikleri ile Elde Edilmiş Sonuçlar	77

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1. Projektör ve kamera kullanılarak derinlik çıkartma düzeneği.....	1
Şekil 1.2. Sol ve sağ kamera ile kurulmuş stereo düzeneği	2
Şekil 1.3. Elektromanyetik spektrum.....	6
Şekil 2.1. 2B ve 3B konvolüsyon işlemi.....	12
Şekil 2.2. DispNetC ağ yapısı	14
Şekil 2.3. FADNet ağ yapısı	14
Şekil 2.4. PSMNet ağ yapısı	15
Şekil 2.5. Pinhole kamera modellemesi.....	22
Şekil 2.6. 2x2'lik kontenjans tablosu.....	24
Şekil 3.1. FADNet ağ yapısında değişikliğin yapıldığı katmanlar	27
Şekil 3.2. PSMNet ağ yapısında değişikliğin yapıldığı katmanlar	28
Şekil 3.3. Kaydırma ve döndürme işlemi için Python tabanlı yazılan yazılımın menüsü	32
Şekil 3.4. Kaydırma ve döndürme işlemi için optimum katsayılar.....	33
Şekil 3.5. Hatalı örneklerin kutucuklar içinde gösterilmiş olduğu görüntü.....	34
Şekil 3.6. HSV renk dağılımı.....	36
Şekil 3.7. Örnek doldurma programı arayüzü.....	39
Şekil 3.8. İlk alandan ikinci alana kopyalama işlemi (ilk).....	39
Şekil 3.9. İlk alandan ikinci alana kopyalama işlemi (son)	40
Şekil 3.10. İkinci alanda morfolojik genişleme işlemi.....	40
Şekil 3.11. Aralık kapatma işlemi	41
Şekil 3.12. Kenarlık ekleme işlemi	41
Şekil 3.13. İkinci alanda morfolojik sınır çıkarma işlemi.....	42

Şekil	Sayfa
Şekil 3.14. İkinci alandan üçüncü alana görüntü aktarılması	42
Şekil 3.15. Dördüncü alana görüntü aktarılması.....	43
Şekil 3.16. İkinci alan görüntü sıfırlama işlemi	43
Şekil 3.17. Kişi objesinin segmente edilme işlemi	44
Şekil 3.18. Dördüncü alana yeni bir objenin eklenmesi	45
Şekil 3.19. Dördüncü alanda derinlik bilgisi verilecek tüm objelerin segmente edilmiş son durumu.....	45
Şekil 3.20. Dördüncü bölgede istenmeyen kısımların silinmesi.....	46
Şekil 3.21. Segmente edilmiş görüntüyü kullanarak derinlik bilgisi atama.....	47
Şekil 3.22. Yol objesine derinlik değeri atama	47
Şekil 3.23. İnsan objesine derinlik değeri atama	48
Şekil 4.1. Oluşturulan veri kümesinin FADNet ağındaki kayıp değer grafiği	52
Şekil 4.2. Oluşturulan veri kümesinin PSMNet ağındaki kayıp değer grafiği	53
Şekil 4.3. CATS veri kümesinin FADNet ağındaki kayıp değer grafiği	54
Şekil 4.4. CATS veri kümesinin PSMNet ağındaki kayıp değer grafiği	54
Şekil 4.5. Derinlik haritalarında kullanılan renk barı.....	57

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 1.1. SceneFlow veri kümesi örneği.....	4
Resim 1.2. Middlebury veri kümesi örneği	4
Resim 1.3. KITTI veri kümesi örneği.....	5
Resim 1.4. Gündüz ve termal kamerayla alınmış görüntü farklılıkları.....	7
Resim 2.1. CATS veri kümesi elde etme düzeneği	17
Resim 2.2. CATS veri kümesinde bulunan örnek veri	18
Resim 2.3. CATS veri kümesindeki gerçek referans değerin uyumsuzluğu	19
Resim 2.4. Veri kümesi elde etme düzeneği.....	20
Resim 2.5. Oluşturulan düzeneden alınan örnek veri	21
Resim 3.1. Doğrultma işlemi yapılmamış sol ve sağ termal görüntüler	29
Resim 3.2. Doğrultma işlemi tamamlanmış sol ve sağ termal görüntüler	30
Resim 3.3. Pinhole kamera modeli kullanılarak elde edilmiş 2B iz düşüm görüntüsü ..	30
Resim 3.4. Kaydırma ve döndürme işlemi sonrası iz düşüm alınan gerçek referans değer görüntüsü.....	31
Resim 3.5. Kaydırma ve döndürme yapılmamış görüntüler	33
Resim 3.6. Kaydırma ve döndürme yapılmış görüntüler	33
Resim 3.7. Hatalı örneklerin giderilmiş olduğu görüntü	34
Resim 3.8. Kaydırma ve döndürme işlemi sonrası elde edilmiş görüntü	35
Resim 3.9. Hatalı örneklerin silinmesi işlemi	36
Resim 3.10. Hatalı örneklerin silinmesi işlemi sonrası elde edilmiş görüntü.....	37
Resim 3.11. Örnek artırma işlemi; Sol: RGB görüntüsü, Orta: Örnekleri sadece LiDAR ile edilmiş görüntü, Sağ: Örnekleri artırılmış görüntü	37
Resim 3.12. Doldurma işlemi; Sol: RGB görüntüsü, Orta: Örnekleri arttırılmış görüntü, Sağ: Örnekleri doldurulmuş görüntü	38
Resim 3.13. Derinlik haritasının son hali.....	49

Resim	Sayfa
Resim 4.1. Soldan sağa; RGB görüntüsü, Gerçek referans değer görüntüsü, FADNet çıktısı, PSMNet çıktısı	58
Resim 4.2. Soldan sağa; Sol termal görüntüsü, Gerçek referans değer görüntüsü, FADNet çıktısı, PSMNet çıktısı	58
Resim 1.1. Oluşturulan veri kümesi ile eğitilen FADNet ağının test çıktıları.....	68
Resim 2.1. Oluşturulan veri kümesi ile eğitilen PSMNet ağının test çıktıları.....	71
Resim 3.1. CATS veri kümesi ile eğitilen FADNet ağının test çıktıları.....	74
Resim 4.1. CATS veri kümesi ile eğitilen PSMNet ağının test çıktıları.....	75

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

cm	santimetre
mm	milimetre
GB	gigabayt
hz	Hertz
um	mikrometre
°	derece

Kısaltmalar

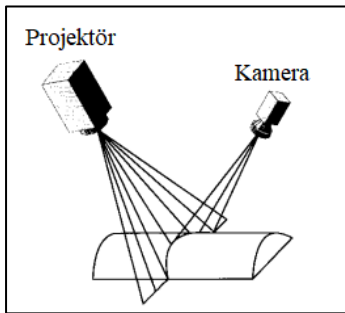
Açıklamalar

CNN	Convolutional Neural Network
CSV	Comma Seperated Values
EPE	End Point Error
FADNet	A Fast and Accurate Network for Disparity Estimation
FOV	Field of View
GC-Net	Geometry and Context for Deep Stereo Regression
HSV	Hue, Saturation, Value
ICP	Iterative Closest Point
IR	Infrared
LiDAR	Laser Imaging Detection and Ranging
PNG	Portable Network Graphics
PSMNet	Pyramid Stereo Matching Network
RGB	Reg, Green, Blue
SGM	Semi Global Matching
TOF	Time of Flight

1. GİRİŞ

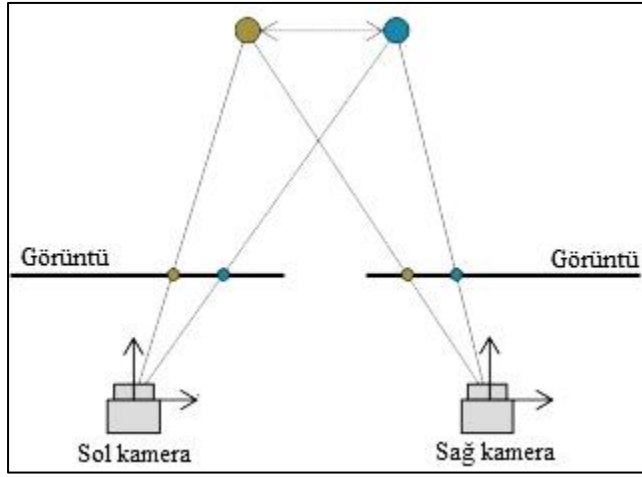
Derinlik tahmini işlemi bilgisayarlı görü alanında halen güncelliğini koruyan, üzerine her geçen gün daha iyi çözüm teknikleri araştırılan bir konudur. Derinlik tahmini yapmak birçok uygulama açısından çok önemlidir ve tahmini daha doğru bir şekilde yapmak çok kritiktir. Günümüzde bu amaç kapsamında uygulanabilir alanlar çok geniş bir yelpazeyi kapsamaktadır. Özellikle gelecek yıllarda daha da yaygınlaşması tahmin edilen otonom sürüş sistemleri, robotik uygulamalar, mesafe sensörü kullanılmayan araçlar vb. gibi birçok alanda derinlik haritasından faydalanmak büyük önem taşımaktadır. Derinlik haritası elde etmenin aktif ve pasif metotlar olmak üzere iki farklı yöntemi bulunmaktadır.

Aktif metotlar, ışık kaynakları, uçuş zamanı (TOF) temelli lazer cihazları, ultrason tabanlı yöntemler ve son olarak bir projektör ve kamera kullanılarak elde edilen yöntem olarak sıralanabilir (Sanz, Mezcuca ve Pena, 2012). Aktif metotlar, mesafe ölçümü olarak doğruluğu çok yüksek olmasına karşın, dezavantaj olarak, enerjiye ihtiyaç duymaktadırlar. Bu metotlardan bazıları günümüzde halen yaygın olarak kullanılmaktadır. Örnek olarak LiDAR, uçuş zamanı temelli, lazer ile ortamı tarayarak mesafe ölçer ve derinlik haritası çıkarır (URL-1). Bu işlemin mesafe ölçümü olarak doğruluğu çok yüksektir, öyle ki bu cihazlar 1 mm hata toleransı ile mesafe ölçümü yapabilir (URL-2). LiDAR, lazer ile mesafe ölçtüğü için derinlik haritası sürekli olmayıp, ayrıktır. Ayrıca lazer kullanımının sakıncalı olabileceği durumlarda olabilir. Bu gibi sebepler, LiDAR kullanımının dezavantajı olarak sıralanabilir. Diğer bir örnek olarak, projektör ve kamera birlikte kullanılarak elde edilen derinlik haritasıdır. Şekil 1.1’de görüleceği üzere projektör belli bir deseni bir sahneye yansıtır. Daha sonra kamera ile bu sahne görüntülenerek, desendeki bozulmaları tespit edilir ve bu bozulmalara göre sahnenin derinliği çıkarılır.



Şekil 1.1. Projektör ve kamera kullanılarak derinlik çıkartma düzeneği (Albrecht ve Michaelis, 1998: 158)

Pasif metotlar ise görüntü üzerinden elde edilen yöntemlerdir. Pasif metotlar, tek (monocular) ve birden fazla (multiview) görüntüden derinlik haritası çıkartma olmak üzere iki kısımda incelenir. Tek görüntüden tahmin edilen derinlik haritası, doğası gereği güvenilir sonuçlar veremez. Özellikle karmaşık görüntülerde hemen hemen hiç derinlik haritası çıkaramazlar. Bu sebeple, tıpkı insanın iki gözü olduğu gibi derinlik tahmini ve tespiti yapmak için referans olarak başka bir görüntünün de olması elzemdir. Buradan hareketle, derinlik tahmini ve tespiti için stereo görüntüler kullanılır. Stereo görüntüler, birden fazla görüntüden derinlik haritası çıkartma işlemi altında incelenir. Şekil 1.2’de örnek bir stereo düzeneği gösterilmiştir.



Şekil 1.2. Sol ve sağ kamera ile kurulmuş stereo düzeneği (Rodríguez-Quiñonez ve diğerleri, 2017: 26)

Stereo görüntüler, aynı yatay doğrultuda bulunup aynı şekilde kalibre edilmiş özdeş iki kameradan elde edilen sol ve sağ görüntü çiftlerinden oluşur (O’Riordan, Newe, Dooly ve Toal, 2018). Stereo eşleme kavramı, sol görüntüde bulunan bir noktanın veya objenin, sağ görüntüdeki yerinin tespit edilmesi işlemidir. Stereo eşleme yaparak derinlik bilgisi çıkartılırken ele alınan temel mantık; görüntüde yakın olan bir noktanın veya objenin sol görüntüdeki yeri ile sağ görüntüdeki yerlerinin çok farklı olması, uzak olan bir noktanın veya objenin de sol görüntüdeki yeri ile sağ görüntüdeki yerinin hemen hemen aynı olması olarak özetlenebilir. Bu mantıkla sol görüntü üzerinden seçilen bir pikselin, aynı doğrultuda sağ görüntüde nerede olduğu tespit edilir. Eğer aradaki fark büyükse, o piksel aynı oranda yakın; fark küçükse, aynı oranda uzak demektir. Bu şekilde tüm piksellere bakılarak, derinlik haritası, referans alınan görüntüye göre elde edilir.

Stereo görüntü çiftleri kullanılarak derinlik haritası elde etme işlemi literatürde iki farklı yöntemle yapılır. İlk yöntem, geleneksel metotlar olarak adlandırılan, temelinde yukarıda bahsedilen mantığı matematiksel olarak yapan yöntemlerdir. Diğer yöntem ise derin sinir ağları kullanılarak yapılan makine öğrenmesi yöntemidir.

Geleneksel metotların kullanımıyla, temelinde matematiksel hesaplamalara dayandığı için çok daha hızlı sonuç elde edilebilir. İşlemin hızlı olmasının avantajı ise gerçek zamanlı uygulamalarda kullanılabilir olmasıdır. Fakat bu metotların başarısı, görüntüleri piksel bakımından ele aldığı için dokusu olmayan alanlar ve tekrarlayan pikseller sebebiyle oldukça düşüktür (Zhang, Prisacariu, Yang ve Torr, 2019). Geleneksel metotlar arasında en yaygın olarak bilinen metot Yarı Küresel Eşleme (SGM) metodudur (Hirschmuller, 2008 : 329).

Makine öğrenmesi, derinlik tahmini yapmak için popüler bir yöntemdir. Son zamanlarda yapılan çalışmalara bakıldığında evrimsel sinir ağları (CNN), bir görüntüde özellik çıkartmada ve benzerlik hesaplamada geleneksel metotlara göre çok daha başarılı olduğu ortaya çıkmıştır (Zagoruyko ve Komodakis, 2015). Ayrıca, derinlik tahmini açısından doğruluğu da geleneksel metotlara göre çok daha iyidir. Makine öğrenmesinde, öğrenme üç şekilde yapılır. İlk ve en yaygın yöntem “denetimli öğrenme” modelidir. Denetimli öğrenmede; ağ, her öğrenme adımında, çıktısını gerçek referans değer (Ground Truth) ile karşılaştırarak belirli bir hata hesaplar. Bu hataya göre ağ, ağırlıklarını günceller ve eğitim adımlarını bu şekilde giderek bu hataları minimize etmeye çalışır (Mohri, Rostamizadeh ve Talwalkar, 2018: 6). İkinci yöntem ise “denetimsiz öğrenme” modelidir. Bu modelde; ağın gerçek referans değer bilgisine ihtiyacı yoktur. Bunun yerine, oluşturduğu çıktıyı, benzer özelliklere göre gruplayarak, ağırlıklarını buna göre günceller ve bu işlemi istatistik alanındaki yoğunluk tahmininden yararlanarak yapar (Jordan ve Bishop, 2004: 66). Son yöntem ise yarı denetimli öğrenme modelidir. Bu modelde, ağ hem denetimli öğrenmede olduğu gibi gerçek referans değer ile hata hesaplaması yaparak, hem de denetimsiz öğrenmede olduğu gibi çıktı özelliklerine göre gruplayarak ağırlıklarını günceller. Bu model eğitiminde veri kümesinin tüm değerleri için gerçek referans değer bilgisine ihtiyaç yoktur. Özellikle veri kümesindeki eksik bilgileri tamamlamada kullanılır.

FADNet (Wang, Shi, Zheng, Zhao ve Chu, 2020), PSMNet (Chang ve Chen, 2018), GC-Net (Kendall ve diğerleri, 2017) vb. gibi başarılı birçok çalışma denetimli öğrenme

modeline dayalıdır. Denetimli öğrenme de yukarıda bahsedildiği gibi gerçek referans değer çok önemlidir ve tüm öğrenme süreci buna göre ilerler. Bu ağlarda derinlik tahmini yapılırken gerçek referans değer olarak olması gereken derinlik bilgisi ağa verilir. Bu sebeple, bu uygulamalarda veri kümesi çok önemlidir ve doğru bir şekilde ağa verilmesi gerekir. Eğer veri kümesi hatalıysa, ağ iyi bir şekilde eğitilemez ve tatmin edici sonuçlar elde edilemez.

Stereo eşleme ve derinlik haritası uygulamaları için sentetik veri kümeleri, laboratuvar veri kümeleri ve gerçek veri kümeleri olmak üzere üç tip veri kümesi vardır. Sentetik veri kümeleri bilgisayarlardan yararlanılarak gerçek olmayan, animasyonla oluşturulmuş görüntülerden oluşur. SceneFlow (Mayer ve diğerleri, 2015) veri kümesi bu tip veri kümesine örnek olarak verilebilir. Bu veri kümeleri bilgisayar ortamında oluşturulduğu için derinlik bilgisi, herhangi bir cihaz kullanılmadan kolayca elde edilebilir ve zahmetsizdir. Resim 1.1’de bu veri kümesinin örneği gösterilmiştir.



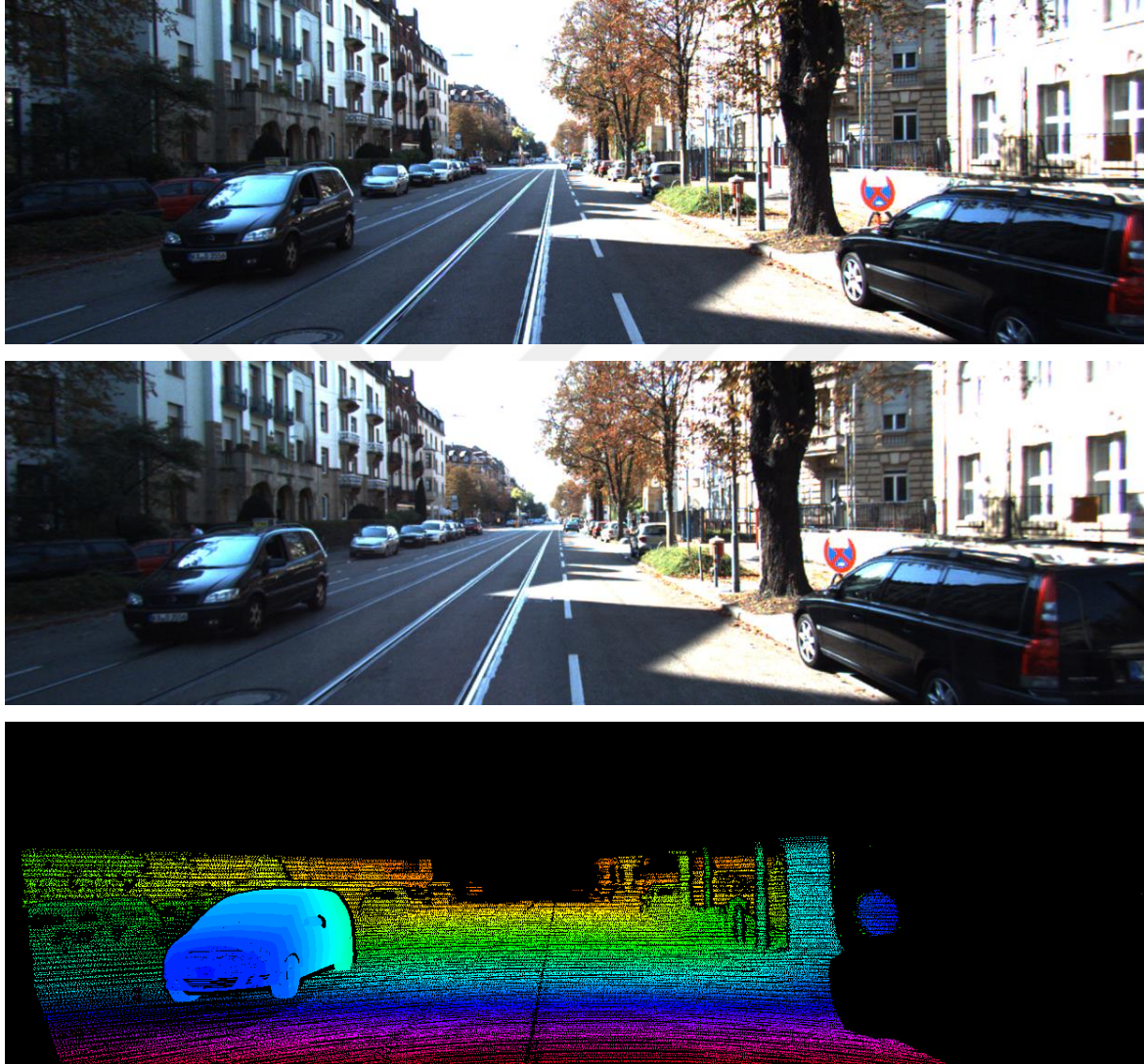
Resim 1.1. SceneFlow veri kümesi örneği

Laboratuvar veri kümeleri denilince akla Middlebury (Scharstein, Szeliski ve Zabih, 2001) veri kümesi gelmektedir. Bu veri kümesi, laboratuvar ortamında özdeş iki kamera ve bir projektörden oluşturulmuştur. Her sahne için satranç tahtasıyla kamera kalibrasyonları yapılır. Sonrasında aktif metotlar kısmında bahsedilen projektör yöntemi kullanılarak derinlik haritası çıkartılır. Resim 1.2’de bu veri kümesine örnek gösterilmiştir.



Resim 1.2. Middlebury veri kümesi örneği

Üçüncü ve son olarak gerçek veri kümeleri, tamamen gerçek sahnede özdeş iki kamera ve mesafe sensörü (LiDAR) kullanılarak oluşturulur. Bu veri kümesine, yaygın bir şekilde kullanılan KITTI (Geiger, Lenz ve Urtasun, 2012a) veri kümesi örnek olarak gösterilebilir. Resim 1.3'te bu veri kümesinin örneği gösterilmiştir. Gerçek senaryo uygulamaları için gerçek veri kümeleri seçilmelidir.

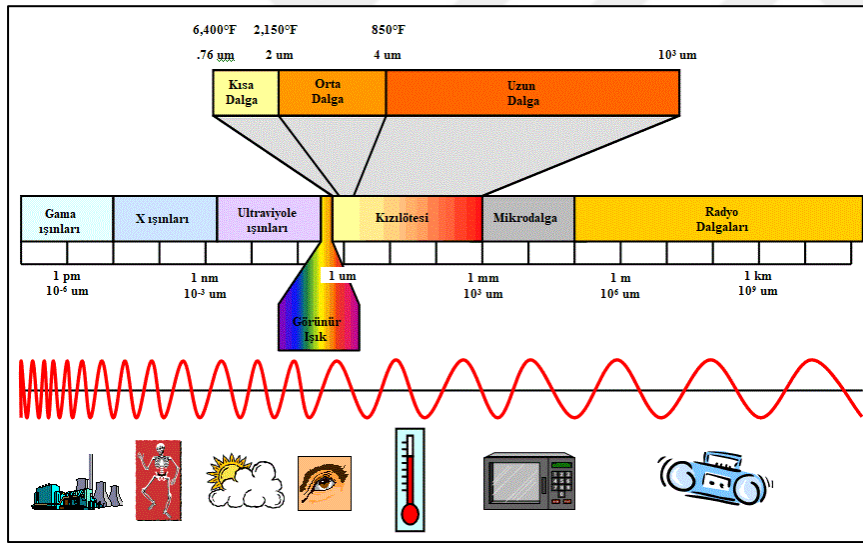


Resim 1.3. KITTI veri kümesi örneği

Gerçek veri kümelerinin denetimli sinir ağlarında kullanılabilmesi için bazı özelliklere sahip olması gerekir. Bu özellikler; kalibrasyon işlemi, doğrultma işlemi ve eşleştirme işlemi yapılarak elde edilir (Geiger ve diğerleri, 2012a). Kalibrasyon işleminde kamera parametreleri optimize edilerek ortalama yeniden yansıtma hatası minimize edilmeye çalışılır (Geiger, Moosmann, Car ve Schuster, 2012b). Doğrultma işleminde, kameradan alınan her iki görüntünün de birbirleri arasında paralel olarak aynı doğrultuda olması ve

odaklarının sonsuzda kesişecek şekilde olması sağlanır. Son olarak, eşleştirme işleminde ise LiDAR cihazı ile referans olarak alınan kameranın uzay koordinatlarının farklı olması sebebiyle, LiDAR cihazından örneklenen uzay koordinat noktalarına gerekli kaydırma ve döndürme verilerek, iki boyutlu görüntü düzlemine iz düşümü alındığında referans görüntüye uygun olacak şekilde eşleştirilmesi sağlanır.

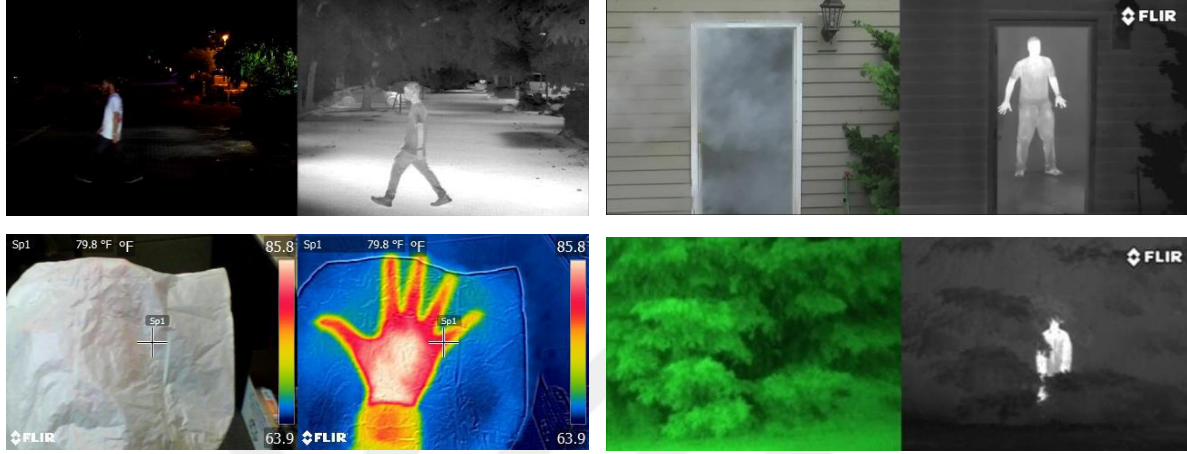
Gelişen teknolojiyle birlikte farklı dalga boyları kullanılarak elde edilen bilgilerden yararlanma büyük önem kazanmıştır. Nükleer enerjiden, radyo yayınlarına kadar birçok uygulama bu dalga boylarının getirdiği kazanımlar sayesinde günümüzde kullanılmaktadır. Bu çalışmada ilgilenilen dalga boyu aralığı $76 \mu\text{m}$ ile $10^3 \mu\text{m}$ arasında bulunmakta olup, kızılötesi (IR) spektrumu olarak adlandırılır. Bu spektrumun en önemli özelliği objelerin termal özelliklerini yansıtmasıdır. Şekil 1.3'te elektromanyetik spektrum ve uygulama alanları gösterilmiştir.



Şekil 1.3. Elektromanyetik spektrum (URL-3)

IR spektrumu, bu dalga boyunu algılayıcı sensörlerinin ucuzlayan maliyetleri ve normal kameralara göre gösterdikleri farklı dalga boylarındaki bilgiler nedeniyle günümüzde daha yaygın kullanılmaya başlanmıştır. Termal kameralar bu spektrumun içerisinde yer alır. Termal kameralar ile görünür spektrumdaki kameralar (gündüz kamerası) karşılaştırıldığında, termal kameraların birçok avantajı bulunmaktadır. Örneğin, gece koşullarında gündüz kameraları ışığın az olmasından ötürü anlamlı bilgi veremezken, termal kameralar ışığa değil termal bilgiye baktığı için daha anlamlı görüntüler verirler. Ayrıca, Resim 1.4'te gösterilen bazı örneklerde de görüleceği gibi, termal kameraların

sisli, puslu, yağmurlu gibi havalarda yine görünür spektrumdaki kameralara göre daha iyi görüntü verdiği bilinmektedir (Beier ve diğerleri, 1994: 175). Termal kameraların gündüz kameralarına göre dezavantajı ise renk ve doku bilgilerinin olmamasıdır. Bu nedenle, termal kameraların görüntü seviyeleri gridir.



Resim 1.4. Gündüz ve termal kamerayla alınmış görüntü farklılıkları (Ivašić-Kos, Krišto ve Pobar, 2019; URL-4; URL-5)

Derinlik haritası çıkartma işleminde termal görüntülerin kullanılması fikri, yukarıda bahsedilen avantajlarından yararlanmak için doğmuştur. Örneğin, sisli hava koşullarında araç sürüş uygulaması ele alınacak olursa, gündüz kameralarında görünür menzil dramatik bir şekilde düşeceği için, gündüz kamerasının yanında termal kamera da kullanılarak derinlik haritasından yararlanılmaya devam edilebilir. Başka bir örnek uygulama olarak, gece koşullarında harp ortamında, düşman mevziilerinden ateşlenen patlayıcı bir mühimmatın mesafe tespitinde, karşı tedbir olarak stereo termal kameralar kullanılarak derinlik haritasından yararlanılabilir. Bunun gibi örnekler çoğaltılabilir.

Termal görüntü veya görüntülere sahip gerçek veri kümesi literatürde araştırılmıştır. Fakat, stereo görüntülerden derinlik haritası çıkartma işlemi, yüksek yoğunlukla görünür spektrumda yapıldığı için, veri kümeleri sadece görünür spektrumu kapsamaktadır. Bu şartı sağlayan sadece CATS veri kümesi bulunmuştur (Treible ve diğerleri, 2017). Ayrıca bu veri kümesine ek olarak, bu çalışmada çapraz modalite özellikli (Termal – RGB) veri kümesi de oluşturulmuştur.

Literatürde stereo termal görüntülerden derinlik haritası elde etme işlemiyle alakalı çeşitli çalışmalar bulunmaktadır. Bu çalışmalar, geleneksel metotlar ile çözümler ve derin sinir ağları ile çözümler olmak üzere iki başlık altında incelenebilir. Geleneksel çözümlerden ilk olarak Dhua, Cutu, Hammoud ve Kiselewich (2003) çözümünden bahsedilebilir. Dhua ve diğerleri (2003), verilen kalibre edilmemiş stereo termal görüntülerle derinlik haritasını hesaplamak için bir yöntem önermişlerdir. Önce köşe eşleştirme yöntemlerini kullanarak aralıklı derinlik haritasını hesaplarlar, daha sonra hesaplanan haritayı önerilen yöntem olan üçgensel kısıtlamalar ve epipolar geometrik kısıtlamalar ile iyileştirmişlerdir.

Bertozzi, Broggi, Lasagni ve Rose (2005), iki özdeş termal kamera kullanarak yayaları tespit etmek için bir algoritma önermişlerdir. Algoritma önce sahnedeki sıcak alanları bulur ve yerlerini tahmin eder, ardından benzer konumlara sahip alanlarda belirli boyutları ve en boy oranlarını dikkate alır ve son olarak olası yayaları listelemek için insan kafasının morfolojik ve termal özelliklerini kullanır.

Zoetgnande, Cormier, Fougères ve Dillenseger (2020: 103161), düşük çözünürlüklü stereo termal görüntülerde bir alt piksel eşleştirme yöntemi önermişlerdir. İlk olarak, faz uyumuna dayalı olarak sağlam öznitelikler çıkarırlar, daha sonra bu öznitelikleri piksel kesinliğinde eşleştirirler ve son olarak, bölgesel faz uyumuna dayalı olarak ara piksel doğruluğunda eşleşmeyi iyileştirmişlerdir.

Iqbal ve Basha (2017) kalibre edilmemiş stereo ve termal görüntülerden derinlik tahmini ve ölçümü yapmışlardır. Metotlarında stereo görüntülerin özelliklerinden en olası olanlarını eşleştirmek için mutlak farklarının toplamı (Sum of Absolute Difference) algoritmasını kullanmışlardır. Bu algoritma sonucunda elde edilen eşlenmiş noktaları, epipolar geometri kuralına uygun olup olmadığına bakmışlardır. Daha sonra uygun olan noktalar ile epipolar temel matrisi hesaplanır ve üç boyutlu düzeltilmiş stereo görüntü için kullanılacak olan içsel noktalar elde edilir. Algoritma bu iki veriyi kullanarak düzeltilmiş görüntüleri elde eder ve piksel kaymasına bakarak derinlik haritasını çıkarır.

Derin sinir ağlarındaki çözümlerden ilk olarak Kim, Choi, Hwang ve Kweon'un (2018) çalışmasından bahsedilebilir. Kim ve diğerleri (2018) tüm gün (gece, gündüz, sis vs.) şartlarında bir adet termal görüntü üzerinden Multispektral Transfer Ağı (Multispectral Transfer Network – MTN) ismini verdikleri bir yapıyla derinlik tahmini yapmışlardır.

Denetimsiz öğrenme modeli tabanlı bu çalışmada, daha iyi derinlik haritası oluşturulması için üç ana katkı sağlanmıştır. İlk adım olarak, derinlik tahmini için çok görevli öğrenmelerde etkili olan kromatiklik konseptini sunmuşlardır. İkinci olarak genel atlamalı bağlantılar ile karşılaştırıldığında, alt katmanların daha hassas ayrıntılarını ve kromatiklik özelliklerini, atlama bağlantılı etkinleştirme üzerine, eşzamanlı olarak kodlayan serpiştirici modülünü önermişlerdir. Son olarak, tüm gün derinlik tahminini, farklı zamanlarda termal kontrast değişimlerinin üstesinden gelmek için, eğitim ve foto metrik düzeltme sırasında büyük farklılıkları kapatmak adına uyarlanabilir çeşitli teknik hususlar sunmuşlardır. Bu ağı eğitim sürecinde iki RGB görüntü ve bir termal görüntü kullanılırken, test sürecinde ise tek termal görüntü yeterlidir.

Y. Lu ve G. Lu (2021), tek termal görüntüden derinlik tahmini yaparlar. Çalışmalarında üç ana adım uygulamaktadırlar. İlk adımda stereo gündüz kameralarından aldıkları görüntülerle derinlik tahmini yaparlar. Kullandıkları termal kamera ile sağ gündüz kamera birbirlerine çok yakın konumlandırıldığı için bu tahmin, sağdaki gündüz kamerasına göre yapılır. Elde edilen bu görüntü, termal kameranın görüntüsünden elde edilecek derinlik haritasını başlatmak için kullanılır. İkinci adımda, soldaki gündüz kamerasından elde edilen görüntü, ikili CycleGAN (Zhu, Park, Isola ve Efros, 2017) ağı vasıtasıyla sentetik termal görüntüye dönüştürülür. Daha sonra, elde edilen bu sentetik termal görüntü, sağdaki gündüz kamerasından elde edilen derinlik haritasıyla iyileştirilir. Son adımda ise CNN yapısı kullanılarak iyileştirilen bu sentetik termal görüntü, sağdaki gündüz kamerasına yakın konumlandırılmış gerçek termal kamerayla birlikte kullanılarak piksel hareketlerinden derinlik haritasının nihai hali elde edilir. Bu ağ yapısında eğitim sürecinde, Kim ve diğerlerinin (2018) çalışmasında da olduğu gibi, iki RGB görüntü ve bir termal görüntü kullanılırken, test sürecinde yine tek termal görüntü yeterlidir.

Shin, Kweon, K. Lee ve S. Lee (2021) yaptıkları çalışmada tek termal videodan derinlik ve poz tahmini yaparlar. Eğitim sürecinde RGB video ve Termal videoyu birlikte kullanırlar. Önce termal videodan tek çerçeveli bir görüntü Pose Net ve Depth Net adında paralel yapıları iki ağı verilir. Pose Net ağından bir sonraki çerçevenin pozunu, Depth Net ağından da anlık derinlik tahmini yapılır. Daha sonra, videoda sonraki termal çerçeveler, elde edilen poz tahmini ve derinlik tahmini kullanılarak yeniden oluşturulur. Aynı anda paralel olarak, “İleri derinlik bükme” (Forward Depth Warping) ağ yapısıyla tahmin edilen derinlikten RGB görüntünün derinliği, “Poz Bükme” (Pose Warping) ağ yapısıyla da tahmin edilen

poz görüntüsüyle RGB görüntüsünün sonraki çerçevelerinin pozunu tahmin edilir. En son olarak RGB görüntülerine göre tahmin edilmiş poz ve derinlik tahminlerinden RGB görüntüleri oluşturulur. Bu tahminlerden elde edilen hatalar minimize edilerek (eğitilerek), tek termal videodan derinlik haritası ve poz tahmini yapılır.

Literatürde yapılan son çalışmalar dikkate alındığında, özellikle termal görüntüler ile yapılan çalışmalar son yıllarda giderek artmaktadır. Bu durum ise çalışmalarda termal görüntü kullanımının önemini gösteren başka bir ölçüttür. Bu tezde, renkli görüntüler ile yapılmış başarılı ağların yapısı optimize edilerek termal görüntüler için de başarılı sonuçlar alınabilir mi sorusuna cevap olarak, denetimli öğrenme modeline sahip FADNet ve PSMNet sinir ağı çözümlerinde stereo termal görüntüler kullanılarak derinlik tahmini yapılmıştır. Bu kapsamda, CATS veri kümesi kullanılmıştır. Fakat, CATS veri kümesinin Bölüm 2.2.1’de bahsedilecek uyumsuzluklarından dolayı doğrudan kullanılamamış olup, veri kümesi üzerinde görüntü doğrultma, iz düşüm alma, LiDAR verilerinde kaydırma ve döndürme ve son olarak hatalı örnek kaldırma işlemleri uygulanmıştır. Sonuçlar incelendiğinde, mevcut sinir ağı çözümlerinde stereo termal görüntüler kullanılarak derinlik tahmininin yapılabilmesi gözlenmiştir.

Ayrıca, bu tez için yine FADNet ve PSMNet ağları üzerinde çapraz modalite (renkli ve termal görüntü) kullanılarak derinlik tahmini yapılmıştır. CATS veri kümesinde bulunan gündüz ve termal kameraların bakış alanlarının çok farklı olması sebebiyle CATS veri kümesindeki çapraz modalite görüntüleri kullanılamamış, yerine özgün veri kümesi oluşturulmuştur. Oluşturulan bu veri kümesindeki herhangi bir görüntünün gerçek referans değerindeki örnek sayısının yeterli seviyede olmaması sebebiyle çeşitli morfolojik teknikler kullanılarak örnek artırma işlemi yapılmıştır. Elde edilen nihai veri kümesiyle eğitim süreci tamamlanmış fakat tatmin edici sonuçlar elde edilememiştir.

2. METOTLAR

2.1. Ağ Mimarileri

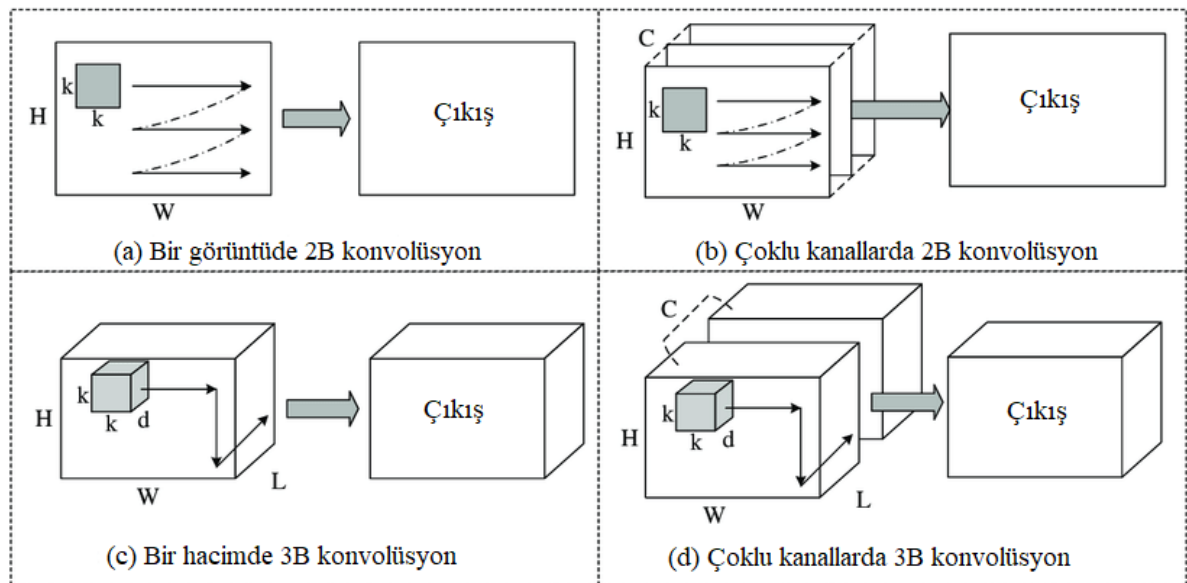
Her alanın kendine has terminolojisi olduğu gibi, makine öğrenmesi alanının da terminolojisi bulunmaktadır. Bu terminolojiye hâkim olmak, ağ mimarilerinin daha iyi anlaşılmasına yardımcı olur. Bu terminolojide sıkça geçen terimler; katmanlar (Layers), kanal sayısı (Channel Size), filtre boyutu (Filter Size) ve konvolüsyon (Convolution), artık bloklar (Residual Blocks), korelasyon katmanı (Correlation Layer), parti boyutu (Batch Size), gerçek referans değer, kayıp değeri (Loss), fazla uyum (Overfitting), yetersiz uyum (Underfitting), eğitim turu (Epoch) ve kaydırma adımı (Stride) olarak sıralanabilir.

- Katmanlar; art arda gelen, her birinde giriş ve çıkış değeri olan, giriş değerini belirli bir ağırlıkla çarparak direk olarak veya verilen eşik değerine göre dolaylı olarak çıkışa atayan yapılardır.
- Kanal sayısı; işlenecek verinin paralel olarak adet miktarını belirtir. Örneğin, RGB görüntüler için bu sayı üçtür. Gri seviye görüntüler için birdir.
- Filtre boyutu ve konvolüsyon katmanı; filtre belirli boyutları olan ve her katmanda kullanılan ağırlıklardır. Konvolüsyon, diğer bir ismiyle evirişim, katmanı ise verilen veri üzerinde bu filtreleri uygulayarak çıkış verisini oluşturan katmanlardır.
- Artık bloklar; peş peşe gelen iki katmandan oluşur. İlk katmana giren verinin aradaki katmanlardan geçerek çıkış tarafında yine kendisiyle toplamından oluşur. Artık bloklar, özellik çıkartmada (Feature extraction), normal katmanlara göre çok daha başarılı olduğu bilinmektedir (Du, El-Khamy, Lee, 2019).
- Korelasyon katmanı; iki farklı kaynaktan gelen verinin benzerliğini bulmak için oluşturulmuş katmandır.
- Parti boyutu; ağ eğitimi işlemi sırasında aynı anda paralel olarak verilen giriş verisi miktarını belirtir.
- Gerçek referans değer; sinir ağının bir eğitim adımı sonunda elde ettiği çıkışı karşılaştırdığı değerdir.
- Kayıp değeri; sinir ağının bir eğitim adımı sonunda elde ettiği çıkışı, gerçek referans değerine göre çeşitli yöntemlerle kıyaslayıp elde ettiği hata değeridir. Bu değere göre katmanlardaki ağırlıklar güncellenir ve daha az kayıp değeri elde edilmeye çalışılır.

- Fazla uyum; sinir ağının eğitim sırasında kayıp değerinin aşırı az olması veya hiç olmaması durumudur.
- Yetersiz uyum; sinir ağının eğitim sırasında kayıp değerinin başarılı olamayacak kadar aşırı fazla olması durumudur.
- Eğitim turu; sinir ağının eğitimi sırasında, veri kümesindeki eğitim amaçlı kullanılan tüm görüntüler özelinde, ilk görüntüden son görüntüye kadar olan sürece denir.
- Kaydırma adımı; verilen veri üzerinde belirli ağırlıklara sahip filtreyi kaydırırken aradaki adım miktarını belirtir. Bu işlem doğrudan çıkış verisinin boyutunu etkiler.

Bu terimlerin kökeni İngilizce olması sebebiyle, Türkçe kullanımında doğru anlaşılama problemi olabilir. Bu maksatla EK-5'te sıkça kullanılan kavramların İngilizce karşılıkları verilmiştir.

Bu çalışmada FADNet ve PSMNet ağının seçilmesinin sebebi, her iki ağ da literatürde iyi başarı oranlarıyla kabul görmüş ve çok sık kullanılmış olmalarından dolayıdır. İkisi arasındaki en temel fark, PSMNet ağında derinlik çıkartma işlemi için üç boyutlu (3B) konvolüsyon katmanları kullanılırken, FADNet ağında iki boyutlu (2B) konvolüsyon katmanları kullanılmıştır. 3B konvolüsyon katmanında 3B filtreler kullanılır, 2B konvolüsyon katmanında ise 2B filtreler kullanılır. Şekil 2.1'de konvolüsyon işlemi görselleştirilmiştir.



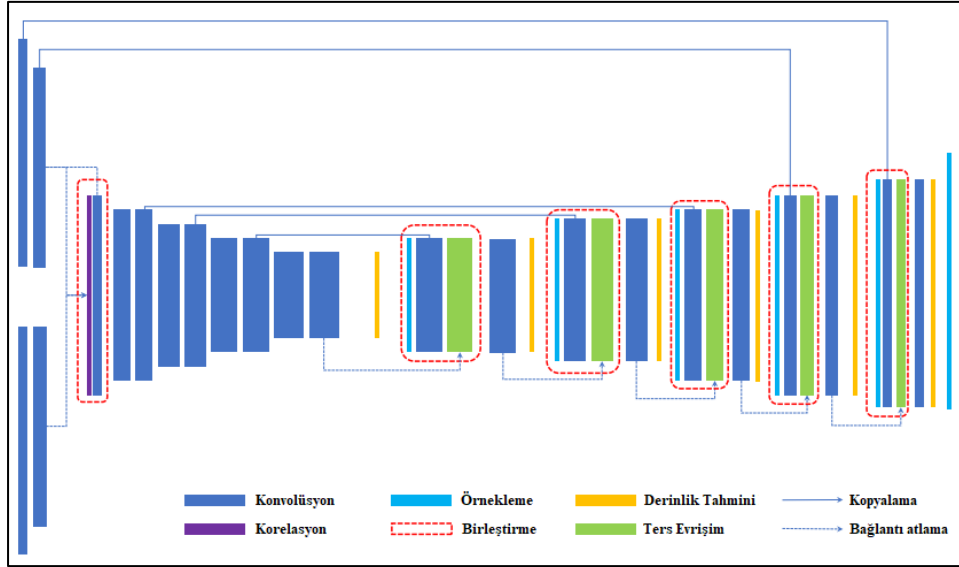
Şekil 2.1. 2B ve 3B konvolüsyon işlemi (Liu ve diğerleri, 2019: 65)

Şekil 2.1-a bölümüne bakıldığında, $H \times W$ boyutlarındaki tek kanallı veriye $k \times k$ boyutlarında bir filtre vasıtasıyla konvolüsyon işlemi yapılıyor. Kaydırma adımı miktarınca bu filtre sağa doğru, bitiminde de yine kaydırma adımı miktarınca aşağıya kaydırılıp tekrar sağa doğru ilerletilerek eleman bazında çarpma işlemi uygulanır ve yeni bir çıkış verisi elde edilir. Şekil 2.1-b bölümünde ise aynı işlem aynı şekilde yapılır. Buradaki fark ise kanal sayısı (C) birden fazla olduğu için bu durumda filtrenin boyutları da C miktarınca $k \times k$ olur ve yine eleman bazında çarpma işlemi uygulanır. Şekil 2.1-c’de ise, veri $H \times W \times L$ boyutlarında üç boyutlu bir veridir. Burada filtre boyutu da $k \times k \times d$ olur ve kaydırma, 2B konvolüsyon işlemine ek olarak derinliğe (içeri) doğru da yapılır. Son olarak Şekil 2.1-d’de veri boyutu $H \times W \times L \times C$, filtre boyutu da yine C miktarınca $k \times k \times d$ olur. Bu filtreler verilere C miktarınca aynı anda kaydırma şeklinde uygulanır. Dolayısıyla, işlem süresi ve hacmi, 3B konvolüsyon katmanlarında 2B konvolüsyon katmanlarına kıyasla çok daha yüksektir.

2.1.1. FADNet Ağ Yapısı

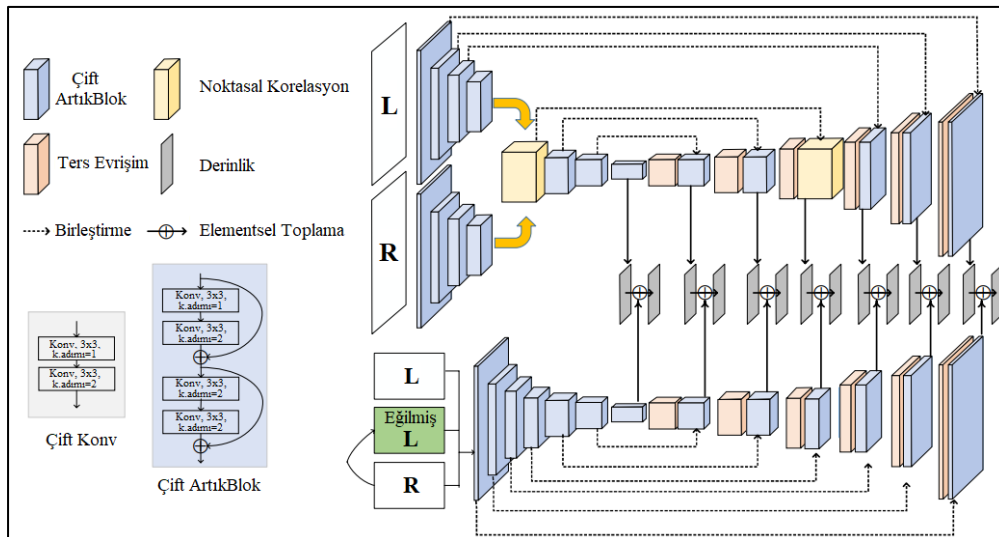
FADNet, 3B CNN katmanları yerine yığılmış bloklarla 2B tabanlı korelasyon katmanlarını kullanır. Bu durum, hızlı hesaplamayı korumaya yardımcı olur. Öyleki, FADNet, PSMNet’e göre 20 kat daha hızlı çalışmasının yanında 10 kat da daha az bellek tüketir. Bu sayede gerçek uygulamalarda kullanılmaya daha uygundur. FADNet, doğruluğu arttırmak için çok ölçekli tahminler yaparak çok ölçekli ağırlık planlama eğitim tekniklerini uygular. FADNet’in ağ yapısı DispNetC ve DispNetS (Mayer ve diğerleri, 2015) olmak üzere iki ağ yapısının yığınınından oluşmaktadır.

FADNet, kodlayıcı – kod çözücü yapısından oluşur. Kodlayıcı taraf özellik çıkartırken, kod çözücü taraf ise çıkartılan özelliklerden derinlik tahmini yapar. Korelasyon katmanı derin sinir ağlarının öğrenme becerisini ve farklı ölçeklerdeki güçlü özelliklerin öğrenimini ciddi miktarda artırır. Bununla birlikte derin sinir ağlarında artık blok ve maliyet hacmi kullanımı, öğrenimi daha da kolaylaştırmak için önerilmiştir. Maliyet hacmi sol özellik haritalarının, ona karşılık gelen tamamlayıcı sağ özellik haritalarıyla farklı ölçeklerde birleştirilmesiyle elde edilir. FADNet öncelikli olarak DispNetC (Şekil 2.2) yapısını omurga olarak kullanır.



Şekil 2.2. DispNetC ağ yapısı (Chen ve diğerleri, 2020: 588)

Bu yapıdaki korelasyon katmanını noktasal korelasyon ile, konvolüsyon katmanlarını ise artık bloklar ile değiştirerek ağın hem doğruluğunu hem de çıkarım hızını artırır. FADNet, DispNetS ve DispNetC direk olarak yığmak yerine; Pang, Sun, Ren, Yang ve Yan'ın (2017) önerdiği çoklu ölçekte artık öğrenme yapısını uygular. Buradaki temel fikir, tüm derinlik haritasını doğrudan tahmin etmek yerine, ikinci iyileştirme ağının (DispNetS) derinlik artıklarını öğrenmesi ve bunları birinci ağ (DispNetC) tarafından oluşturulan ilk sonuçlarla toplamasıdır (Şekil 2.3). Bu şekilde, değişim derecesini (gradient) kaybolmasını önlemekte etkili olan ikinci ağın, yalnızca yüksek düzeyde doğrusal olmayan artıkları öğrenmeye odaklanması beklenir.

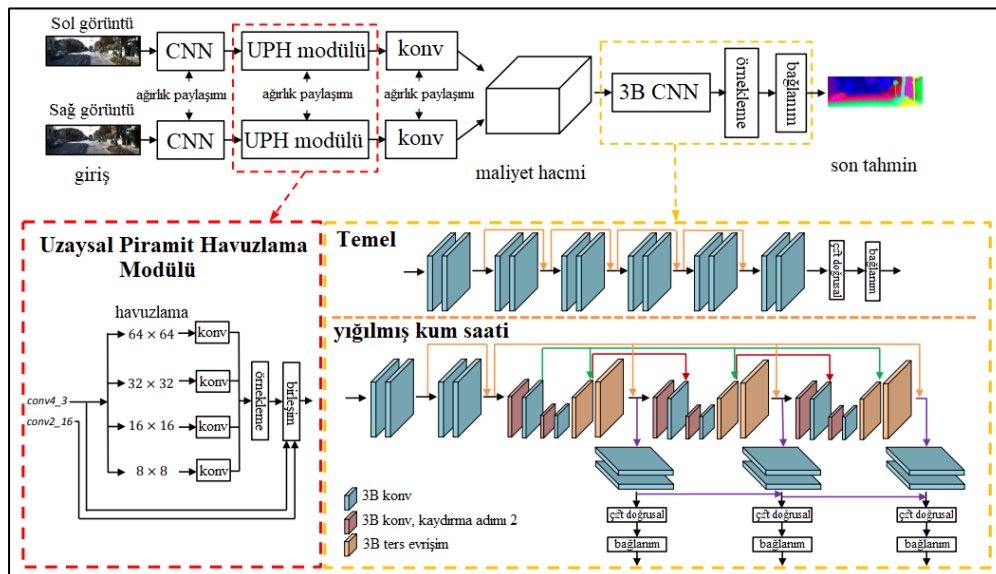


Şekil 2.3. FADNet ağ yapısı (Wang ve diğerleri, 2020)

Son olarak, FADNet kayıp değer fonksiyonu geliştirmiştir. Verilen stereo görüntüden yedi farklı derinlik tahmini yapar. Bu tahminler farklı ölçeklerdeki. Uçtan uca eğitim için, FADNet, tahmin edilen derinliklerle gerçek referans değer arasında piksel bazında pürüzsüz L1 kayıp değer (pixel-wise smooth L1 loss) hesaplaması yapar. Aynı ayrı hesaplanan kayıp değerler birbirleriyle toplanmasıyla eğitim kayıp değeri ortaya çıkar ve ağ, bu değeri minimum hale getirmek için ağırlıklarını günceller. Eğitim sırasında ilk olarak Mayer ve diğerlerinin (2015) önerdiği kayıp değer ağırlık planlama tekniği kullanılmıştır. Toplamda yedi ölçekte tahmin edilen derinlik haritası, başlangıçta, kaba özellikleri öğrenmek için düşük değerli ağırlıklar büyük ölçekli derinlik haritalarına atanır. Ardından, ağın, daha ince özellikleri kademeli olarak öğrenmesini sağlamak için büyük ölçeklerin kayıp ağırlıkları artırılır. Son olarak doğrulama sırasında, orijinal girdi boyutunun olduğu son tahmin dışındaki tüm kayıplar devre dışı bırakılır. Bu teknik ile doğrulama adımı son nokta hatasında (EPE) ciddi bir gelişme kaydedilir.

2.1.2. PSMNet Ağ Yapısı

PSMNet (Şekil 2.4), piksel düzeyindeki özellikleri bölge düzeyindeki özelliklere genişletmek için farklı alıcı alan ölçeklerinden yararlanır. Bu, güvenilir derinlik tahmini elde etmek için maliyet hacmini oluşturan küresel ve yerel özellik ipuçlarına imkân verir. PSMNet ayrıca maliyet hacmini düzenlemek için tekrarlı 3B CNN katmanları uygulanan yığılmış kum saati mimarisini kullanır.



Şekil 2.4. PSMNet ağ yapısı (Chang ve Chen, 2018)

Derin sinir ağlarında derinlik tahmini ile ilgili birçok çalışmada Siyam ağları kullanılır. Siyam ağları, CNN katmanlarının peş peşe gelmesiyle oluşturulmuş bir kol ile, aynı yapıya ve ağırlıklara sahip ikinci bir kolu, paralel bir şekilde oluşturularak bir araya getirmesiyle elde edilir. Burada bir kol sol görüntüyle, diğer kol ise sağ görüntüyle beslenir ve kol ağırlıkları aynı olması nedeniyle benzer özellikler çıkartılır. Bu yapı sebebiyle Siyam ikizlerine benzetilir ve ismini buradan alır. Yapısından da anlaşılacağı üzere bu yapı özellik çıkartma esnasında kullanılır. Birçok çalışmada yama tabanlı Siyam ağları kullanılır. Fakat, Siyam ağları kör bölgelerde (ill-posed regions) karşılıklı benzeşme bulmak için bağlam (context) bilgisinden yararlanamaz. PSMNet, bu sorunu çözmek için; He, Zhang, Ren ve Sun'un (2014) önermiş olduğu Uzamsal Piramit Havuzu (UPH) ve 3B CNN işlemi olmak üzere iki ana modülden oluşan bir piramit stereo eşleştirme ağ yapısıyla oluşturulmuştur.

UPH modülü, bir maliyet hacmi oluşturmak için kaynakları farklı ölçeklerde ve konumlarda kümeleyerek küresel kaynak bilgisinden yararlanır. UPH modülü CNN'nin sabit boyutlu kısıtlamasını ortadan kaldırmak için tasarlanmıştır. PSMNet'te UPH için dört adet sabit boyutlu ortalama havuzlama blokları tasarlanmıştır. Daha sonra maliyet hacmi hesaplaması için, UPH üzerinden elde edilen sol özellikler ve buna karşılık gelen sağ özellikler derinlik seviyesine göre birleştirilerek $H \times W \times D \times F$ olmak üzere dört boyutlu bir veri edilir. Burada, H yükseklik, W genişlik, D derinlik ve F özellik sayısını belirtir. Bu veri daha sonra 3B konvolüsyon işlemine sokulur.

3B CNN yapısı için bu çalışmada temel ve yığılmış kum saati adında iki farklı mimari önerilmiştir. Temel mimari, on iki adet artık bloklardan oluşmakta olup, peş peşe yerleştirilmiştir. Artık blokların yapısı da 3B CNN yapısıyla oluşturulmuştur. Maliyet hacmi bu katmanlarda örneklenerek, en son katmanda çift doğrusal ara değer (bilinear interpolation) yöntemiyle boyutu $H \times W \times D$ olacak bir veri elde edilir. Son olarak elde edilen bu veriye bağlanım (regression) uygulanarak, derinlik haritası $H \times W$ boyutunda olacak şekilde hesaplanır.

Yığılmış kum saati mimarisi ise kodlayıcı – kod çözücü yapısında olup, daha fazla kaynak bilgisinin öğrenilmesi için tasarlanmıştır. İki ara katman ve son katman olmak üzere toplam üç katmandan elde edilen derinlik haritaları kum saatine benzetilmiş ve yığın şekilde oluşturulduğu için “yığılmış kum saati” adı verilmiştir. Üç katmandan elde edilen

verilere ayrı ayrı çift doğrusal ara değeri bulma ve bağlanım uygulanarak derinlik haritaları elde edilmiştir. Bu derinlik haritaları için eğitim sürecinde ayrı ayrı kayıp değer hesaplaması yapılarak, elde edilen sonuçlar toplanmış ve ağ bu sonuçlara göre ağırlıklarını güncellemiştir. Doğrulama sürecinde ise sadece son katmandaki derinlik haritası dikkate alınmış ve kayıp değer direk olarak bu değere atanmıştır. Bu sebepten dolayı, PSMNet ağının eğitimi sonucunda elde edilen kayıp değerinin, test sonucunda elde edilen kayıp değere göre genelde daha yüksek çıkması beklenir. Başka bir deyişle, kayıp değer grafiklerinde eğitim değerlerinin test değerleri üzerinde seyretmesine yol açar. Bu çalışmada da FADNet'te olduğu gibi, kayıp değer hesaplaması için pürüzsüz L1 kayıp değer hesaplaması kullanılmıştır.

2.2. Veri Kümeleri

2.2.1. CATS veri kümesi

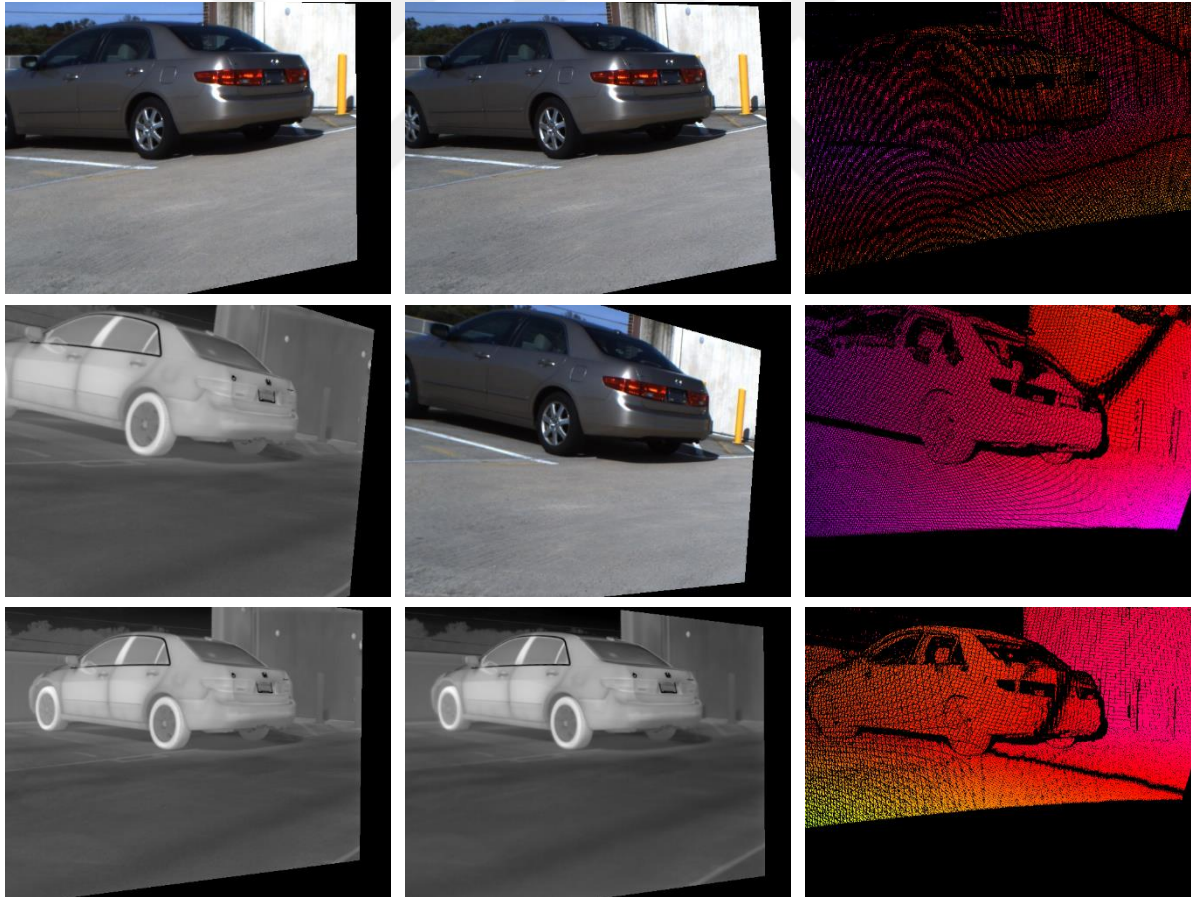
CATS veri kümesi, iki adet termal kamera, iki adet gündüz kamerası ve LiDAR cihazından oluşmaktadır. Düzenekte tam belli olmasa da sol kameralar ile sağ kameralar arasında bir yükselti farkı bulunmaktadır. Bu yükselti farkı dikey doğrultuda bir farklılığa sebep olmaktadır. Düzenek, Resim 2.1'den de görüleceği üzere sabit tutulmamıştır. Her ne kadar LiDAR cihazı kameraların ortasına konumlandırılmış olsa da kameralara göre pozisyonu sabit tutulmamıştır. Bazı sahnelerden veriler alınırken, LiDAR cihazı kameraların üstünde, bazı sahnelerde ise kameraların alt kısmında konumlandırılmıştır. Benzer şekilde, LiDAR cihazının kameralara göre sahneye daha yakın, bazılarında ise daha uzak olduğu tespit edilmiştir.



Resim 2.1. CATS veri kümesi elde etme düzeneği (Treible ve diğerleri, 2017; URL-6)

CATS veri kümesi, 80 dış mekân, 100 iç mekân görüntüsüyle oluşturulmuştur. Her sahne stereo renkli görüntü (RGB), stereo termal görüntü ve derinlik bilgisinden oluşmaktadır. LiDAR cihazıyla bir sahnenin taranması yaklaşık 8 dakika sürmektedir ve sahne yaklaşık 300 000 uzaysal noktalar ile örneklenir. Bu noktalar, 2 mm'den daha küçük hata payıyla elde edilir. LiDAR taramasının uzun sürmesi nedeniyle sahneler genellikle durağan şekilde oluşturulmuştur.

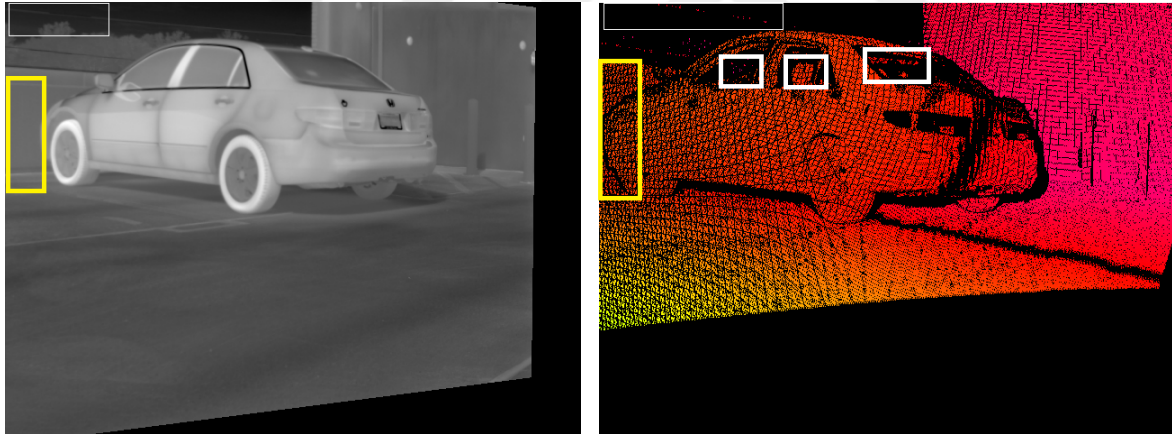
Bu veri kümesinde her çeşit uygulama için gerekli altyapı sunulmuştur. Bu kapsamda, RGB-RGB, termal-RGB ve termal-termal olmak üzere üç farklı uygulama kapsamında ayrı ayrı görüntüler arası doğrultma ve görüntülere uygun gerçek referans değer olarak derinlik haritası verilmiştir. Bu derinlik haritaları LiDAR tarafından taranan sahnedan elde edilen uzaysal noktaların 2B düzleme iz düşümünün alınmasıyla elde edilir.



Resim 2.2. CATS veri kümesinde bulunan örnek veri

Resim 2.2 incelendiğinde bu veri kümesinin stereo uygulamalar için uygun olmadığı anlaşılmaktadır. Düzenek stereo görüntü aldığı halde, görüntüler arasında doğrultma işlemi

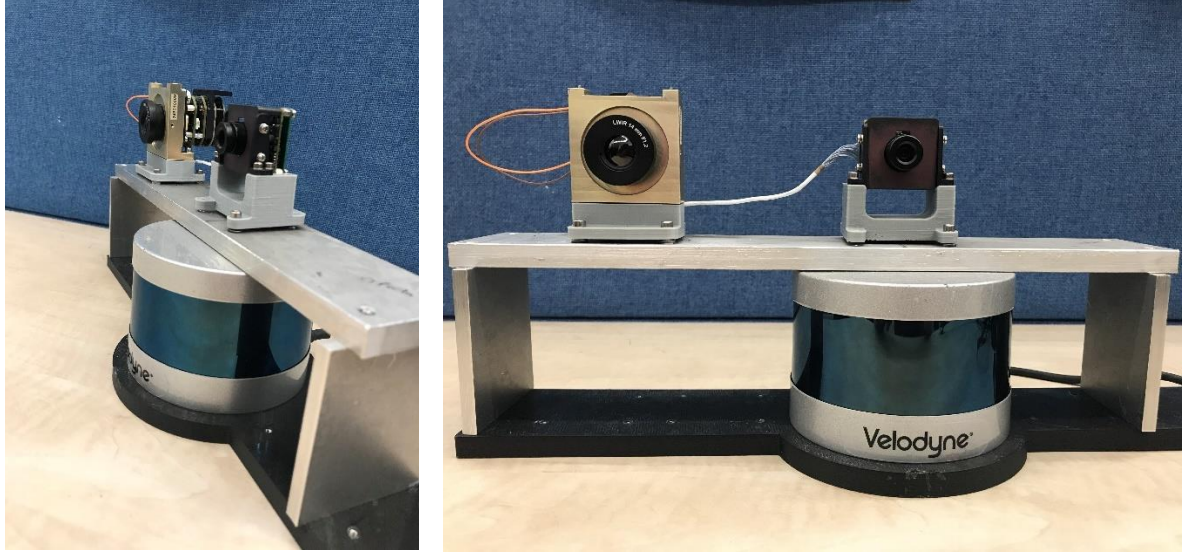
objeler özelinde yapılmıştır. Bundan dolayı, yakın olan objelerde yatayda kayma miktarının çok olması beklenirken kayma hiç olmamış, uzaktaki noktalar için kayma miktarı daha az veya hiç olmaması beklenirken, kayma miktarı fazla olmuştur. Bu doğrultma işlemi de stereo uygulamaları için uyumsuzdur. Öte yandan, gerçek referans değere bakıldığı zaman, üzerindeki bazı örneklemelerin görüntüye göre taşıdığı, bazı örneklemelerde de derinlik farkı olduğu gözlenmektedir. Örneğin Resim 2.3'e bakıldığında, termal-termal görüntü çifti ele alınacak olursa, gerçek referans değerde arabanın ön-sol kısmı görüntünün sol kenarından arada boşluk olmaksızın başlamıştır. Fakat termal görüntülere bakıldığında arada boşluk bulunmaktadır (Sarı kutucuk). Arabanın camlı kısımlarında derinlik arabanın derinliğine göre farklıdır (Beyaz kalın kutucuk). Bunun sebebi LiDAR cihazının mesafe ölçümünde camdan yansımayı değil camdan sonra gelen nesnenin yansımaları örnek olarak kabul etmesinden dolayıdır. Yine arabanın arkasındaki duvar üstünden geçen iki tele bakıldığında, üstteki telde tutarsızlık mevcuttur (Beyaz ince kutucuk).



Resim 2.3. CATS veri kümesindeki gerçek referans değer uyumsuzluğu

2.2.2. Oluşturulan veri kümesi

Veri alma düzeneği, Velodyne (VLP-16) LiDAR, 1 adet soğutmasız termal dedektör ve 1 adet de gündüz kamerası kullanılarak hazırlanmıştır. Bu kameraların objektifleri ve donanım tasarımları ASELSAN tarafından yapılmış olup, bilgisayar ile haberleşme ve görüntü aktarım işlemleri Ethernet üzerinden yapılmaktadır. Gündüz kamerası ile LiDAR cihazı, gerçek referans değeri için iz düşüm alınırken kolaylık olsun diye aynı dikey doğrultuda konumlandırılmıştır.



Resim 2.4. Veri kümesi elde etme düzeneği

Kullanılan LiDAR cihazı, 100 metre menzile sahip olup, dikeyde 2° açısal çözünürlüklü 30° görüş alanı, yatayda $0,1^\circ$ - $0,4^\circ$ arası değişen açısal çözünürlüklü 360° görüş alanına (FOV) sahiptir. Bu cihazın hata payı ± 3 cm olup, dönme hızı 5-20 hz arasında değişen frekanstadır. Kameralardan alınan görüntüler, 680×509 piksel çözünürlüğündedir ve birbirleri arasında 3° görüş alanı farkı vardır.

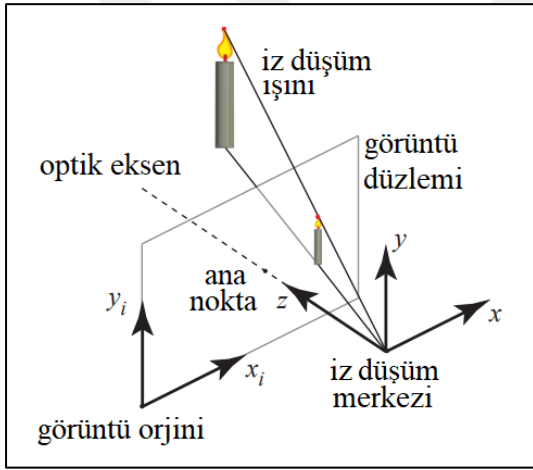
Düzenegin bilgisayar ile bağlantıları bir Ethernet modemi aracılığıyla sağlanmıştır. Kamera donanımından gelen Ethernet kablosu ile LiDAR cihazından gelen Ethernet kablosu modeme bağlanmış, modem ile bilgisayar arasında da yine Ethernet kablosu üzerinden haberleşme sağlanmıştır. Sahneler Python tabanlı yazılan bir yazılım ile alınmıştır. Görüntüler PNG formatı ile alınırken, LiDAR verileri uzaysal noktalardan oluştuğu için CSV formatı ile alınmıştır. Daha sonra CSV formatındaki, sahneyi oluşturan bu uzaysal noktalar, yine Python tabanlı yazılan bir uygulama ile bir sonraki bölümde bahsedilecek olan Pinhole kamera modeli baz alınarak, 2B görüntü düzlemine iz düşümü elde edilmiş ve gerçek referans değerler oluşturulmuştur. 170'i eğitim, 20'si test maksadıyla toplam 190 görüntü alınmıştır. Resim 2.5'ye bakıldığında, elde edilen gerçek referans değerinde bulunan örnek sayısı, literatürde kullanılan gerçek veri kümelerindeki gerçek referans değerler ile karşılaştırıldığında oldukça yetersiz sayıda olduğu görülmektedir.



Resim 2.5. Oluşturulan düzeneden alınan örnek veri

2.3. Pinhole Kamera Modeli

Pinhole kamera modeli (Trucco ve Verri, 1998: 24), uzaysal (3B) noktaların 2B düzleme aktarılmasında kullanılan en yaygın metottur. Burada, merkez bir nokta belirlenir. Görüntü düzlemi, iz düşüm alınacak noktalar ile merkez nokta arasında konumlandırılıp düzlem boyutu için istenilen genişlik ve yükseklik belirlenir. Daha sonra iz düşümü yapılacak nokta ile merkez nokta arasında doğru çizilip, görünüş düzleminde nereden kestiğine bakılır. Bu şekilde düzlemin görüş açısına göre tüm uzaysal noktaların görüntü düzlemine iz düşürülerek, görüntü düzleminde nihai görüntü elde edilmiş olur. Bu yöntem Şekil 2.5'te görselleştirilmiştir.



Şekil 2.5. Pinhole kamera modellemesi (URL-7)

Uzaydaki bir noktanın 2B görüntü düzlemine karşılık gelen noktası, üçgen benzerlikleriyle bulunur.

$$x_i = x \frac{W}{2z \tan \frac{Q_x}{2}} \quad (2.1)$$

$$y_i = y \frac{H}{2z \tan \frac{Q_y}{2}} \quad (2.2)$$

$$P_{im}[x_i, y_i] = z \quad (2.3)$$

Burada, P_{im} 2B görüntü düzlemini; Q_x ve Q_y yatayda ve dikeyde bakış alanı açılarını; H ve W sırasıyla 2B görüntü düzleminin yükseklik ve genişliğini; x , y , z uzay koordinatlarını ve x_i , y_i ise 2B görüntü koordinatlarını ifade eder.

Eş. 2.1 ve Eş. 2.2 denklemleri kullanılarak uzaydaki bir noktanın 2B görüntü düzleminde hangi noktaya karşılık geleceği belirlenir. Daha sonra elde edilen bu noktaya, Eş. 2.3 denklemi kullanılarak, derinliği ifade eden z değeri atanır.

2.4. KITTI Piksel Hata Hesaplaması

KITTI piksel hata hesaplaması, literatürde derinlik tahmini çalışmalarında sıkça kullanılan performansı belirten bir metriktir. Bu metriğin sonucu sifıra ne kadar yakınsa, tahmin o derece tutarlı ve doğru kabul edilir. Bu hata hesaplaması özellikle gerçek referans değerler ayırık örnekleli olması söz konusuysa kullanılır. Çünkü ayırık örnekleli gerçek referans değerlerde, geçerli örnekler filtrelendir. Daha sonra filtrelendirilmiş görüntü pozisyonları aynı şekilde tahmin edilen görüntü pozisyonlarıyla karşılaştırılır ve hata oranı hesaplanır.

Hata hesaplaması sırasıyla şu şekilde gerçekleşir. Önce gerçek referans değeri (G_{grad}) ile tahmin edilen görüntünün (G_{tahmin}) mutlak fark değeri alınır (Eş. 2.4).

$$D_{fark} = |G_{grad} - G_{tahmin}| \quad (2.4)$$

Daha sonra bu değeri (D_{fark}) iki aşamada kullanılır. İlk aşamada bu değeri, kaç piksellik bir fark hata olarak kabul edilecekse o piksel değeri (S_p) ile karşılaştırılır (Eş. 2.5). Eğer fark büyükse “mantıksal doğru”, küçükse “mantıksal yanlış” olarak bir boole dizide (A_{piksel}) tutulur.

$$A_{piksel} = (D_{fark} > S_p) \quad (2.5)$$

İkinci aşamada ise bu değerin gerçek referans değeriyle oranı bulunup (Eş. 2.6) bu oran (D_{oran}), yüzde kaçlık eşik değeri belirlenecekse, o eşik değeri (S_e) ile karşılaştırılır (Eş. 2.7). Yine fark büyükse “mantıksal doğru”, küçükse “mantıksal yanlış” olarak başka bir boole dizide ($A_{eşik}$) tutulur.

$$D_{oran} = \frac{D_{fark}}{|G_{grad}|} \quad (2.6)$$

$$A_{eşik} = (D_{oran} > S_e) \quad (2.7)$$

Gerçek referans değerin örneklerinin ayırık olmasından dolayı, örnek olmayan yerler sıfır kabul edilmiştir ve bu sıfırların hata hesaplamasına dahil olmaması gereklidir. Bu sebeple gerçek referans değerde sıfır olmayan değerler “mantıksal doğru”, sıfır olanlar “mantıksal yanlış” olarak yine iki boyutlu bir boole dizide (A_{maske}) tutulur (Eş. 2.8). Bu diziye de maske adı verilir.

$$A_{maske} = |G_{grd} > 0| \quad (2.8)$$

Elde edilen üç dizi (A_{piksel} , $A_{eşik}$ ve A_{maske}) “mantıksal ve” işlemine sokulur (Eş. 2.9).

$$A_{sonuç} = A_{piksel} \& A_{eşik} \& A_{maske} \quad (2.9)$$

Elde edilen dizide ($A_{sonuç}$) bulunan “mantıksal doğru” toplam sayısı, maske dizisindeki “mantıksal doğru” olan değerlerin toplam miktarına bölünmesiyle bu hata (ε) hesaplanmış olur (Eş. 2.10). Eş. 2.10’daki toplam sembollerinde belirtilen H ve W görüntünün boyutlarını ifade etmektedir.

$$\varepsilon = \frac{\sum_{n=1}^H (\sum_{m=1}^W (A_{sonuç}[n,m] > 0))}{\sum_{n=1}^H (\sum_{m=1}^W (G_{grd}[n,m] > 0))} \quad (2.10)$$

2.5. McNemar Test Sınaması

Literatürde birçok istatistiksel testler bulunmaktadır. Bu testlerin birçoğu 2x2’lik kontenjans tablosunu kullanır. Bu tablo, iki farklı test için elde edilen sonuçların doğru veya yanlış olarak 2x2’lik bir tabloda tutulmasıyla elde edilir. Bu tabloya örnek Şekil 2.6’da gösterilmiştir.

	model 1 doğru	model 1 yanlış
model 2 doğru	a	b
model 2 yanlış	c	d

Şekil 2.6. 2x2’lik kontenjans tablosu

Şekil 2.6'daki tabloya bakıldığında, a; her iki modelinde doğru tahmin ettiği değeri, b; model 1'in yanlış, model 2'nin doğru, c; model 1'in doğru, model 2'nin yanlış ve son olarak d; her iki modelinde yanlış tahmin ettiği değerleri göstermektedir. McNemar (1947) yaptığı çalışmada, bu tabloyu kullanarak iki modelin bağımsız olup olmadığını sınamaktadır. Bu sınama, elde edilen verileri gerçek referans değere göre karşılaştırmak yerine, verileri iki model arasında karşılaştırmak için sunulmuştur. Buradan hareketle McNemar, her iki modelin birbirinden farklı kararlar verdiği değerleri kullanarak Eş. 2.11'deki denklemi ortaya koymuştur.

$$\chi^2 = \frac{(b - c)^2}{b + c} \quad (2.11)$$

Bu denklem sonucunda elde edilen değer ki-kare dağılım grafiğindeki olasılıksal karşılık değerine bakılır. Bu değer daha önce belirlenen bir eşik değerinden büyükse, bu iki model birbirine bağımlıdır. Değer, eşik değerinden küçükse iki model birbirinden bağımsızdır sonucu çıkarılabilir. Bu çalışmada FADNet ve PSMNet ağlarından elde edilen sonuçların birbirleriyle kıyaslanmasında McNemar sınaması kullanılmıştır.



3. ÖNERİLEN YÖNTEM

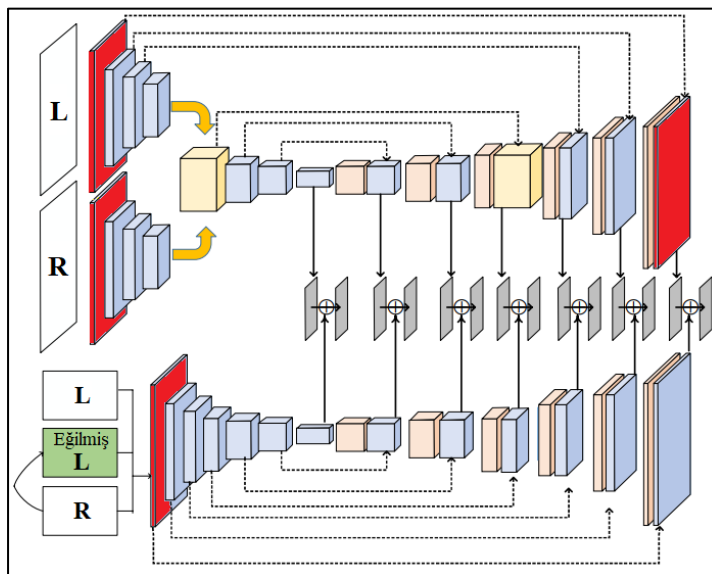
Deneysel çalışmalara geçmeden önce, bu işlemi doğasına uygun bir şekilde uyarlamak gerekir. Buradan hareketle hem ağ yapılarının hem de veri kümelerinin birbirine uyumlu olması gerekmektedir.

3.1. Ağ Yapılarındaki Değişiklikler

Kullanılan ağ yapıları, görünür spektrum (RGB) görüntülerinden beslendiği için giriş görüntü katmanlarında giriş verisi 3 kanallı olarak kullanılmaktadır. Termal spektrumda, görüntüler daha önce de bahsedildiği gibi gri ölçeklidir. Bu nedenle, RGB görüntülerinin kullanıldığı her iki ağın katmanları tek kanallı girişi sağlayacak şekilde değiştirilmelidir. Bu kapsamda her iki ağ için bazı değişiklikler yapılmıştır.

3.1.1. FADNet yapısındaki değişiklikler

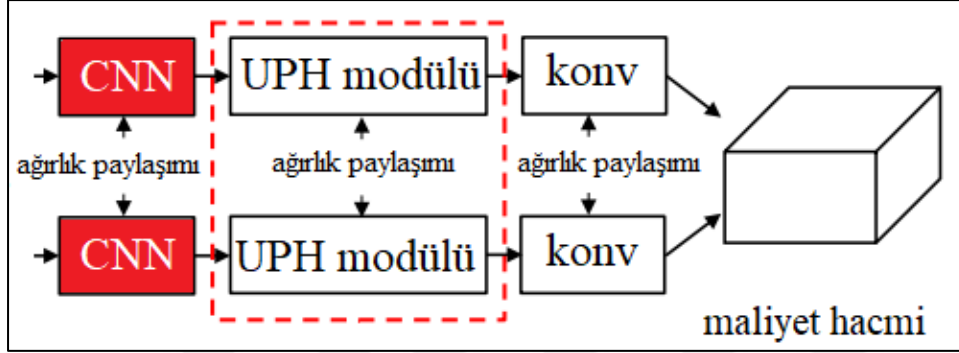
FADNet ağ yapısında üç değişiklik vardır. İlk olarak, DispNetC kolunda, ilk konvolüsyon katmanı üç filtre boyutundan bir filtre boyutuna indirilmiştir. İkinci olarak, aynı kolda, son konvolüsyon katmanı 20 filtre boyutundan 18 filtre boyutuna indirilmiştir. Üçüncü değişiklik DispNetS kolundadır. Bu kolda, giriş filtresi boyutu 11'den 5'e düşürülmüştür. Bu değişiklikler, Şekil 3.1'de gösterildiği gibi kırmızı ile boyanmış kısımlarda yapılmıştır.



Şekil 3.1. FADNet ağ yapısında değişikliğin yapıldığı katmanlar

3.1.2. PSMNet yapısındaki değişiklikler

PSMNet ağı yapısında sadece bir değişiklik bulunmaktadır. Özellik çıkarma kısmında ilk katmandaki filtre boyutu üçten bire düşürülmüştür. Bu değişiklik, Şekil 3.2.’de gösterildiği gibi kırmızı ile boyanmış kısımda yapılmıştır.



Şekil 3.2. PSMNet ağı yapısında değişikliğin yapıldığı katmanlar

3.2. Veri Kümeleri Üzerindeki Değişiklikler

3.2.1. CATS veri kümesinde yapılan değişiklikler

Daha önce Bölüm 2.2.1’de bahsedildiği gibi, CATS veri kümesinde uyumsuzluklar bulunmaktadır. Bu uyumsuzlukları gidermek adına, CATS veri kümesindeki dış mekân termal – termal görüntü çiftlerine aşağıdaki işlemler ayrı ayrı uygulanmıştır.

- Görüntü doğrultma
- İz düşüm alma
- LiDAR verilerinde kaydırma ve döndürme
- Hatalı örnek kaldırma

Görüntü Doğrultma

CATS veri kümesindeki dış mekândan alınan herhangi bir termal stereo görüntülerin dikey doğrultuda doğrultulmamış olduğu gözlenmektedir. Bu çalışmada Mallon ve Whelan’ın (2005: 643) önermiş olduğu “*Temel matrizen Perspektif Düzeltme*” isimli metot doğrultma işlemi için referans alınmıştır. Resim 3.1 incelendiğinde, sol termal görüntüde bulunan

adamın kafasının hemen üstünden sağ termal görüntüye doğru kırmızı ile renklendirilen düz bir doğru çizildiğinde, sarı ile renklendirilen dik doğru kadar bir fark söz konusudur.



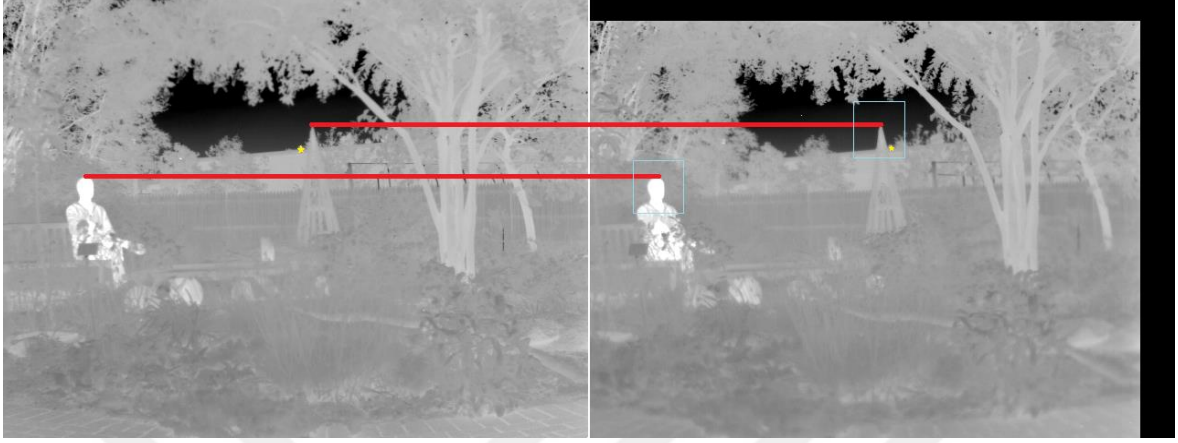
Resim 3.1. Doğrultma işlemi yapılmamış sol ve sağ termal görüntüler

Stereo görüntüler için yatay doğrultuda herhangi bir doğrultma işlemine gerek yoktur. Çünkü, stereo görüntülerin doğası gereği, sol görüntüdeki bir nokta, sağ görüntüde yatay doğrultuda farklı noktada olacağı aşıkardır. Daha önce de bahsedildiği gibi, zaten bu fark bize o noktanın derinliğini verir. Fakat yine de bu çalışmada, LiDAR cihazının ölçebildiği en uzak noktaların yatay doğrultuda piksel kayması sıfır olarak tayin edilmiştir. Bu nedenle doğrultma işlemi olarak aşağıdaki adımlar izlenmiştir.

- LiDAR cihazının ölçebildiği en uzak mesafenin olduğu görüntünün tespit edilmesi
- Tespit edilen bu görüntü üzerindeki en uzak piksele ait 2B koordinat bilgisinin tespit edilmesi
- Sol görüntüyü sabit alıp sağ görüntüyü kaydırarak, her iki görüntüde de bu pikselin aynı koordinat konumuna getirilmesi
- Sağ görüntüde hem yatayda hem de dikeyde kaydırılan piksel miktarının tespit edilmesi
- Tespit edilen piksel kayma miktarlarının veri kümesi içerisindeki tüm sağ görüntülere uygulanması

Bu adımlara göre sağ görüntüdeki kayma miktarları yatayda 38 piksel sola, dikeyde 28 piksel aşağıya olacak şekilde tespit edilmiştir. Resim 3.2' ye bakıldığında doğrultma işlemi sonunda, görüntülerin stereo mantığına uygun hale geldiği gözlenmiştir. Tüm veri

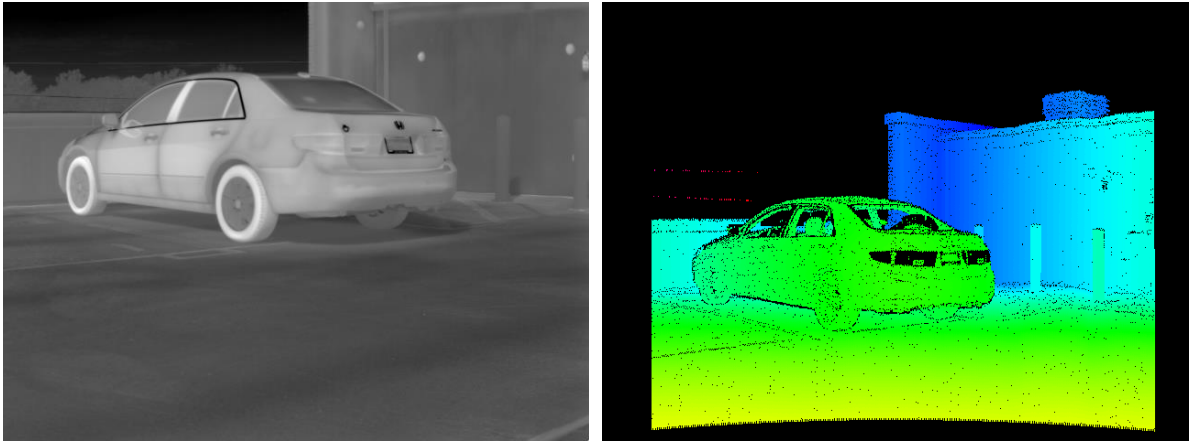
kümesinde LiDAR cihazının ölçebildiği en uzak nokta, sarı renk ile gösterilmiştir. Yataydaki hizalama bu noktaya göre yapılmıştır.



Resim 3.2. Doğrultma işlemi tamamlanmış sol ve sağ termal görüntüler

İz düşüm alma

İz düşüm, uzaydaki noktaların 2B düzlemde nereye yerleşeceği anlamına gelmektedir. Bu uygulama için Bölüm 2.3'te belirtilen Pinhole kamera modeli kullanılmıştır. İşlem sonucunda elde edilen görüntü, gerçek referans değer olarak kullanılmak üzere derinlik haritasını oluşturur. Yapılan işleme örnek Resim 3.3'te gösterilmektedir.



Resim 3.3. Pinhole kamera modeli kullanılarak elde edilmiş 2B iz düşüm görüntüsü

Bu görüntü incelendiğinde, derinlik haritasındaki arabanın pozisyonu referans alınan görüntüye (sol termal görüntü) göre konumunun farklı olduğu bariz gözükmemektedir. Bu

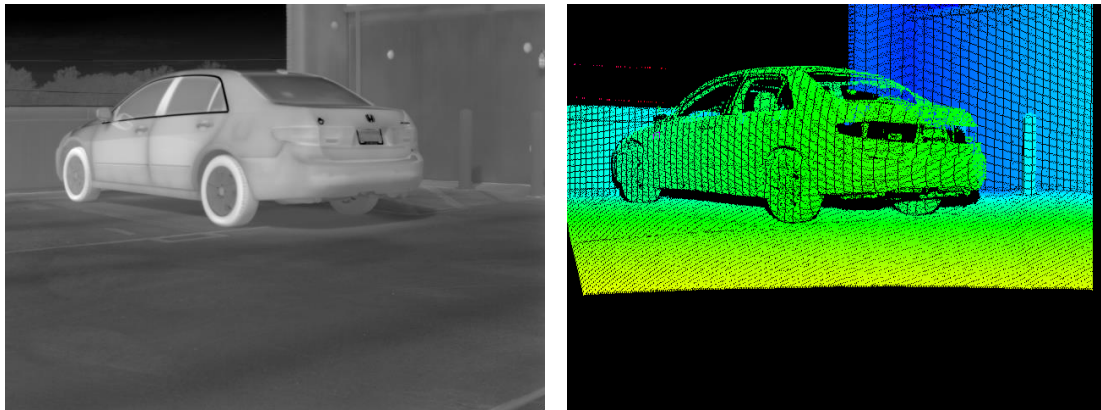
sebeple, iz düşüm alınmadan önce LiDAR verilerinde kaydırma ve döndürme işlemlerinin yapılması gereklidir.

LiDAR verilerinde kaydırma ve döndürme

LiDAR verilerinde kaydırma ve döndürme işlemi, LiDAR cihazının merkez noktası ile referans alınan kameranın merkez noktalarını aynı konuma getirmek, bu sayede LiDAR cihazından alınan verilerin sanki kameradan alınıyormuşçasına referans alınan görüntüye göre uyumluluğunu arttırmak için yapılır. Böylelikle, bu işlem sayesinde gerçek referans değer elde edilirken öğrenme oranını artırılır ve hatalı öğrenmenin de önüne geçilmiş olur. LiDAR verilerinde kaydırma ve döndürme işlemi matris işlemleriyle yapılır.

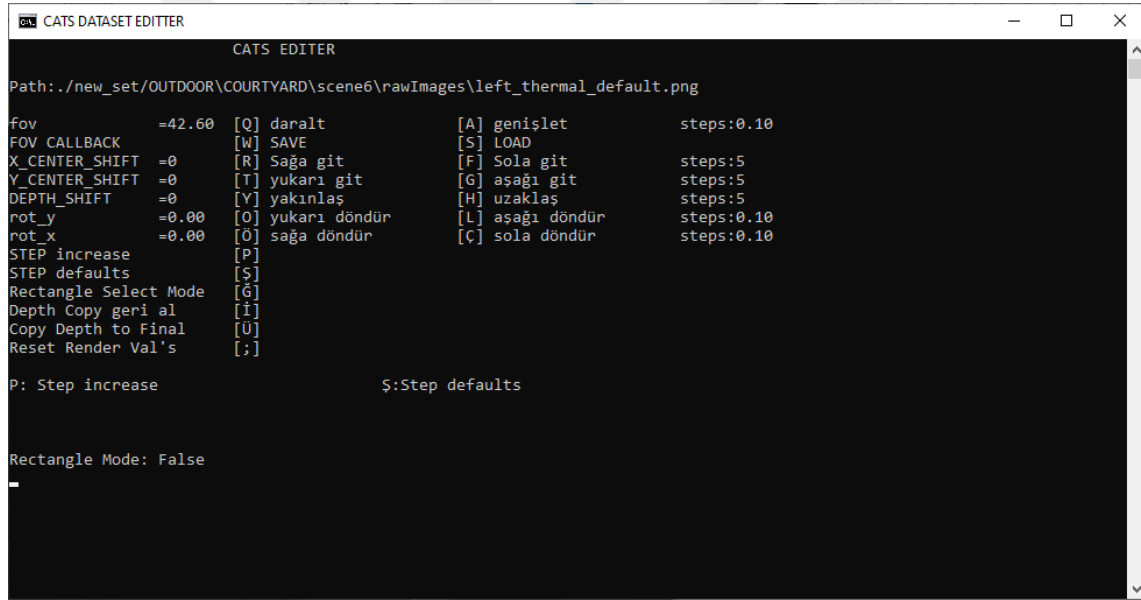
$$P_{cam} = R(P_{lid} - t) \quad (3.1)$$

Burada referans alınan kameranın uzay koordinatına göre kaydırılmış ve döndürülmüş uzay kümesi P_{cam} , LiDAR verilerine P_{lid} , kaydırma vektörü t , 3B döndürme matrisi de R olarak ifade edilmiştir. Eş. 3.1 denklemi ile elde edilen yeni uzaysal noktaları kullanarak, bir önceki bölümde de bahsedildiği üzere 2B görüntü düzlemine Pinhole kamera modeliyle tekrar iz düşüm alınır. Oluşturulan yeni görüntüyle referans alınan görüntünün birbirlerine olan uygunluğuna bakılır. Eğer fark var ise t ve R parametrelerini güncelleyerek en optimum eşleme elde edilmeye çalışılır. Bu işlem ICP metodu (Arun, Huang ve Blostein, 1987: 698) kullanılarak yapılmaya çalışılmış fakat başarılı sonuç elde edilememiştir. Dolayısıyla bu işlem veri kümesindeki her görüntü için manuel olarak yapılmıştır. Resim 3.4'te bu uygulama sonucu elde edilen en optimum görüntü gösterilmiştir.



Resim 3.4. Kaydırma ve döndürme işlemi sonrası iz düşüm alınan gerçek referans değer görüntüsü

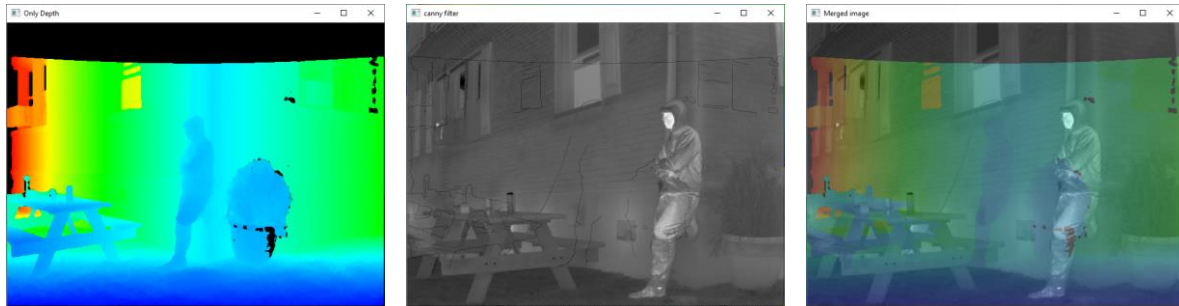
Bu işlem, Python tabanlı bir uygulama ile yapılmıştır. Şekil 3.3'te bu kapsamda yazılan uygulamanın parametrelerinin olduğu kullanıcı arayüzü gösterilmiştir. Bu arayüz sayesinde t ve R parametreleri istenilen seviyede ayarlanabilir yapılmıştır. Burada, “fov” görüş alanını, “X_CENTER_SHIFT”, “Y_CENTER_SHIFT” ve “DEPTH_SHIFT” t parametresinin katsayılarını belirleyen uzay koordinatında kaydırma miktarlarını, “rot_x” ve “rot_y” R parametresinin katsayılarını belirleyen döndürme açılarını ifade eder. Başlangıç değeri olarak “fov” değeri 42,60 olarak, diğer değerler 0 olarak atanmıştır. Klavye vasıtasıyla bu değerler en sağda “steps” ile belirtilen miktara göre artırılır veya azaltılır. Ayrıca arayüze katsayıları kaydetme, geri yükleme, bölgesel olarak çalışma (Rectangle mode) gibi işlemeyi daha doğru ve hızlı yapmak adına çeşitli özellikler eklenmiştir.



Şekil 3.3. Kaydırma ve döndürme işlemi için Python tabanlı yazılan yazılımın menüsü

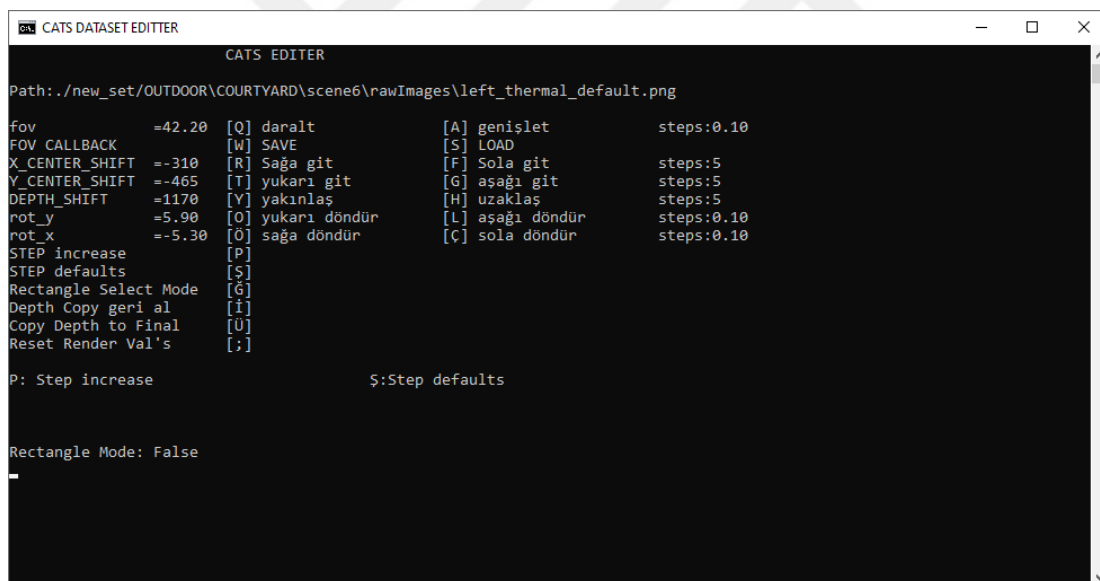
Resim 3.5'te yine yazılan bu uygulamanın anlık olarak sonuç görüntüleri gözükmemektedir. Burada üç adet çıktı görüntüsü gözükmemektedir. İlk görüntü, uzaysal noktaların belirlenen t ve R parametreleri ile alınmış 2B iz düşüm görüntüsünü göstermektedir. İkinci görüntü, elde edilen iz düşüm görüntüsünün Canny (1986) kenar filtresinden geçirilip elde edilen kenar görüntüsünü, referans görüntü üzerine eklenmesiyle oluşturulmuş görüntüdür. Buradaki maksat, t ve R parametreleri için optimum değerler bulunurken, bir yandan da bu kenar bilgilerinin görüntü üzerinde uyumunun daha net gösterilmesidir. Son görüntü ise ilk

görüntünün belirli bir ağırlıkla referans görüntü üzerine bindirilmesidir. Bunun amacı da tıpkı ikinci görüntüde olduğu gibi eşlemeyi kolaylaştırmak içindir.

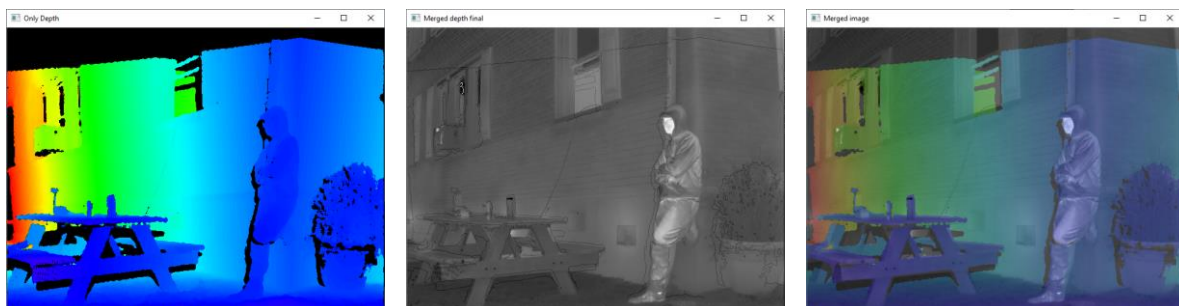


Resim 3.5. Kaydırma ve döndürme yapılmamış görüntüler

Şekil 3.4'te denemeler sonucu en uygun eşleme yapılan parametre katsayıları yazılım arayüzü üzerinden gösterilmiştir. Elde edilen sonuçlar Resim 3.6'da gösterilmiştir.



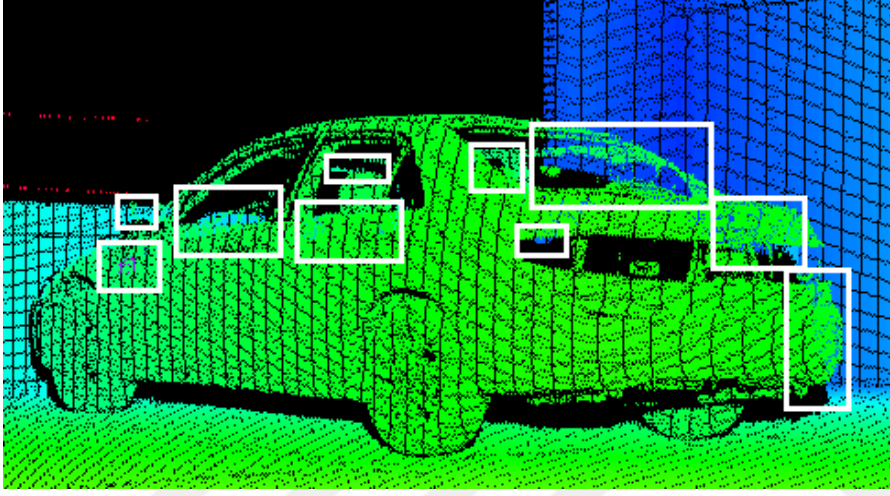
Şekil 3.4. Kaydırma ve döndürme işlemi için optimum katsayılar



Resim 3.6. Kaydırma ve döndürme yapılmış görüntüler

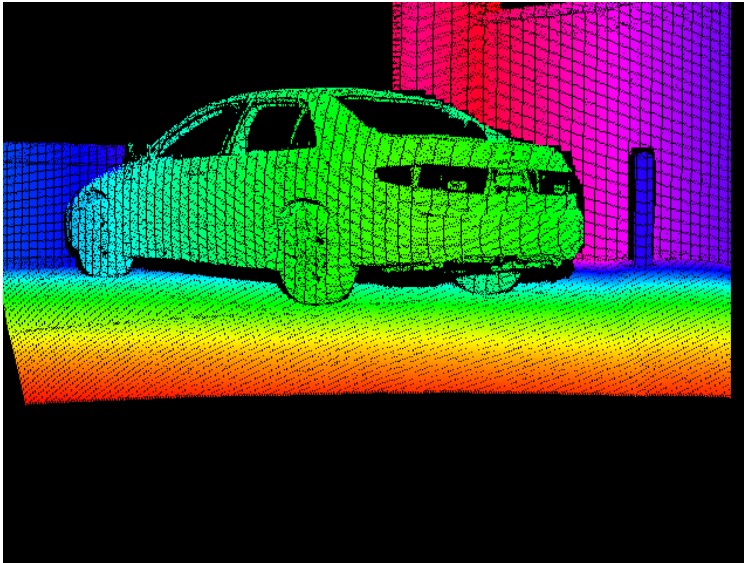
Hatalı örnek kaldırma

LiDAR cihazından elde edilen veriler ayrıık örneklemelerle oluşturulduđu için uzayda bulunan daha uzaktaki bir noktanın iz düşümü alındığında daha yakın noktalar arasına yerleşmesi söz konusu olabilir. Bu durumda, bu noktaların silinmesi, denetimli öğrenme modelinde hatalı öğrenmelerin önüne geçilmesi bakımından gereklidir. Şekil 3.5'te beyaz kutucuk içinde bu durumun gözlemlendiği bölgeler gösterilmiştir.



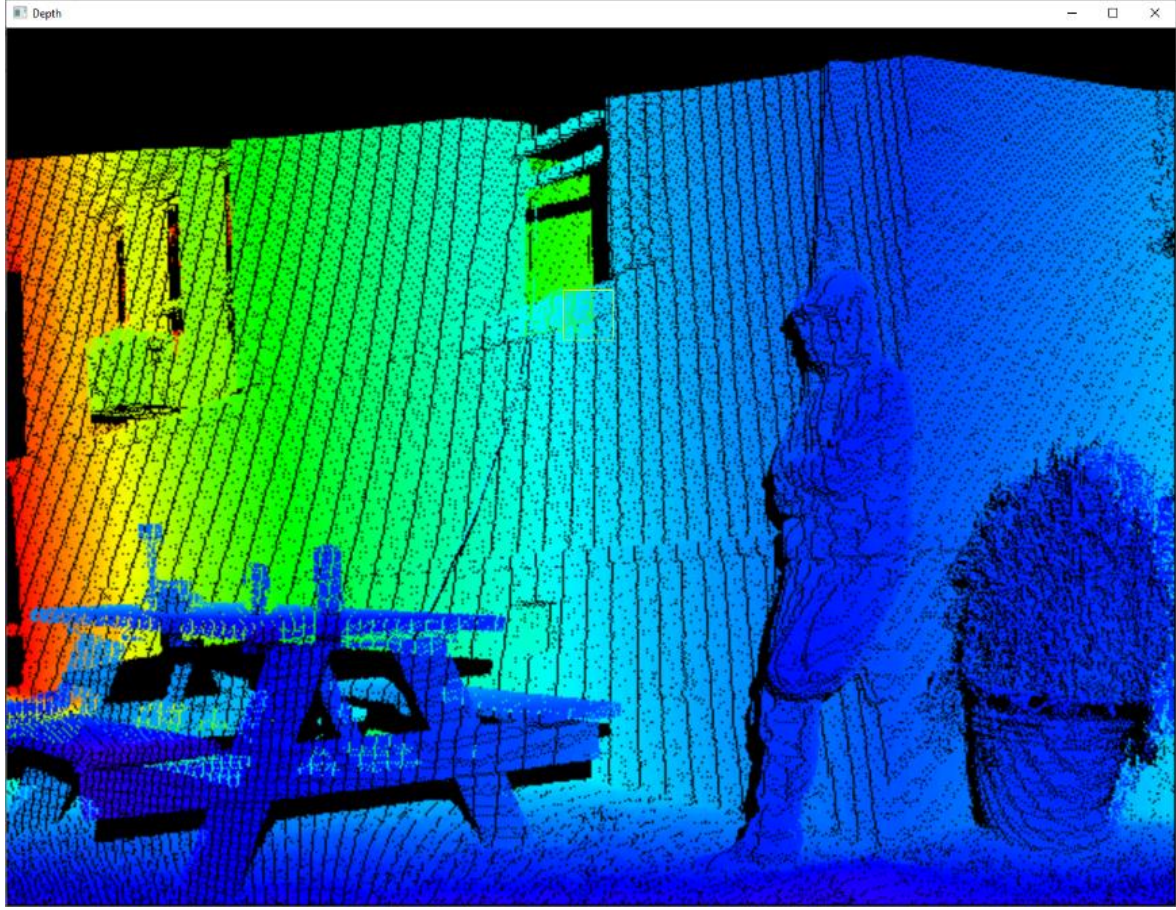
Şekil 3.5. Hatalı örneklerin kutucuklar içinde gösterilmiş olduğu görüntü

Python dilinde yazılan bir uygulama ile hatalı örneğe sebep olacak bu noktalar manuel olarak silinmiştir. Resim 3.7'de bu sorunların giderildiği görüntü gösterilmiştir.



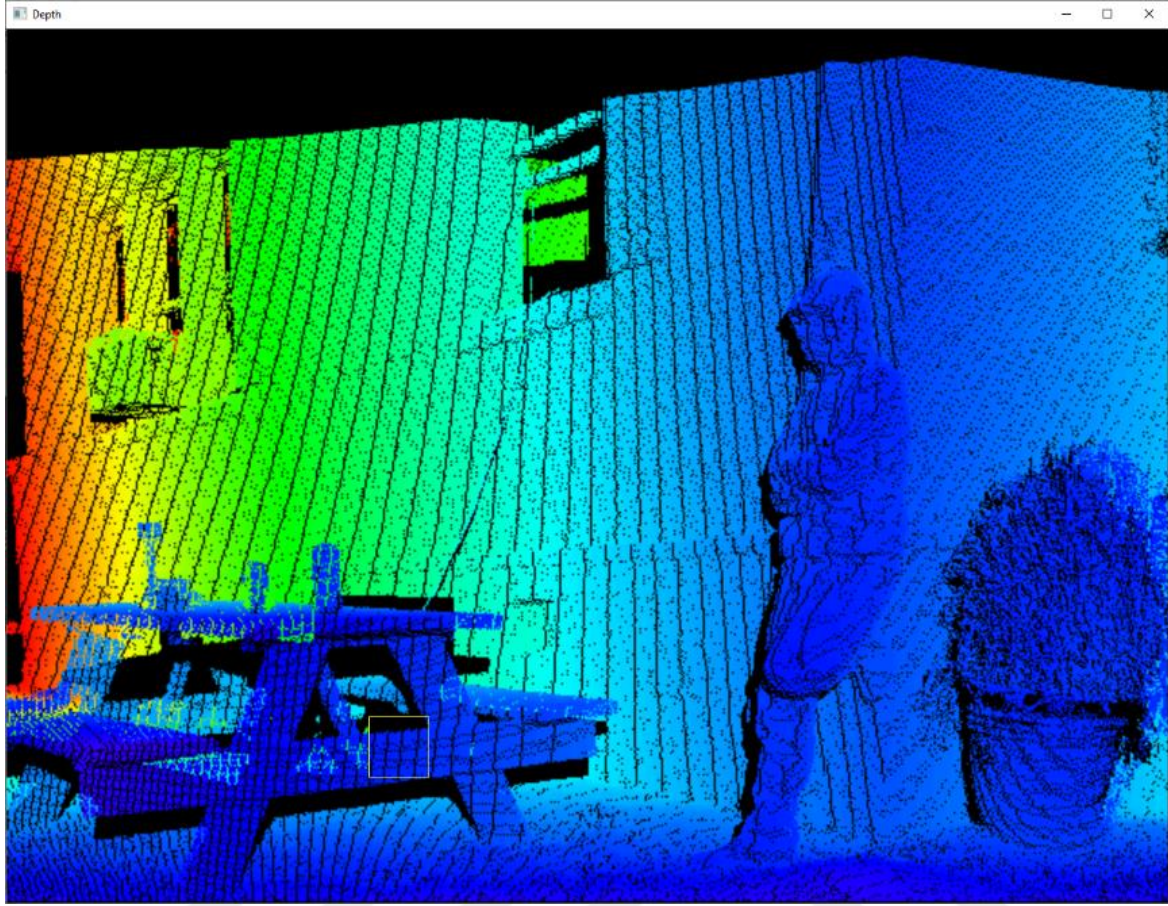
Resim 3.7. Hatalı örneklerin giderilmiş olduğu görüntü

Bu işlemin nasıl yapıldığı başka bir örnek üzerinden gösterilmiştir. Resim 3.8'e bakıldığında masanın içinde, arkasındaki duvara ait derinlik örnekleri gözlenmektedir.



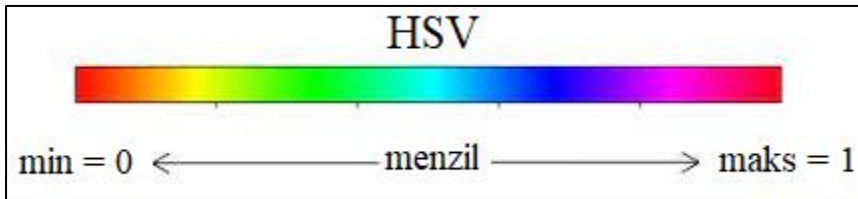
Resim 3.8. Kaydırma ve döndürme işlemi sonrası elde edilmiş görüntü

Bu hatalı örnekleri kaldırmak için, Python tabanlı bir uygulama ile görüntüde hatalı örnekler silinir. Bu uygulamada, farenin kaydırma tuşu ile sarıyla gösterilen bir dörtgen kutucuğun boyutları artırılıp, azaltılır. Farenin sol tuşuna basıldığında, sarı kutucuk içinde kalan örneklerin ortalaması hesaplanır. Bu değer, klavyeden dinamik olarak belirlenen bir eşik değerine göre kıyaslanır ve daha uzağı ifade eden örnekler, eğer bu eşik değerinin üstündeyse silinir. Bu şekilde yakın objelerin örnekleri muhafaza edilirken, uzaktaki objelerin örnekleri kaldırılmış olur. Ayrıca, bazı durumlarda gerekliliğinden dolayı farenin sağ tuşuna basıldığında ise sarı kutucuk içinde kalan tüm örnekler silinir. Resim 3.9'da bu uygulamaya örnek gösterilmiştir. Resimde masanın üst sağ kısmı ve yakın pencere kısmı silinmiş, sarı kutucuk masanın alt kısmında silme işlemini yapmaktadır.



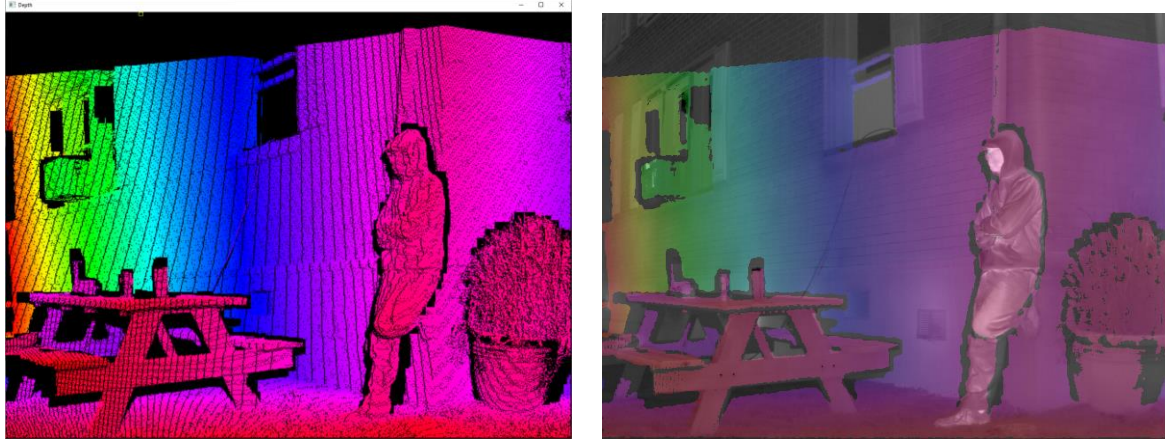
Resim 3.9. Hatalı örneklerin silinmesi işlemi

Buradaki renklendirme işleminde HSV renk dağılımı uygulanmıştır (Şekil 3.6). Bu renklendirme, her görüntü özelinde yapılır. Bu sebeple hatalı örnekler silinirken görüntü üzerinde renk değişimi meydana gelmektedir. Bunun faydası, yapılan işlemin doğruluğunun gözle görülmesini kolaylaştırmak içindir.



Şekil 3.6. HSV renk dağılımı

Resim 3.10'a bakıldığında hatalı örneklerin silinmesi işleminin tamamlanmış görüntüsü ilk görüntüde, elde edilen görüntünün referans görüntüsüne bindirilmesiyle elde edilmiş görüntü ise ikinci görüntüde gösterilmiştir.



Resim 3.10. Hatalı örneklerin silinmesi işlemi sonrası elde edilmiş görüntü

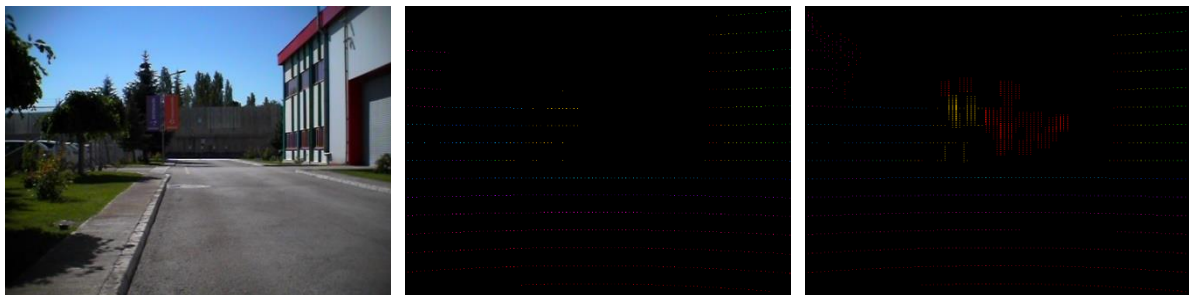
3.2.2. Oluşturulan veri kümesinde yapılan değişiklikler

Daha önce Bölüm 2.2.2’de de bahsedildiği gibi, oluşturulan veri kümesinde derinlik örneklemelerinde yetersizlik bulunmaktadır. Bu örneklemeleri arttırmak adına aşağıdaki işlemler yapılmıştır. Oluşturulan veri kümesinde ayırık örnek artırma işlemi tüm görüntüler için uygulanırken, örnek doldurma işlemi sürecin zorluğu ve meşakkati sebebiyle sadece 30 görüntüye uygulanabilmektedir.

- Ayırık örnek artırma
- Örnek doldurma

Ayırık örnek artırma

Python tabanlı bir uygulama yazılarak örnek sayısı, bulunulan objeye göre manuel olarak artırılmıştır. Bu adım sonunda elde edilen görüntü Resim 3.11’de gösterilmiştir.



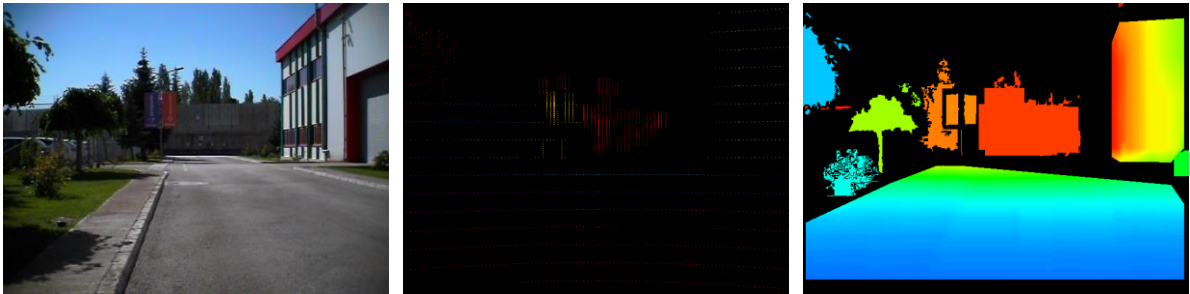
Resim 3.11. Örnek artırma işlemi; Sol: RGB görüntüsü, Orta: Örnekleri sadece LiDAR ile edilmiş görüntü, Sağ: Örnekleri artırılmış görüntü

Örnek doldurma

Arttırılan ayırık görüntüyü kullanarak Python tabanlı uygulamayla örnek sayısının daha da artırılması için örnek doldurma işlemi yapılmıştır. Bu işlem aşağıdaki adımlar uygulanarak yapılmıştır.

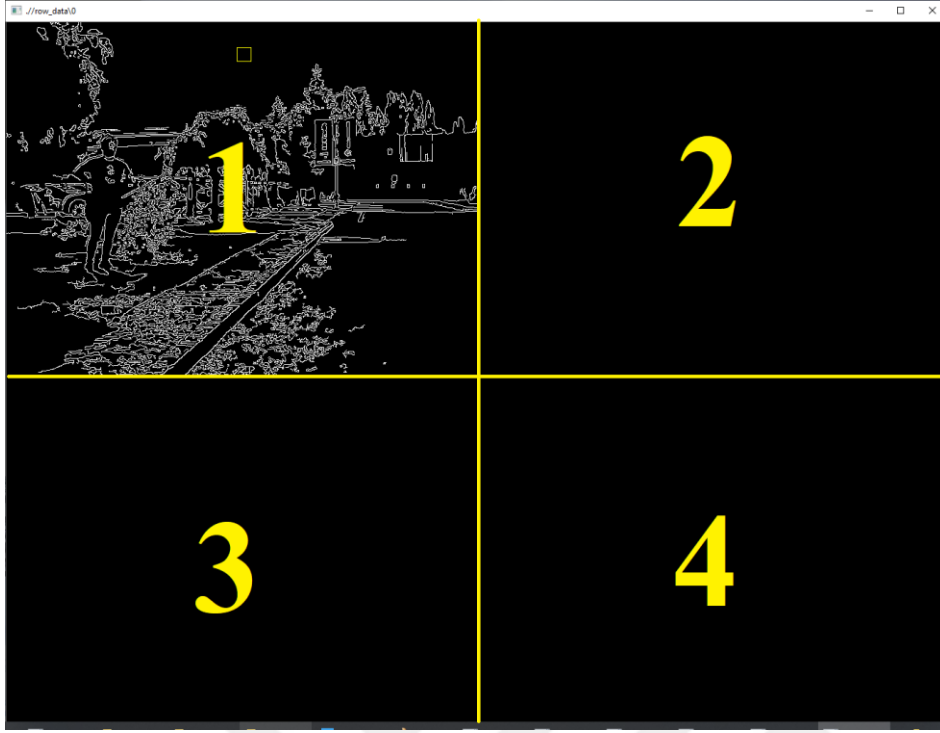
- Canny (1986) kenar filtresi kullanarak, obje kenarlarının referans alınacak olan görüntüye aktarılması (Bu işlem, filtre parametrelerini değiştirerek ve referans alınacak olan görüntüye kopyalama yapılarak ilgili objeyi en doğru yansıtacak şekilde adımlar halinde yapılır.)
- Elde edilen referans görüntüsünü morfolojik aşındırma, yayma ve sınır çıkarma (Serra, 1982: 184, 1988: 13; Dougherty, 1992: 10) işlemlerinden geçirerek, ilgili obje için kapalı alan elde edilmesi
- Elde edilmiş bu kapalı alanların objenin derinlik bilgisine göre doldurulması (Örneğin, yol, duvar gibi derinliği değişen objelere değişken derinlik verilirken; insan, ağaç, pano gibi derinliği çok değişmeyen objelere sabit derinlik verilmiştir.)

Resim 3.12’de bu işlemin sonucu görsel olarak gösterilmiştir.



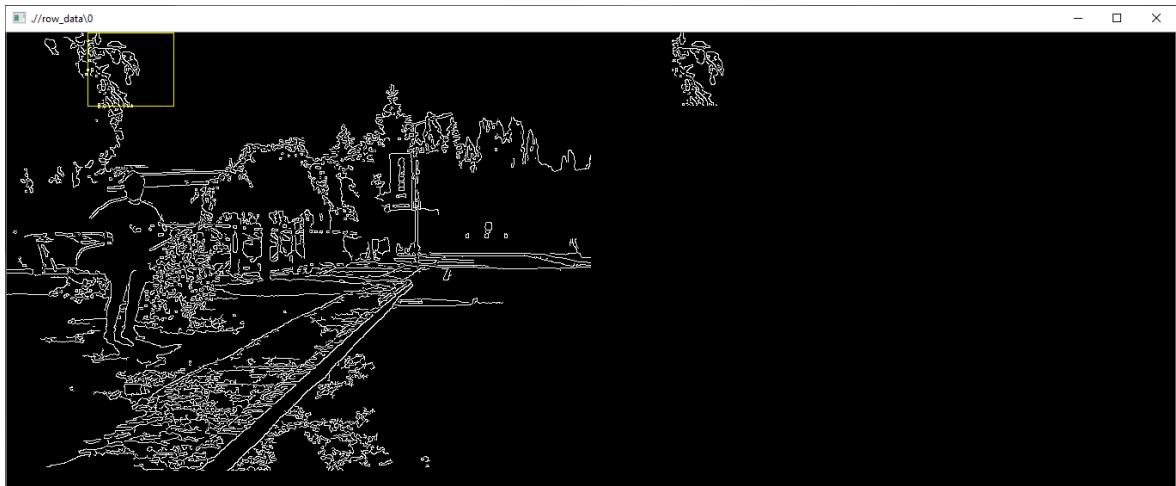
Resim 3.12. Doldurma işlemi; Sol: RGB görüntüsü, Orta: Örnekleri arttırılmış görüntü, Sağ: Örnekleri doldurulmuş görüntü

Python tabanlı yazılan bir uygulama ile bu işlemler sırayla gerçekleştirilmiştir. Şekil 3.7’de gösterildiği gibi uygulama arayüzü dört alandan oluşur. İlk alan, ilgili resmin Canny (1986) kenar filtresi ile elde edilmiş görüntüyü gösterir. İkinci alan, ilk görüntüden kopyalanacak olan görüntü parçalarını gösterir. Üçüncü alan, ikinci alandan elde edilecek kapalı alanların rastgele renklendirilmesini gösterir. Son olarak dördüncü alan ise, üçüncü alandan seçilen kapalı alanın aktarıldığı alandır.



Şekil 3.7. Örnek doldurma programı arayüzü

İlk alanda gözüken sarı ile gösterilmiş kutucuk, fare ile kontrol edilerek kutucuğun içinde kalan alan parçalarını ikinci alana aktarmaya yarar. Burada farenin kaydırma tuşuyla kutucuğun boyutu ayarlanır. Bu alanda farenin sol tuşu, kutucuğun içeriğini olduğu gibi ikinci alana aktarır. Farenin sağ tuşu ise kutucuğun içeriğinin tersini ikinci alana aktarmaya yarar. İlk alandaki görüntü elde edilirken uygulanan Canny (1986) kenar filtresinin parametrelerini dinamik bir şekilde klavye üzerinden değiştirme imkânı vardır. İlk alandan ikinci alana kopyalama işlemi bir örnek üzerinden Şekil 3.8’de gösterilmiştir.



Şekil 3.8. İlk alandan ikinci alana kopyalama işlemi (ilk)

Bu örnekte gökyüzü kapalı alan olarak elde edilmeye çalışılmıştır. Şekil 3.9'te ilk alandan ikinci alana kopyalama işleminin, gökyüzü için olan kısmı tamamlanmıştır.



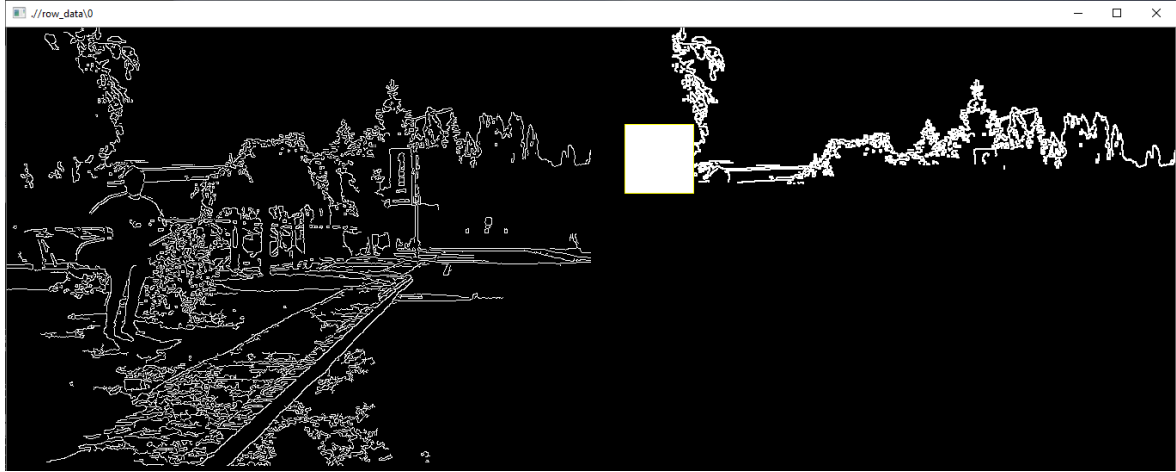
Şekil 3.9. İlk alandan ikinci alana kopyalama işlemi (son)

İkinci alanda, kapalı alana gidilmeye çalışıldığı için klavye üzerinden çeşitli morfolojik işlemler uygulama yeteneği eklenmiştir. Şekil 3.10'da ikinci alanda mevcut olan görüntüye morfolojik genişleme işlemi uygulanmış, bu sayede arada kalan boşluklar giderilmeye çalışılmıştır. Fakat sarı kutucuk ile gösterilen yerde ufak bir açıklık gözükmemektedir.



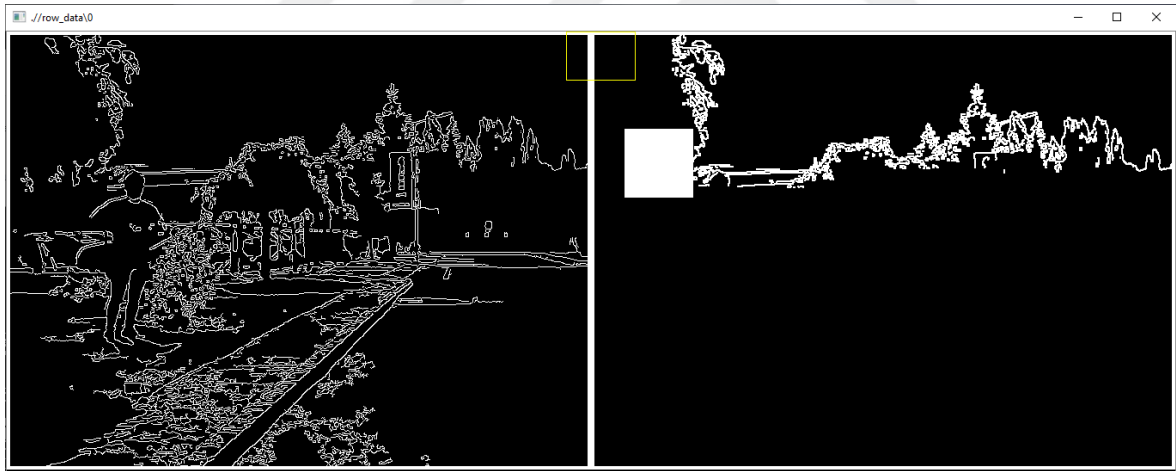
Şekil 3.10. İkinci alanda morfolojik genişleme işlemi

İkinci alanda, ilk alanda olduğu gibi sarı kutucuk fare vasıtasıyla kontrol edilmekte olup, çeşitli fonksiyonları vardır. Farenin kaydırma tuşları ile bu kutucuğun boyutları artırılıp azaltılabilir. Farenin sol tuşu, kutucuğun içeri beyaz; sağ tuşu ise kutucuğun içeri siyah doldurur (silme işlemi). Şekil 3.11'de sarı kutucuğun içi doldurularak, aralık kapatılmıştır.



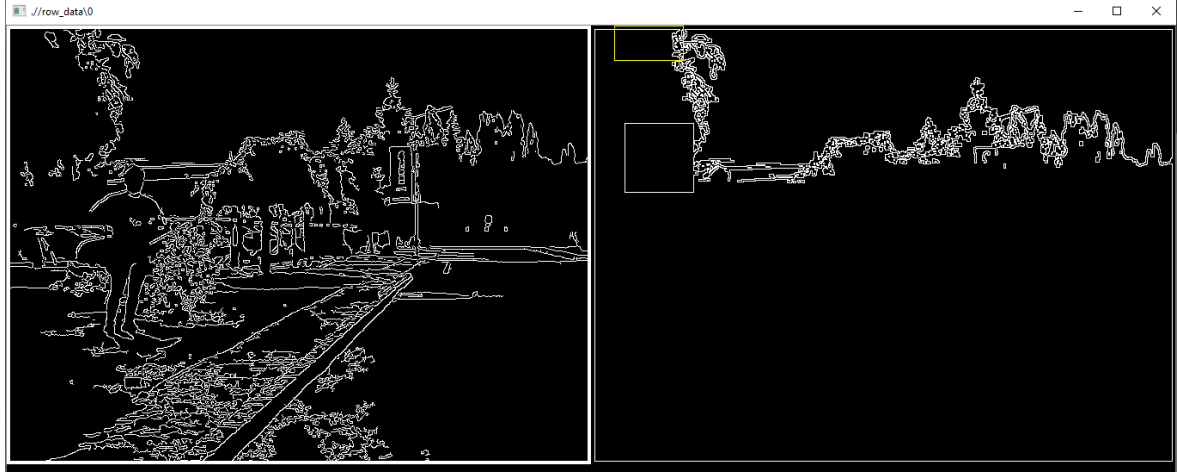
Şekil 3.11. Aralık kapatma işlemi

Klavye üzerinden bir tuş vasıtasıyla birinci ve ikinci alana beyaz kenarlık eklenir. Kenarlık eklenmesinin amacı da kenarların açıklığından kaynaklı sızıntıları engellemek içindir. Bu uygulamaya örnek Şekil 3.12’de gösterilmiştir.



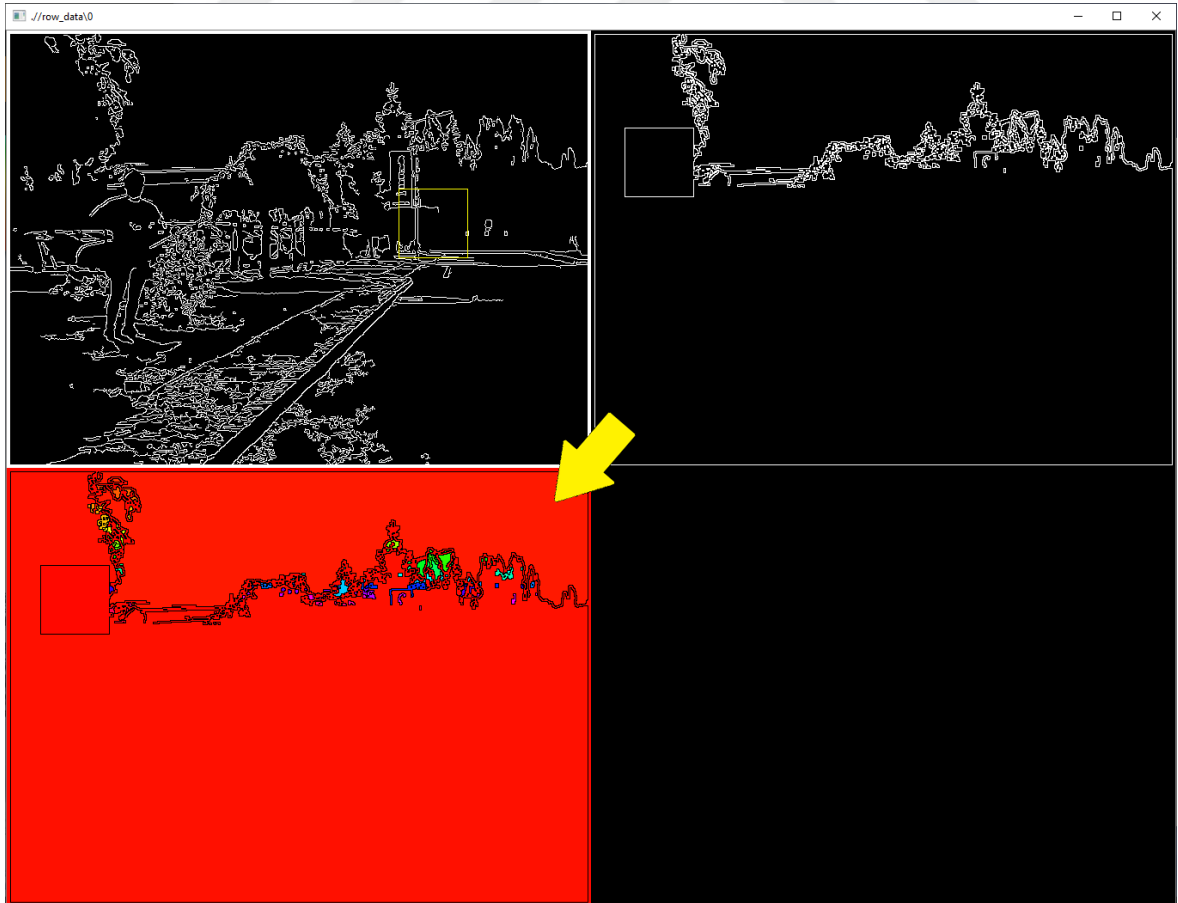
Şekil 3.12. Kenarlık ekleme işlemi

Klavye üzerinden daha önce belirlenmiş bir tuşa basarak ikinci alanda morfolojik sınır çıkarma işlemi uygulanır. Bu işlemin amacı kapalı alanı elde etmeye yönelik pekiştirme yapılmasıdır. Bu sayede göz ile görülemeyen muhtemel açık alanlar (sızıntılar) engellenir. Ayrıca ağaç, dal vb. gibi daha yoğun objelerde arada kalmış kapalı alanların elde edilmesine de faydası olur. Sınır çıkarma işlemi yapılmadığı takdirde birçok bölge tamamlanamaz ve daha büyük bölgeler ile kapalı alan oluşturma ihtimali artar. Bu işlem, Şekil 3.13’te örnek olarak gösterilmiştir.



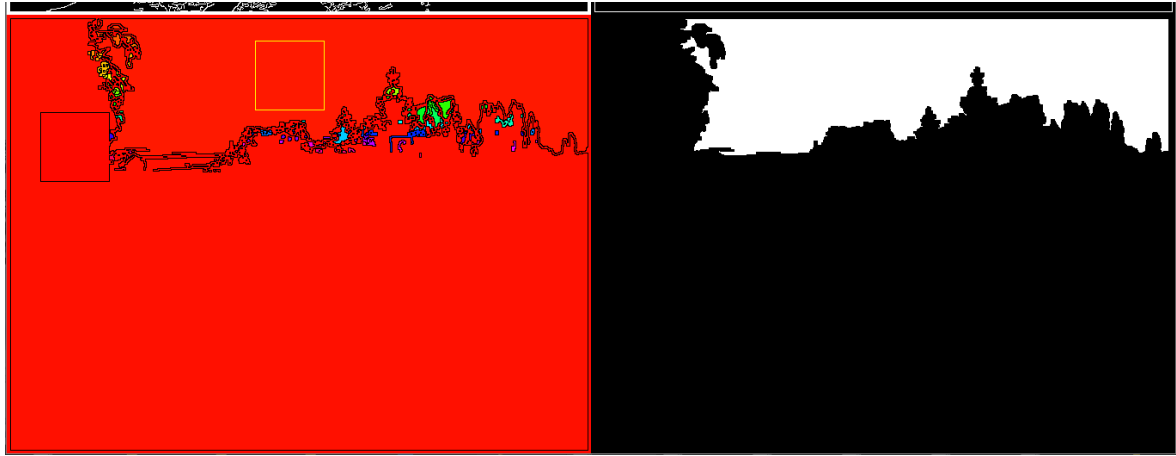
Şekil 3.13. İkinci alanda morfolojik sınır çıkarma işlemi

Gerekli kapalı alanın elde edildiğine kanaat getirildikten sonra, klavye üzerinden bir tuşla üçüncü alana görüntü aktarılır. Görüntü aktarma işlemi, ikinci alanda elde edilen kapalı bölgeler, hazır bir fonksiyon vasıtasıyla farklı ve rastgele olacak şekilde renklendirilerek üçüncü alanda gösterilir. Bu uygulamaya örnek Şekil 3.14’te gösterilmiştir.



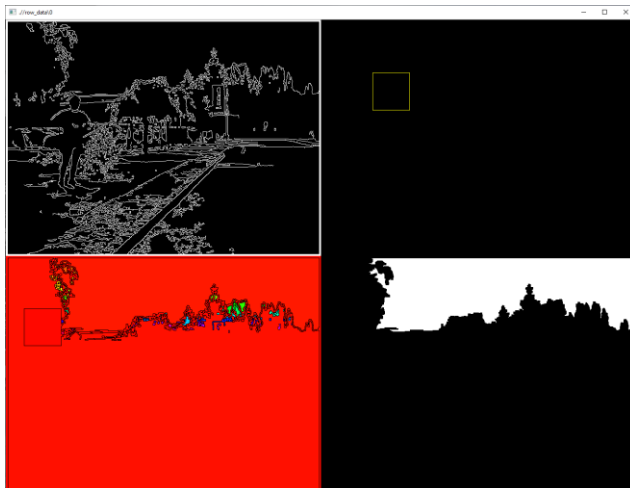
Şekil 3.14. İkinci alandan üçüncü alana görüntü aktarılması

Üçüncü alanda elde edilen görüntü üzerinde, dördüncü alana aktarılması istenilen kapalı alanın (örnekte gökyüzü), herhangi bir noktasına farenin sol tuşu ile tıklanmasıyla, o alanın aynı renk tonuna sahip tüm noktalar dördüncü alana aktarılır. Benzer veya birbirine yakın renkleri ayırt etmek için ise farenin sağ tuşu ile o pikselin renk tonunun RGB değeri ekrana bastırılır. Eğer renk tonları aynı ise ikinci alana geri dönülüp gözden kaçan açık alanların kapatılması gereklidir. Şekil 3.14'te gökyüzü ile diğer büyük alan benzer kırmızı tona sahiptir. Fakat RGB tonları farklıdır. Bu sebeple ikinci alana geri dönülmesine ihtiyaç yoktur. Gökyüzüne ait herhangi bir noktaya tıklanarak dördüncü alana görüntü aktarımı sağlanabilir. Bu uygulamaya örnek Şekil 3.15'te gösterilmiştir.



Şekil 3.15. Dördüncü alana görüntü aktarılması

Gökyüzü segmente edildikten sonra, diğer objeleri segmente etmek için ikinci alan klavyedeki bir tuş vasıtasıyla sıfırlanmıştır. Bu işlem Şekil 3.16'da gösterilmiştir.



Şekil 3.16. İkinci alan görüntü sıfırlama işlemi

Buraya kadar uygulanan adımlar, görüntüdeki diğer objeleri segmente etmek için tekrar edilir. Şekil 3.17’de görüntüdeki kişinin segmente edilmesinin örneği gösterilmiştir.



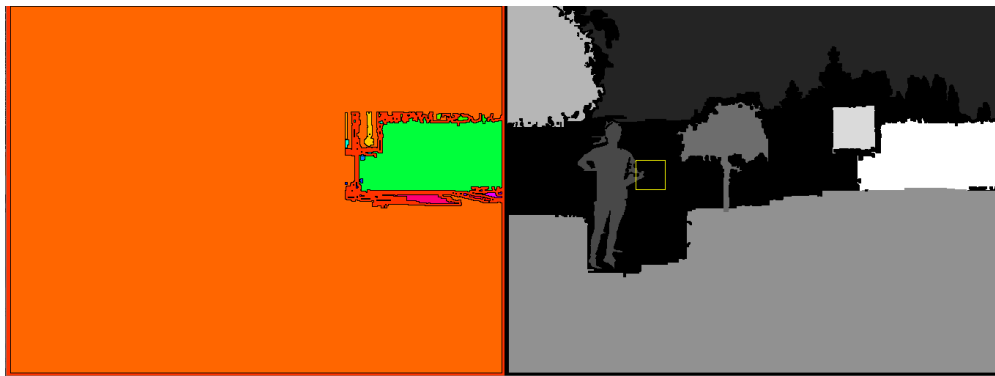
Şekil 3.17. Kişi objesinin segmente edilme işlemi

Görüntüdeki kişi objesi üçüncü alanda segmente edildikten sonra, klavyedeki bir tuş vasıtasıyla yeni obje olduğunu göstermek için yeni bir numara atanır ve bu numara “derinlik seviyesi” olarak ifade edilir. Örnek olarak gökyüzünün derinlik seviyesi 0’dır. Kişi objesinin derinlik bilgisiyle gökyüzünün derinlik bilgisi farklı olacağından dolayı, kişi objesine yeni değer atanır ve bu değer de 1’dir. Bu şekilde her obje için yeni derinlik seviyesi verilir. Yeni değer atandıktan sonra üçüncü alanda fare vasıtasıyla seçilen kişi objesi dördüncü alana aktarılır. Dördüncü alanda toplam kaç derinlik seviyesi varsa, 0-255 değeri o kadar kısımda eşit olarak gruplandırılır. Örnek olarak, sadece gökyüzü varken gökyüzü 255 değerini almıştır. Fakat ikinci bir obje gelince, değer ikiye bölüneceğinden gökyüzü 128 değerini, kişi 255 değerini alır. Buradaki maksat dördüncü alanda segmente edilmiş görüntüde objelerin derinlik seviyelerine göre gözle rahat bir şekilde görülebilmesidir. Bu uygulamaya örnek Şekil 3.18’de gösterilmiştir.



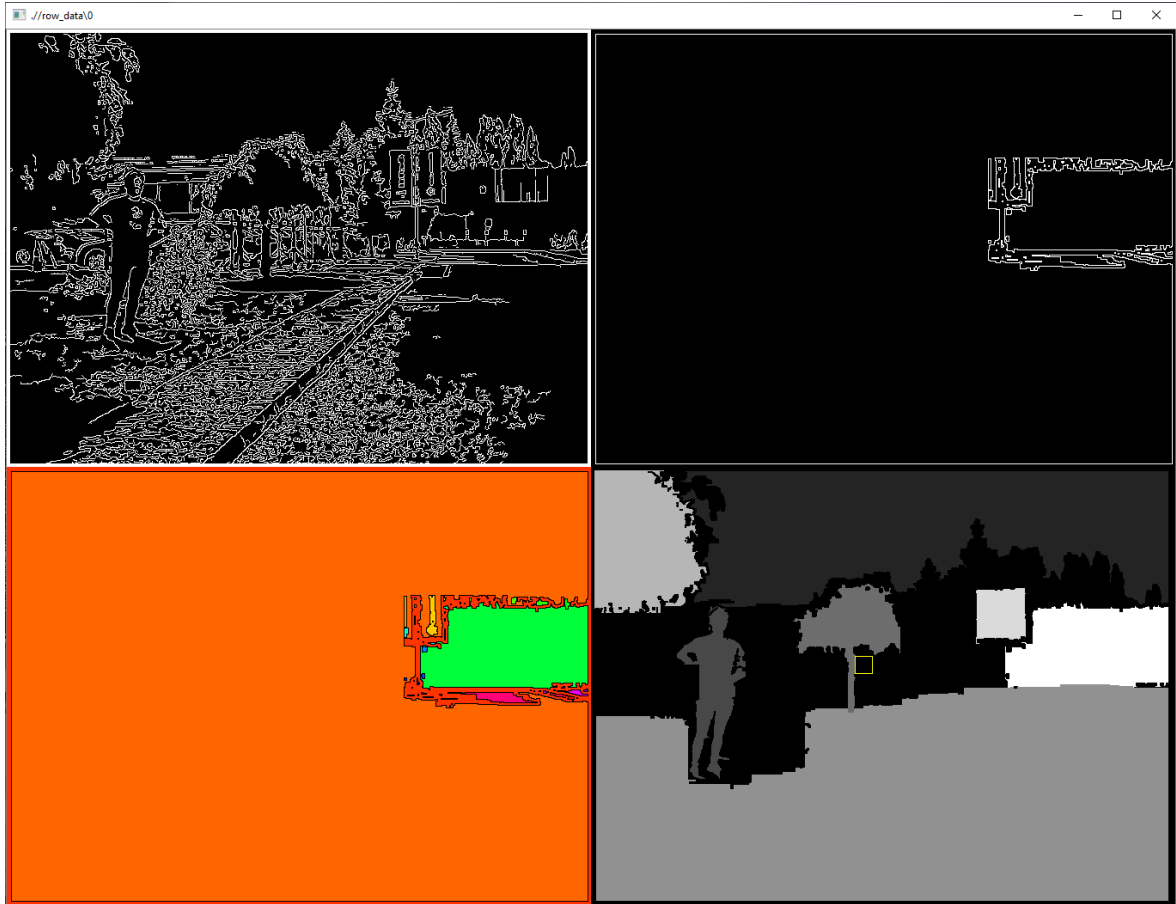
Şekil 3.18. Dördüncü alana yeni bir objenin eklenmesi

Bu şekilde yapılarak derinlik bilgisi verilmek istenen tüm objeler ayrı ayrı segmente edilmiş ve Şekil 3.19’da gösterilmiştir.



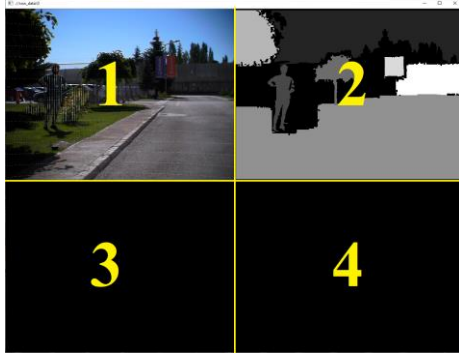
Şekil 3.19. Dördüncü alanda derinlik bilgisi verilecek tüm objelerin segmente edilmiş son durumu

Segmente edilen görüntülerde istenmeyen veya hatalı olan noktalar bulunuyorsa, dördüncü alanda bu durum giderilebilir. Farenin kaydırma tuşuyla sarı ile gösterilen kutucuğun boyutları arttırılıp azaltılabilir. Farenin sol tuşuyla üzerine bulunulan piksele ait derinlik bilgisi seçilir, bu sayede geriye dönük üçüncü alanda elde edilecek ve aynı derinlik seviyesinde olduğu düşünülen bölgeler, seçilen derinlik seviyesine tekrar eklenebilir. Farenin sağ tuşuyla da dördüncü alanda aktif derinlik seviyesine bağlı bölgelerin sarı kutucuğun içinde kalan kısmı silinebilir. Bu işleme örnek Şekil 3.20’de verilmiştir.



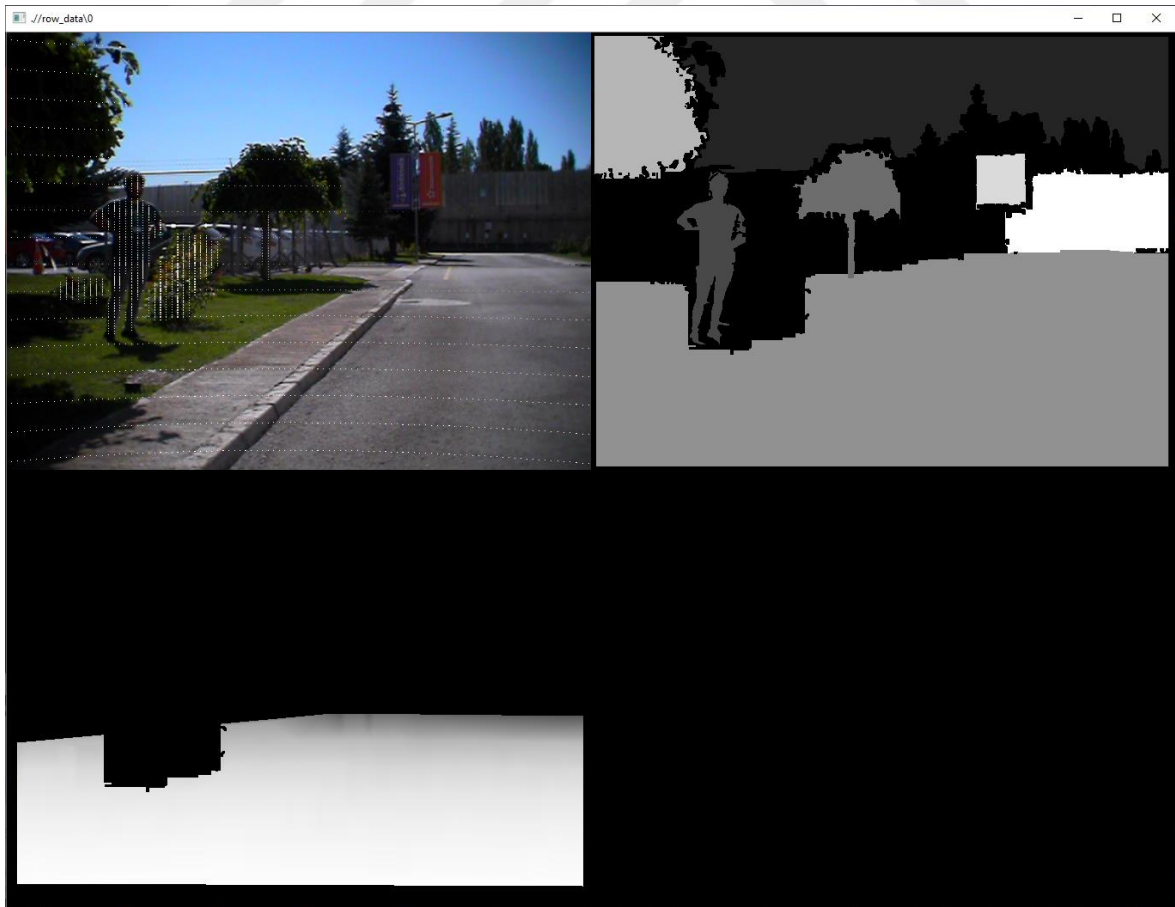
Şekil 3.20. Dördüncü bölgede istenmeyen kısımların silinmesi

Bu adımlar uygulanarak elde edilen segmentasyon görüntüsü, daha sonra derinlik vermek için kullanılır. Bu işlem Şekil 3.21’de gösterildiği gibi yeni pencere üzerinden gösterilir. Burada da pencere dört alana bölünür. İlk alanda mevcut derinlik bilgisi, referans görüntüsüne bindirilir. İkinci alanda, bir önceki adımda elde edilen segmentasyon görüntüsü gösterilir. Üçüncü alanda yeni oluşturulacak derinlik haritasının son hali gösterilir. Dördüncü alanın herhangi bir fonksiyonu yoktur.



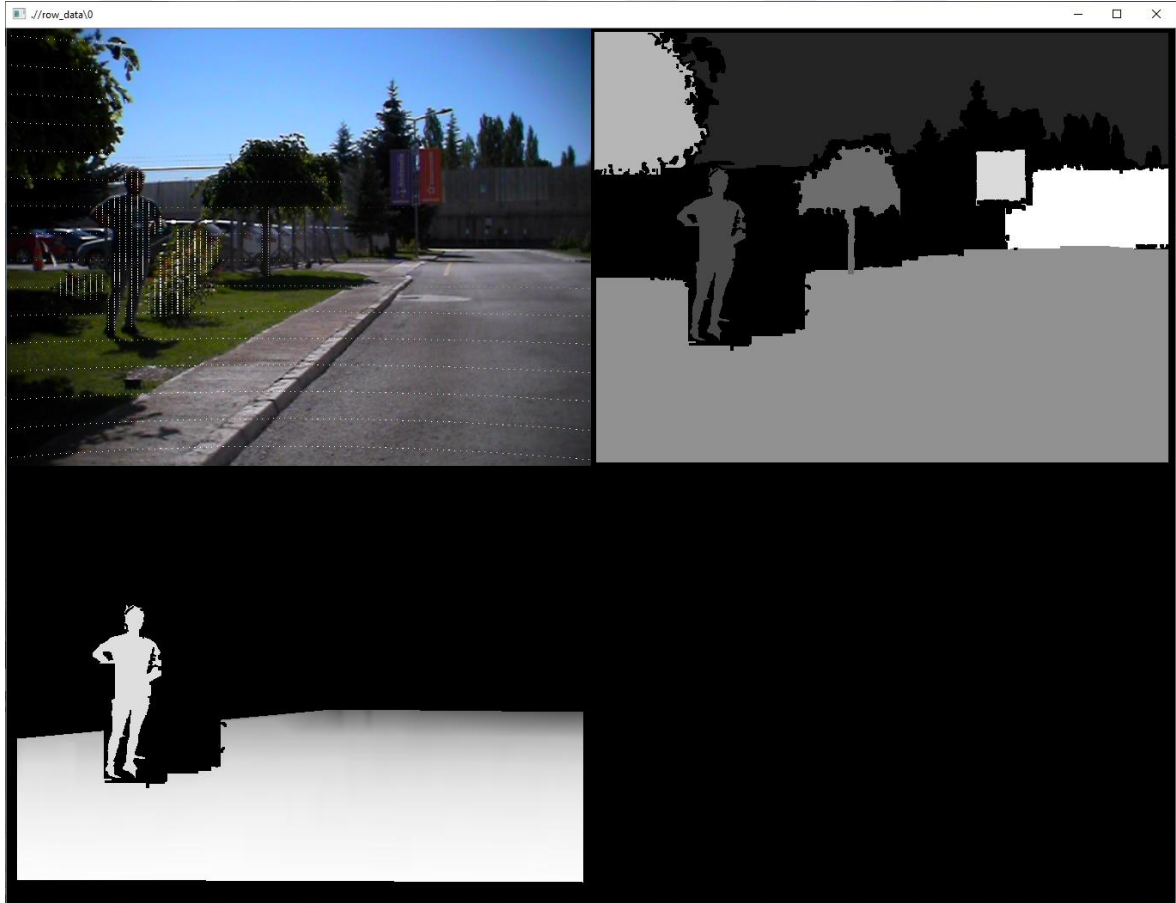
Şekil 3.21. Segmente edilmiş görüntüyü kullanarak derinlik bilgisi atama

Yol, duvar gibi objeler için derinlik sabit olmadığından dolayı bu kısımların derinlik değerleri, çift doğrusal ara değeri bulma yöntemiyle doldurulur. Örnek olarak yolun olduğu kısmı doldurmak için, ikinci alanda farenin sağ tuşuyla yolun herhangi bir yerine tıklanması yeterlidir. Uygulama, derinlik haritasında ara değerleri bularak, ikinci alanda yolun olmadığı kısımları filtreleyip üçüncü alana ekler. Bu uygulamaya örnek Şekil 3.22’de gösterilmiştir.



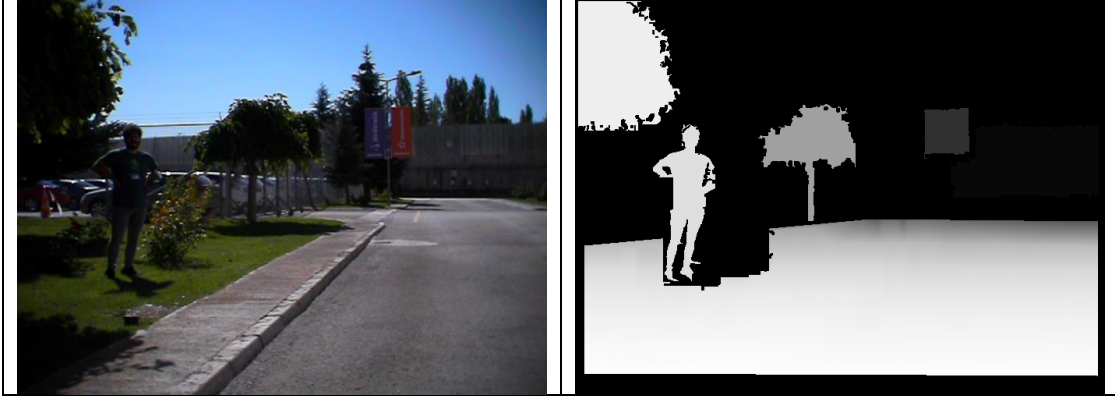
Şekil 3.22. Yol objesine derinlik değeri atama

Sabit derinlikli veya derinlik bilgisinin çok farklı olmadığı objelerin (insan, ağaç, tabela vb.) üçüncü alana aktarılmasında öncelikli olarak ilk alanda derinlik miktarını tespit etmek gerekir. Bunun için farenin sol tuşuyla hangi objenin derinliği bulunmak isteniyorsa o objeye tıklanır. Şekil 3.23'ün birinci alanında bulunan insan objesinin üzerindeki herhangi bir yere tıklanmıştır. Mevcut derinlik haritasında bulunan örnekler ayırık örnekler olduğu için tıklanan piksele en yakın geçerli derinlik bilgisi bulunur ve hafızaya alınır. Daha sonra ikinci alanda yine ilgili objeye farenin sol tuşuyla tıklanarak, o objenin kapsadığı alan üçüncü alanda filtre gibi kullanılır. Böylelikle hafızaya alınan derinlik değeri, üçüncü alana elde edilen filtre kullanılarak atanır. Bu uygulamaya örnek olarak Şekil 3.23'te gösterilmiştir.



Şekil 3.23. İnsan objesine derinlik değeri atama

Bu şekilde yapılarak segmente edilen her obje için ilgili uygun metot uygulanarak Resim 3.13'te gösterildiği gibi derinlik haritasının son hali elde edilir.



Resim 3.13. Derinlik haritasının son hali





4. DENEYSEL ÇALIŞMALAR

Ön çalışma tamamlandıktan sonra her iki veri kümesi için ağ eğitimine başlanmıştır. Öncelikle, oluşturulan veri kümesi sırasıyla FADNet ve PSMNet çalışmalarında sıfırdan eğitilmiş, daha sonra CATS veri kümesi yine sırasıyla FADNet ve PSMNet çalışmalarında sıfırdan eğitilmiştir. Çalışmada 16 GB hafızaya sahip Nvidia Quadro RTX 5000 grafik kartı kullanılmıştır. Her iki ağda Pytorch yapısı kullanılarak oluşturulmuştur. Her iki ağ için Adam optimizasyonu (Kingma ve Ba, 2014) kullanılmıştır. Bu optimizasyonda kullanılan katsayılar $\beta_1 = 0,9$ ve $\beta_2 = 0,999$ olarak belirlenmiştir. Eğitim süresince veriler 256×512 boyutlarında rastgele kırılarak verilmiştir. Bu sayede görüntü çeşitliliği sağlanır ve örnek sayısı artırılır. CATS veri kümesiyle ağlar eğitilmeden önce, veri kümesi eğitim ve test amacıyla önceden ayrılmadığı için bu işlem bu adımda yapılmıştır. Veri kümesi içerisindeki her sahneden birer örnek test seti için saklanmış olup, eğitimde kullanılmamıştır. 70'i eğitim, 10'u test olmak üzere toplam 80 görüntü ile ağ eğitimlerine başlanılmıştır.

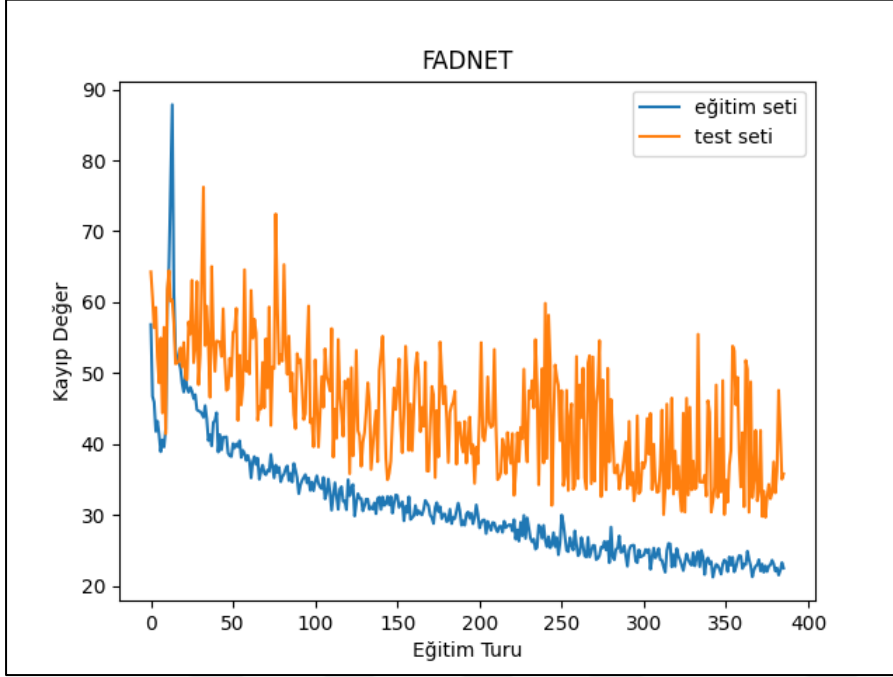
Eğitim sürecinde kayıp grafikleri incelendiğinde eğitim süreçlerinin normal seyrinde olduğu gözlenmektedir. Yapılan dört farklı eğitim için de herhangi bir aşırı uyum veya yetersiz uyum durumu söz konusu değildir. Eğitim sonunda elde edilen model, diğer bir deyişle öğrenilen katman katsayıları kaydedilir ve test için kullanılır. Test işleminde daha önce de bahsedildiği gibi test verisi kullanılır. Test verisi eğitim için kullanılmamıştır. Dolayısıyla, ağ daha önce eğitim sırasında hiç görmediği bir görüntüden derinlik haritasını tahmin etmeye çalışır. Sonuçlar sayısal sonuçlar ve görsel sonuçlar olmak üzere iki başlık altında incelenmiştir. Sayısal sonuçlar için literatürde kabul gören KITTI piksel hata hesaplaması ve McNemar test sınaması kullanılmıştır. Bu metrik, veri kümelerine göre hem FADNet ağı hem de PSMNet ağı için karşılaştırmalı olarak verilmiştir.

4.1. Ağ Eğitimleri

4.1.1. Oluşturulan veri kümesi ile eğitim

Oluşturulan bu veri kümesi ile FADNet ağının eğitim işlemi 386 eğitim turu sürmüştür. İlk 250 eğitim turu boyunca öğrenme katsayısı 0,001, sonraki 136 eğitim turu boyunca

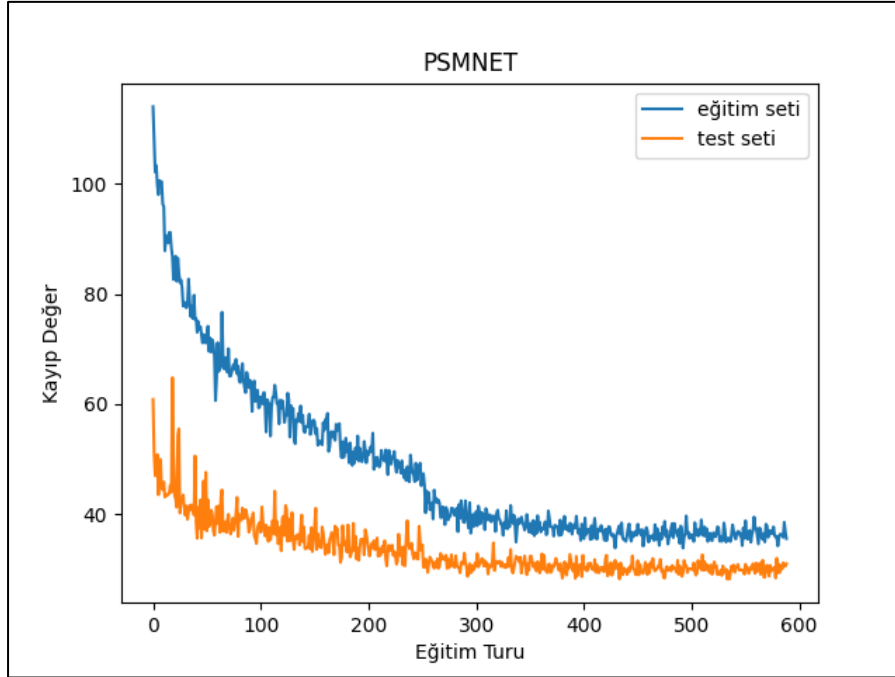
öğrenme katsayısı 0,0001 olarak belirlenmiştir. Öğrenme için parti boyutu, grafik kartının hafıza limitlemesinden ötürü 4 olarak verilebilmiştir. Öğrenme süresince elde edilen kayıp değer grafiği Şekil 4.1’de verilmiştir.



Şekil 4.1. Oluşturulan veri kümesinin FADNet ağındaki kayıp değer grafiği

Şekil 4.1 incelendiğinde eğitim esnasındaki kayıp değer giderek azalmaktadır. Fakat aynı durum test esnasında söylenemez. Bu grafik bize ağın tam olarak öğrenemediğini gösterir. Eğitim sonunda kayıp değer 20'nin üzerinde olması, ağın tam olarak öğrenemediğinin bir başka göstergesidir. Bunun birkaç sebebi olabilir. İlk olarak çapraz modalite kullanıldığı için, özellik çıkartma adımında farklı özellikler çıkartılmış olabilir. Bu da korelasyonun tam olarak sağlanamamasına yol açar. Bir başka sebep, veri kümesindeki tutarsızlıklardan kaynaklı olabilir. 30 görüntü için yapılan örnek doldurma işlemi veri kümesinin hepsine uygulanabilseydi daha iyi sonuçları gözlemlemek kaçınılmaz olabilirdi.

PSMNet ağına eğitim işlemi ise 589 eğitim turu sürmüştür. İlk 250 eğitim turu boyunca öğrenme katsayısı 0,001, sonraki 150 boyunca 0,0001 ve son 189 eğitim turu için öğrenme katsayısı 0,00001 olarak belirlenmiştir. Öğrenme için parti boyutu, grafik kartının hafıza limitlemesinden ötürü 2 olarak verilebilmiştir. Öğrenme süresince elde edilen kayıp değer grafiği Şekil 4.2’de verilmiştir.



Şekil 4.2. Oluşturulan veri kümesinin PSMNet ağındaki kayıp değer grafiği

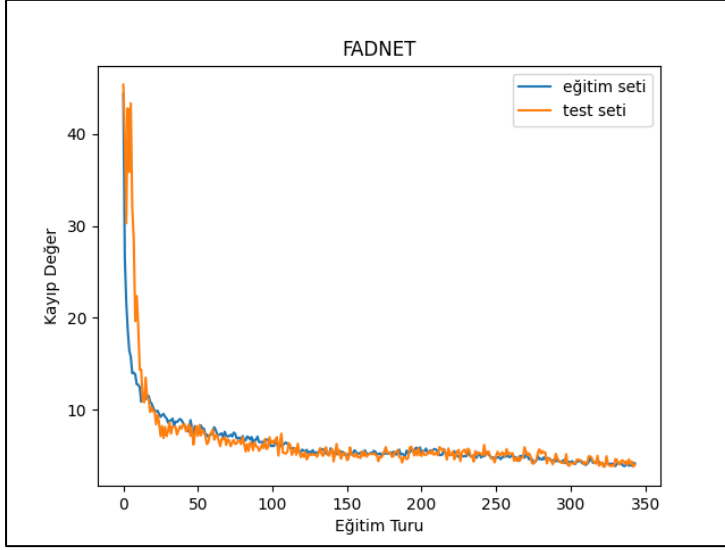
Şekil 4.2 incelendiğinde eğitim ve test esnasındaki kayıp değer giderek azalmaktadır. Çizelge 4.1'e bakıldığında kayıp değer yaklaşık 35'in üzerinde olduğu görülmektedir. Bu grafik bize ağın tam olarak öğrenemediğini gösterir. FADNet'te gözlemlenen test setinin kayıp değer dalgalanması PSMNet'te görülmemiştir. FADNet'te değinilen problemlerin benzer durumu bu ağı için de geçerlidir.

Çizelge 4.1. Oluşturulan veri kümesinin ortalama kayıp değerleri

	FADNet	PSMNet
Eğitim kümesi	22,52	35,57
Test kümesi	35,81	31,01

4.1.2. CATS veri kümesi ile eğitim

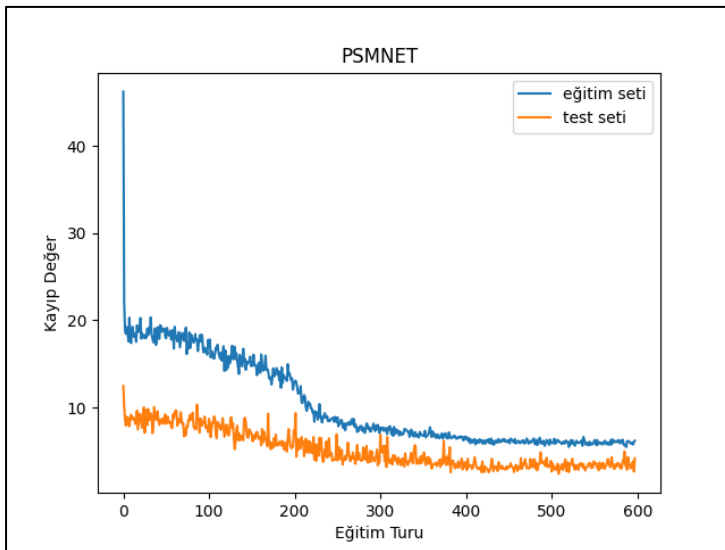
CATS veri kümesi ile FADNet ağına eğitim işlemi 350 eğitim turu sürmüştür. İlk 100 eğitim turu boyunca öğrenme katsayısı 0,001, sonraki 250 eğitim turu boyunca öğrenme katsayısı 0,0001 olarak belirlenmiştir. Öğrenme için parti boyutu, grafik kartının hafıza limitlemesinden ötürü yine 4 olarak verilebilmiştir. Öğrenme süresince elde edilen kayıp değer grafiği Şekil 4.3'te verilmiştir.



Şekil 4.3. CATS veri kümesinin FADNet ağındaki kayıp değer grafiği

Şekil 4.3 incelendiğinde eğitim süreci ve test sürecinin kayıp değeri birbirlerine uygun olarak seyretmektedir. Bu durum FADNet'in CATS veri kümesiyle öğrendiğini ortaya koyar.

PSMNet ağına eğitim işlemi 600 eğitim turu sürmüştür. İlk 250 eğitim turu boyunca öğrenme katsayısı 0,001, sonraki 350 eğitim turu boyunca öğrenme katsayısı 0,0001 olarak belirlenmiştir. Öğrenme için parti boyutu, grafik kartının hafıza limitlemesinden ötürü bir önceki bölümde olduğu gibi yine 2 olarak verilebilmiştir. Öğrenme süresince elde edilen kayıp değer grafiği Şekil 4.4'te verilmiştir.



Şekil 4.4. CATS veri kümesinin PSMNet ağındaki kayıp değer grafiği

Şekil 4.4 incelendiğinde eğitim süreci ve test sürecinin kayıp değeri, FADNet'te olduğu gibi birbirlerine uygun olarak seyretmektedir. Bu durum PSMNet'in CATS veri kümesiyle öğrendiğini ortaya koyar. Ancak Çizelge 4.2'den de görüleceği üzere kayıp değerler FADNet ile kıyaslandığında, değer daha yüksektir.

Çizelge 4.2. CATS veri kümesinin ortalama kayıp değerleri

	FADNet	PSMNet
Eğitim kümesi	4,16	6,18
Test kümesi	3,99	4,17

4.2. Ağ Eğitim Sayısal Sonuçları

4.2.1. Oluşturulan veri kümesi sayısal sonuçları

Oluşturulan veri kümesindeki test görüntüleriyle FADNet ve PSMNet ağları, eğitim modeli yüklenerek test edilmiştir. Çizelge 4.3'te KITTI piksel hata hesaplaması, Çizelge 4.4'te de kontenjans tablosu gösterilmiştir.

Çizelge 4.3. Oluşturulan veri kümesindeki eğitim sonuçlarının KITTI piksel hata hesaplamaları

	>3px (%)		>4px (%)		>5px (%)	
	FADNet	PSMNet	FADNet	PSMNet	FADNet	PSMNet
Eğitim Seti	90,4	99,1	88,1	98,8	85,9	98,5
Test Seti	91,8	99	89,8	98,6	87,8	98,3

Çizelge 4.3'e bakılarak oluşturulan veri kümesiyle eğitilen her iki ağında yüksek miktarda hatalı sonuç verdiği görülebilir. FADNet ağının PSMNet ağına göre ortalama %10 oranında daha iyi sonuç verdiği görülmektedir. Piksel toleransı arttırıldıkça FADNet ağından elde edilen sonuçların hata oranı azalırken, PSMNet ağından elde edilen sonuçların hata oranının çok fazla değişmediği görülmektedir. Bu bilgiye dayanarak, FADNet ağından elde edilen çıktıların görselliği ile verilen eğitim görüntüsünün bir ilişkisinin olduğu, fakat aynı durumun PSMNet için söylenemeyeceği yorumu çıkarılabilir.

Çizelge 4.4. Oluşturulan veri kümesi test çıktıları üzerindeki farklı toleranslarda 2x2'lik kontenjans tablosu

		Tolerans (<3px)		Tolerans (<4px)		Tolerans (<5px)	
		FADNet					
		Doğru Tahmin	Yanlış Tahmin	Doğru Tahmin	Yanlış Tahmin	Doğru Tahmin	Yanlış Tahmin
PSMNet	Doğru Tahmin	24	302	57	407	147	547
	Yanlış Tahmin	6 398	29 337	8 049	27 548	9 536	25 831

Benzer bir şekilde Çizelge 4.4 incelendiğinde her iki ağın da ortak doğrularının çok az olduğu, tolerans arttıkça FADNet ağından elde edilen sonuçların doğruluk sayısının PSMNet'ten elde edilen sonuçlara göre daha fazla arttığı görülmektedir. Bu tablodan FADNet'in PSMNet'e göre daha iyi sonuç verdiği yorumu bir kez daha çıkarılabilir. Çizelge 4.4 kullanılarak McNemar test sınaması yapıldığında ise her iki modelin de birbirinden bağımsız olduğu sonucu ortaya çıkmaktadır.

4.2.2. CATS veri kümesi sayısal sonuçları

CATS veri kümesindeki test görüntüleriyle FADNet ve PSMNet ağları, eğitim modeli yüklenerek test edilmiştir. Çizelge 4.5'te KITTI piksel hata hesaplaması, Çizelge 4.6'da da kontenjans tablosu gösterilmiştir.

Çizelge 4.5. CATS veri kümesindeki eğitim sonuçlarının KITTI piksel hata hesaplamaları

	>3px (%)		>4px (%)		>5px (%)	
	FADNet	PSMNet	FADNet	PSMNet	FADNet	PSMNet
Eğitim Seti	41,1	42,9	24,7	29,9	15,9	22,2
Test Seti	44,3	45,6	29,1	30	20	22

Çizelge 4.5'e bakılarak CATS veri kümesiyle eğitilen her iki ağında birbirlerine yakın sonuç verdiği görülebilir. Her durumda FADNet ağının PSMNet ağına göre daha iyi sonuç verdiği görülmektedir. Piksel toleransı arttırıldıkça her iki ağdan elde edilen sonuçların hata oranının azaldığı gözlenmektedir. Bu bilgi göz önünde bulundurulduğunda, her iki ağdan elde edilen sonuçların görsel olarak iyi sonuçlar vereceği yorumu yapılabilir.

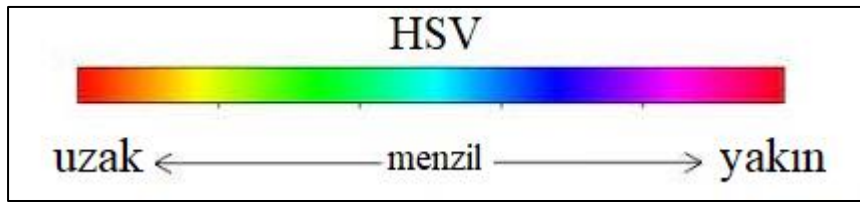
Çizelge 4.6. CATS veri kümesi test çıktıları üzerindeki farklı toleranslarda 2x2'lik kontenjans tablosu

		Tolerans (<3px)		Tolerans (<4px)		Tolerans (<5px)	
		FADNet					
		Doğru Tahmin	Yanlış Tahmin	Doğru Tahmin	Yanlış Tahmin	Doğru Tahmin	Yanlış Tahmin
PSMNet	Doğru Tahmin	63 274	20 761	96 579	12 484	113 932	7 655
	Yanlış Tahmin	33 604	28 359	22 682	14 251	15 870	8 541

Çizelge 4.6 incelendiğinde ise her iki ağında ortak doğruları oldukça fazla olduğu görülmektedir. Tolerans arttıkça bu doğruların sayısı da beklendiği gibi artmaktadır. Her durumda FADNet'in PSMNet'e göre daha iyi sonuçlar verdiği bu tablodan da görülmektedir. Bu tablo kullanılarak McNemar test sınaması yapıldığında, yine her iki modelin de bir önceki bölümde olduğu gibi, birbirinden bağımsız olduğu sonucu ortaya çıkmaktadır.

4.3. Ağ Eğitim Görsel Sonuçları

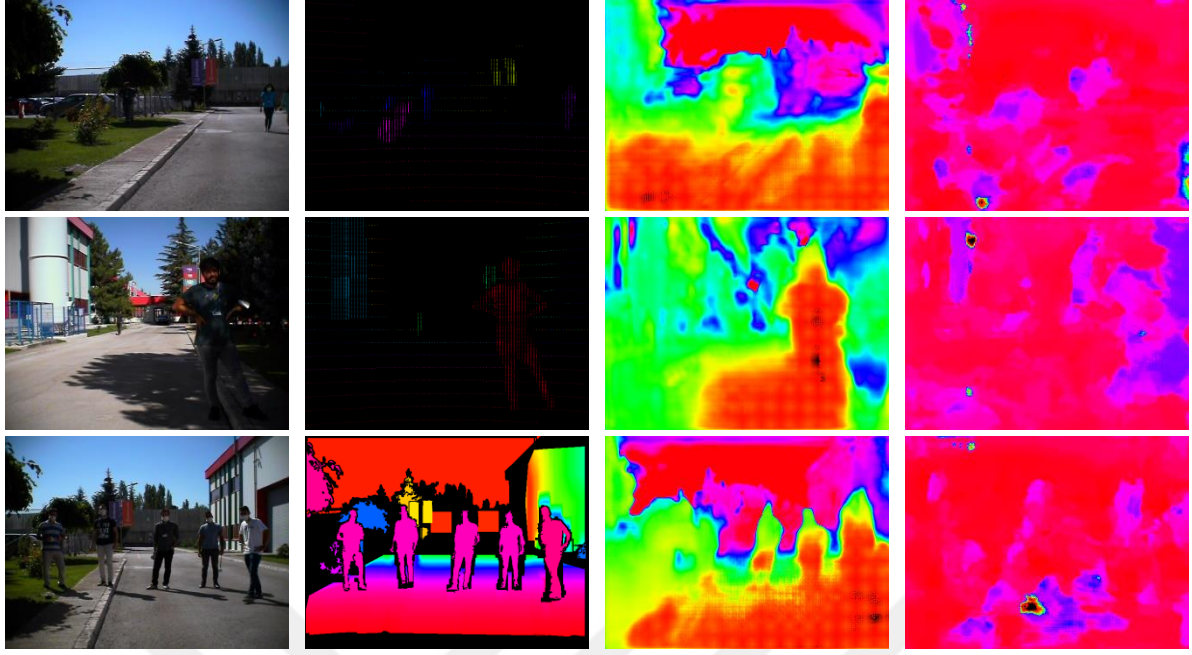
Elde edilen görsel sonuçlar için HSV renk dağılım skalası kullanılmıştır. Şekil 4.5'e bakıldığında turuncudan kırmızıya doğru giden renk skalası gözükmemektedir. Turuncu en uzak mesafeyi ifade ederken, kırmızı ise en yakın mesafeyi göstermektedir.



Şekil 4.5. Derinlik haritalarında kullanılan renk barı

4.3.1. Oluşturulan veri kümesi görsel sonuçları

Oluşturulan veri kümesindeki test görüntüleriyle FADNet ve PSMNet, eğitim modeli yüklenerek, test edilmiş ve Resim 4.1'de gösterildiği gibi sonuçlar elde edilmiştir.

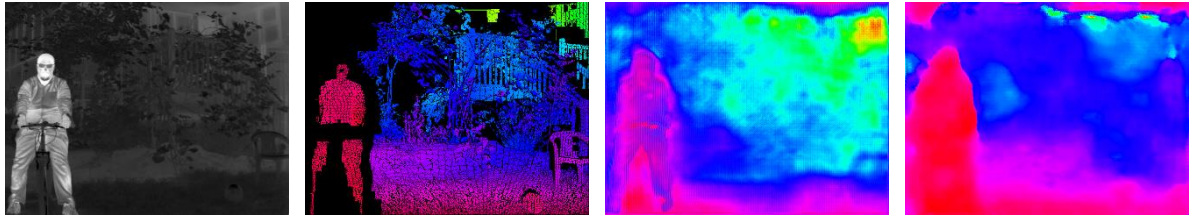


Resim 4.1. Soldan sağa; RGB görüntüsü, Gerçek referans değer görüntüsü, FADNet çıktısı, PSMNet çıktısı

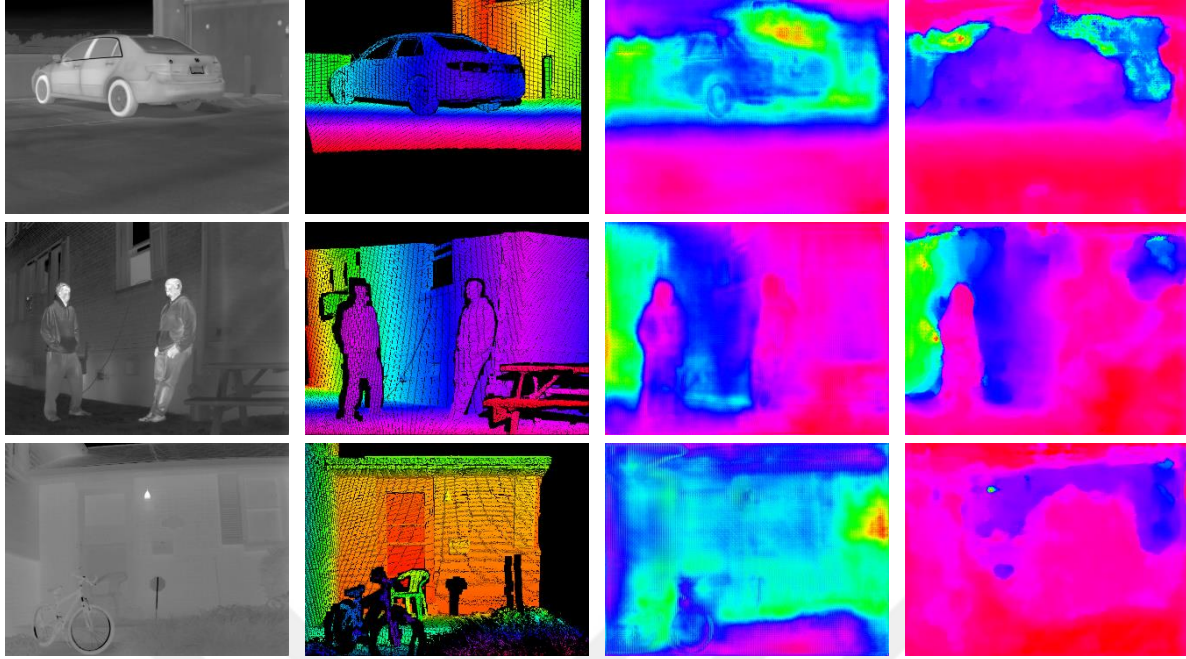
Resim 4.1'e bakıldığında FADNet ağının çıktı görüntüsünde kişi ve yol obje silüetlerinin seçilebilir olduğu, diğer obje silüetleri için herhangi belirgin bir sonuç oluşmadığı görülmektedir. PSMNet ağ çıktılarının tamamen rastgele değerler gibi olduğu gözlenmektedir. Bölüm 4.2.1'de sayısal sonuçlar üzerinden yapılan varsayımın doğru olduğu bu görseller ile tasdik edilmiştir. Diğer test görüntüleri ayrı ayrı olarak EK-1 ve EK-2'de görülebilir.

4.3.2. CATS veri kümesi görsel sonuçları

CATS veri kümesindeki test görüntüleriyle FADNet ve PSMNet, eğitim modeli yüklenerek, test edilmiş ve Resim 4.2'de gösterildiği gibi sonuçlar elde edilmiştir.



Resim 4.2. Soldan sağa; Sol termal görüntüsü, Gerçek referans değer görüntüsü, FADNet çıktısı, PSMNet çıktısı



Resim 4.2. (devam) Soldan sağa; Sol termal görüntüsü, Gerçek referans değer görüntüsü, FADNet çıktısı, PSMNet çıktısı

Resim 4.2'ye bakıldığında her iki ağ çıktısının çoğu zaman güzel ve anlaşılır sonuç verdiği gözlenmektedir. Sayısal değerlerde olduğu gibi görsel değerlerde de FADNet ağının çıktıları PSMNet ağının çıktılarına göre daha iyi sonuç verdiği gözlenmektedir. Örneğin bisikletin tekeri, araba lastikleri gibi detay objeler FADNet çıktısında daha belirgin, PSMNet çıktısında bu detaylar gözükmemektedir. Diğer test görüntüleri ayrı ayrı olarak EK-3 ve EK-4'te görülebilir.



5. SONUÇ VE ÖNERİLER

Bu çalışmada, günümüz teknoloji atılımları ışığında bilgisayarlı görü alanındaki güncelliğini ve tazeliğini koruyan stereo eşleme konusu üzerine, derin sinir ağı tabanlı mevcut çözümlerde termal görüntüler de kullanılarak derinlik haritasının çıkarılabileceği gösterilmiştir. Sonuçlar, görünür spektrumda alınan veri kümesi ile eğitilen stereo görüntü eşleme yöntemlerine göre daha başarısız gözükse de bu haliyle bile çok umut vericidir. Literatürde, Derin sinir ağlarında stereo eşleme uygulamaları için yayınlanan veri kümelerinin sadece görünür spektrumda olması, KITTI gibi başarısı kanıtlanmış veri kümeleri gibi stereo termal görüntüler içeren veri kümelerinin olmaması veya yetersiz olması bu çalışmayı sınırlandıran etmenler arasındadır. CATS veri kümesi, stereo termal görüntüler ve derinlik bilgileri içermesine rağmen, önceki bölümlerde de bahsedildiği üzere, sinir ağı eğitilmesi konusunda uyumsuzluğu bulunmaktadır. Bu çalışmada bu veri kümesi büyük ölçüde uyumlu hale getirilmiştir. Bu süreç, her görüntü üzerinde manuel olarak yapıldığı için, gözden kaçan noktalar, fark edilmeyen hususlar, hatalı örnekler bulunabilir. Bu gibi sebepler, ağ eğitimiyle doğrudan alakalı olduğu için, ağ çıktıları bu sebeplerden etkilenmiştir. Buna rağmen, 5 piksellik hata toleransında bile %80 başarı oranlarına erişilmiştir.

Bu tezde ayrıca çapraz modalite ile görüntü alınmış ve bu veri kümesi kullanılarak FADNet ve PSMNet ağı eğitilmiştir. FADNet ağı, PSMNet ağına göre daha iyi sonuç verse de sonuçlar çok iyi değildir. Bunun başlıca sebebi ise, kullanılan LiDAR cihazının örnek sayısının yetersiz olmasıdır. Ayrıca, kullanılan kameraların görüş alan açılarının birbirinden farklı olması, CATS veri kümesinde olduğu gibi oluşturulan bu veri kümesinde de gerçek referans değerlerinin manuel olarak arttırılmasından kaynaklı verilerin hatalı, yanlış, eksik olması gibi sebepler başarı oranının çok düşük olmasına neden olarak gösterilebilir.

Bu çalışmada, otonom sürüş, robotik, askeri vb. gibi uygulama alanlarında termal görüntü kullanımının artmasıyla birlikte, derinlik tahminin yapılmasında stereo termal görüntülerden de yararlanmanın temeli atılmıştır. Gelişen teknolojiyle birlikte, ilerleyen dönemlerde bu amaç doğrultusunda stereo termal görüntüler ile oluşturulmuş KITTI gibi verimli veri kümelerinin yaygınlaşacağı öngörülmektedir.

Mevcut şartlarda bu çalışmanın iyileştirilmesi için, veri kümesinin daha doğru bir şekilde eğitime verilmesi gereklidir. CATS veri kümesi üzerinden ilerlenecek olursa, bu veri kümesini daha uygun hale getirmek için ICP metodunun, bu ve bunun gibi problemliler için daha optimize hale getirmek ve otomatikleştirmek, ileride üzerine çalışılabilir konuların başında gelmektedir. Çapraz modalite kullanılarak derinlik haritası çıkartma uygulamaları için seçilen kameraların görüş alanlarının hemen hemen aynı seçilmesi, kullanılan LiDAR cihazının literatürde bu konu üzerinde kullanılan cihazlar gibi daha çok örnek alabilen cihazlardan seçilmesi ve referans alınan görüntü ile LiDAR verilerinin birbirlerine göre doğrultma işlemlerinin daha doğru yapılması, veri kümelerinin uyumluluğu açısından bu uygulamayı daha başarılı hale getirecektir. Ayrıca söz konusu çapraz modalite olunca, ağ yapılarında da güncellemeye gidilmesi başarı performansını artıracaktır. Örneğin, farklı spektrumda alınmış iki görüntünün mevcut uygulamalarda olduğu gibi ortak özellik çıkartma işlemine sokulması yerine, ayrı ayrı özellik çıkartma işlemine sokulup segmentasyonunun yapılması, daha sonra bu segmente edilmiş görüntüler üzerinden derinlik tahmini yapması derinlik tahmininde başarı oranını artıracak bariz hususlardan biridir.

KAYNAKLAR

- Albrecht, P., and Michaelis, B. (1998). Improvement of the Spatial Resolution of an Optical 3-D Measurement Procedure. *IEEE Transactions on Instrumentation and Measurement*, 47(1), 158-162.
- Arun, K., Huang, T., and Blostein, S. (1987). Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5), 698-700.
- Beier, K. R., Boehl, R., Fries, J., Hahn, W., Hausamann, D., Tank, V., Wagner, G., and Weisser, H. (1994). Measurement and modeling of infrared imaging systems at conditions of reduced visibility (fog) for traffic applications. W. R. Watkins and D. Clement (Eds.), *Characterization and Propagation of Sources and Backgrounds*. Bellingham, USA: SPIE, pp. 175-186.
- Bertozzi, M., Broggi, A., Lasagni, A., and Rose, M. (2005). *Infrared stereo vision-based pedestrian detection*. Intelligent Vehicles Symposium, 24-29, Las Vegas, USA.
- Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 679-698.
- Chang, J. R., and Chen, Y. S. (2018). *Pyramid Stereo Matching Network*. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5410-5418, Salt Lake City, USA.
- Chen, W., Luo, X., Liang, Z., Li, C., Wu, M., Gao, Y., and Jia, X. (2020). A Unified Framework for Depth Prediction from a Single Image and Binocular Stereo Matching. *Remote Sensing*, 12(3), 588.
- Dhua, A., Cutu, F., Hammoud, R., and Kiselewich, S. (2003). *Triangulation based technique for efficient stereo computation in infrared images*. IEEE IV2003 Intelligent Vehicles Symposium, 673-678, Columbus, USA.
- Dougherty, E. R. (1992). *An introduction to morphological image processing*. Bellingham, USA: SPIE Optical Engineering Press, 10-160.
- Du, X., El-Khamy, M., and Lee, J. (2019). *AMNet: Deep atrous multiscale stereo disparity estimation networks*. Paper presented at the 2019 Computer Vision and Pattern Recognition, Long Beach, USA.
- Geiger, A., Lenz, P., and Urtasun, R. (2012a). *Are we ready for autonomous driving? The KITTI vision benchmark suite*. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, 3354-3361, Rhode Island, USA.
- Geiger, A., Moosmann, F., Car, Ö., and Schuster, B. (2012b). *Automatic camera and range sensor calibration using a single shot*. In 2012 IEEE International Conference on Robotics and Automation, 3936-3943, St. Paul, USA.

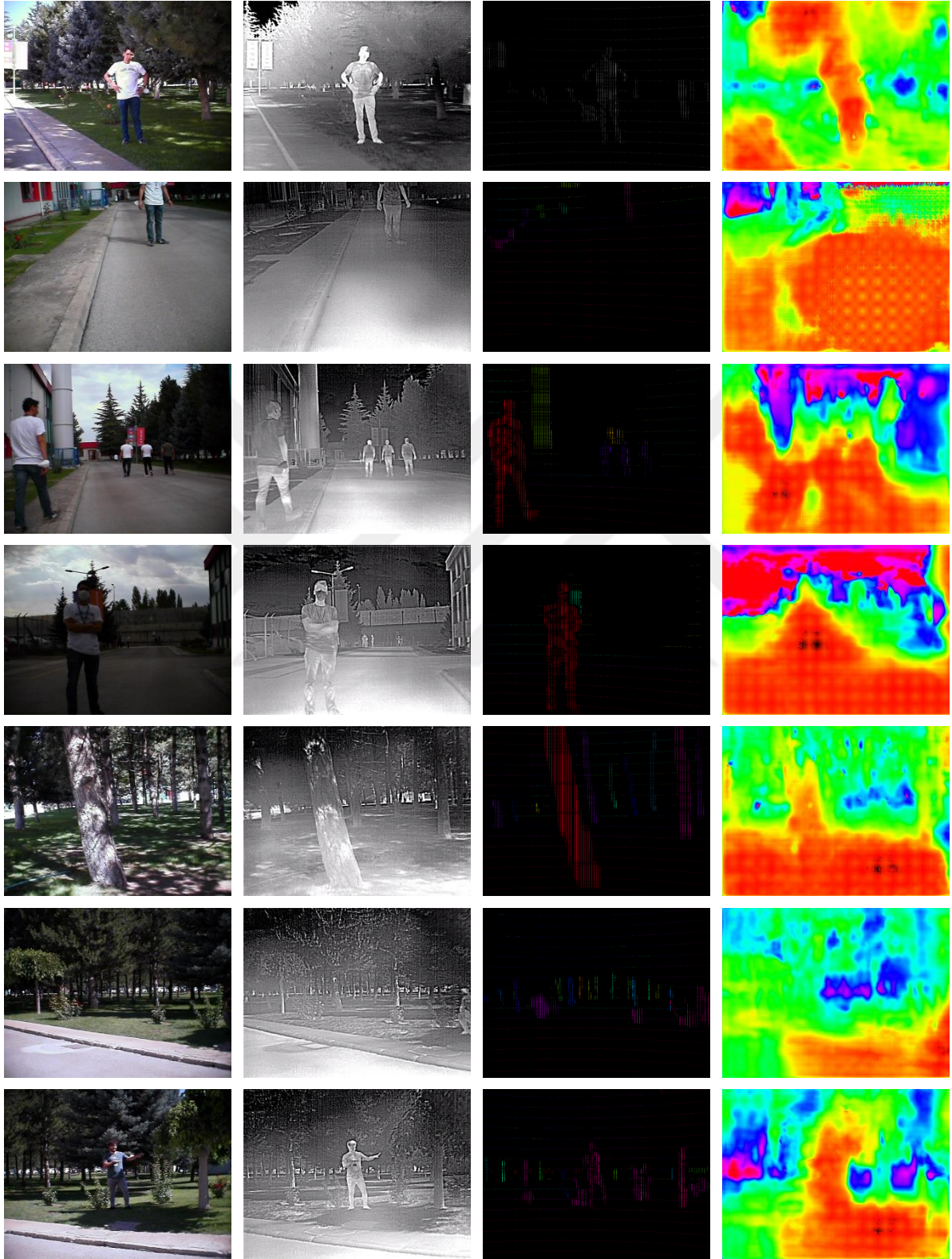
- He, K., Zhang, X., Ren, S., and Sun J. (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. D. Fleet, T. Pajdla, B. Schiele and T. Tuytelaars (Eds.), *Lecture Notes in Computer Science*. New York, USA: Springer International Publishing, pp 346–361.
- Hirschmuller, H. (2008). Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2), 328-341.
- Iqbal, J., and Basha, S. (2017). *Real time 3D depth estimation and measurement of uncalibrated stereo and thermal images*. In 2017 International Conference on Nascent Technologies in Engineering, 1-6, New Jersey, USA.
- İnternet: Autonomous vehicles need thermal cameras. (2020). URL-5: <https://www.foresightauto.com/autonomous-vehicles-need-thermal-cameras/>, Son Erişim Tarihi: 04.08.2021.
- İnternet: Can Thermal Imaging See Through Walls? And Other Common Questions. (2019). URL-4: <https://www.flir.com/discover/cores-components/can-thermal-imaging-see-through-walls/>, Son Erişim Tarihi: 04.08.2021.
- İnternet: Image Formation. (2008). URL-7: <https://courses.cs.duke.edu/fall08/cps274/notes/ImageFormation.pdf>, Son Erişim Tarihi: 04.08.2021.
- İnternet: Infrared Basics. (t.y.). URL-3: https://www.pro-therm.com/infrared_basics.php, Son Erişim Tarihi: 04.08.2021.
- İnternet: LIDAR – Wikipedia. (2021) URL-1: <https://tr.wikipedia.org/wiki/LIDAR>, Son Erişim Tarihi: 04.08.2021.
- İnternet: Setup and Calibration. (t.y.). URL-6: <http://bigdatavision.org/cats/setup.html>, Son Erişim Tarihi: 04.08.2021.
- İnternet: The FARO Focus S70 Lidar Hits 1 mm Accuracy. (2017). URL-2: <https://www.geoweekevents.com/news/faro-focus-s70-lidar-hits-1-mm-accuracy>, Son Erişim Tarihi: 04.08.2021.
- Ivašić-Kos, M., Krišto, M., and Pobar, M. (2019). *Human detection in thermal imaging using YOLO*. 5th International Conference on Computer and Technology Applications, 20-24, New York, USA.
- Jordan, M. I., and Bishop, C. M. (2004). Neural Networks. In A. B. Tucker (ed.). *Computer Science Handbook*, (Second Edition). Florida, USA: Chapman & Hall/CRC Press LLC, pp. 66.
- Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., and Bry, A. (2017). *End-to-end learning of geometry and context for deep stereo regression*. IEEE International Conference on Computer Vision, 66-75, Venice, Italy.

- Kingma, D., and Ba, J. (2015). *Adam: A method for stochastic optimization*. Paper presented at the 3rd International Conference on Learning Representations, San Diego, USA.
- Kim, N., Choi, Y., Hwang, S., and Kweon, I. S. (2018). Multispectral Transfer Network: Unsupervised Depth Estimation for All-Day Vision. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 6983-6991.
- Liu, Z., Chow, P., Xu, J., Jiang, J., Dou, Y., and Zhou, J. (2019). A Uniform Architecture Design for Accelerating 2D and 3D CNNs on FPGAs. *Electronics*, 8(1), 65.
- Lu, Y., and Lu, G. (2021). *An alternative of LiDAR in nighttime: unsupervised depth estimation based on single thermal image*. IEEE Winter Conference on Applications of Computer Vision, 3832-3842, Waikoloa, USA.
- Mallon, J., and Whelan, P. (2005). Projective rectification from the fundamental matrix. *Image and Vision Computing*, 23(7), 643-650.
- Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. (2015). *A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation*. 2016 IEEE Conference on Computer Vision and Pattern Recognition, 4040-4048, Las Vegas, USA.
- McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2), 153–157.
- Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of Machine Learning*. (Second Edition). Boston, USA: The MIT Press, 6-7.
- O’Riordan, A., Newe, T., Dooly, G., and Toal, D. (2018). *Stereo vision sensing: review of existing systems*. 12th International Conference on Sensing Technology, 178-184, Limerick, Ireland.
- Pang, J., Sun, W., Ren, J. S. J., Yang, C., and Yan, Q. (2017). *Cascade residual learning: A two-stage convolutional neural network for stereo matching*. 2017 IEEE International Conference on Computer Vision Workshops, 878-886, Venice, Italy.
- Rodríguez-Quíñonez, J. C., Sergiyenko, O., Flores-Fuentes, W., Rivas-lopez, M., Hernandez-Balbuena, D., Rascón, R., and Mercorelli, P. (2017). Improve a 3D distance measurement accuracy in stereo vision systems using optimization methods’ approach. *Opto-Electronics Review*, 25(1), 24-32.
- Sanz, P. R., Ruiz Mezcua, B. R., and Pena, J. M. S. (2012). Depth Estimation – An Introduction. In A. Bhatti (ed.). *Current advancements in stereo vision* (Chapter 5). Croatia: InTech Prepress, pp. 93-114.
- Scharstein, D., Szeliski, R., and Zabih, R. (2001). *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*. Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision, 131-140, Kauai, USA.

- Serra, J. (1983). *Image analysis and mathematical morphology*. London, U.K.: Academic Press, 184-185.
- Serra, J. (1988). *Image analysis and mathematical morphology, theoretical advances*. London, UK: Academic Press, 13-35.
- Shin, U., Kweon, I., Lee, K., and Lee, S. (2021). *Unsupervised depth and ego-motion estimation for monocular thermal video using multi-spectral consistency loss*. Paper presented at the 2021 IEEE Conference on Computer Vision and Pattern Recognition, Online Conference.
- Treible, W., Saponaro, P., Sorensen, S., Kolagunda, A., O’Neal, M., Phelan, B., Sherbondy, K., and Kambhamettu, C. (2017). *CATS: A color and thermal stereo benchmark*. IEEE Conference on Computer Vision and Pattern Recognition, 134-142, Venice, Italy.
- Trucco, E., and Verri, A. (1998). *Introductory techniques for 3-D computer vision*. Upper Saddle River, USA: Prentice Hall PTR, 24-27.
- Wang, Q., Shi, S., Zheng, S., Zhao, K., and Chu X. (2020). *FADNet: A fast and accurate network for disparity estimation*. IEEE International Conference on Robotics and Automation, 101-107, Paris, France.
- Zagoruyko, S., and Komodakis, N. (2015). *Learning to compare image patches via convolutional neural networks*. IEEE Conference on Computer Vision and Pattern Recognition, 4353-4361, Boston, USA.
- Zhang, F., Prisacariu, V. A., Yang, R., and Torr, P. H. S. (2019). *Guided aggregation net for end-to-end stereo matching*. IEEE Conference on Computer Vision and Pattern Recognition, 185-194, Long Beach, USA.
- Zhu, J. Y., Park, T., Isola, P., and Efros, A. A. (2017). *Unpaired image-to-image translation using Cycle-consistent adversarial networks*. IEEE International Conference on Computer Vision, 2242-2251, Venice, Italy.
- Zoetgnande, Y. W. K., Cormier, G., Fougères, A. J., and Dillenseger, J. L. (2020). Sub-pixel matching method for low-resolution thermal stereo images. *Infrared Physics & Technology*, 105, 103161.

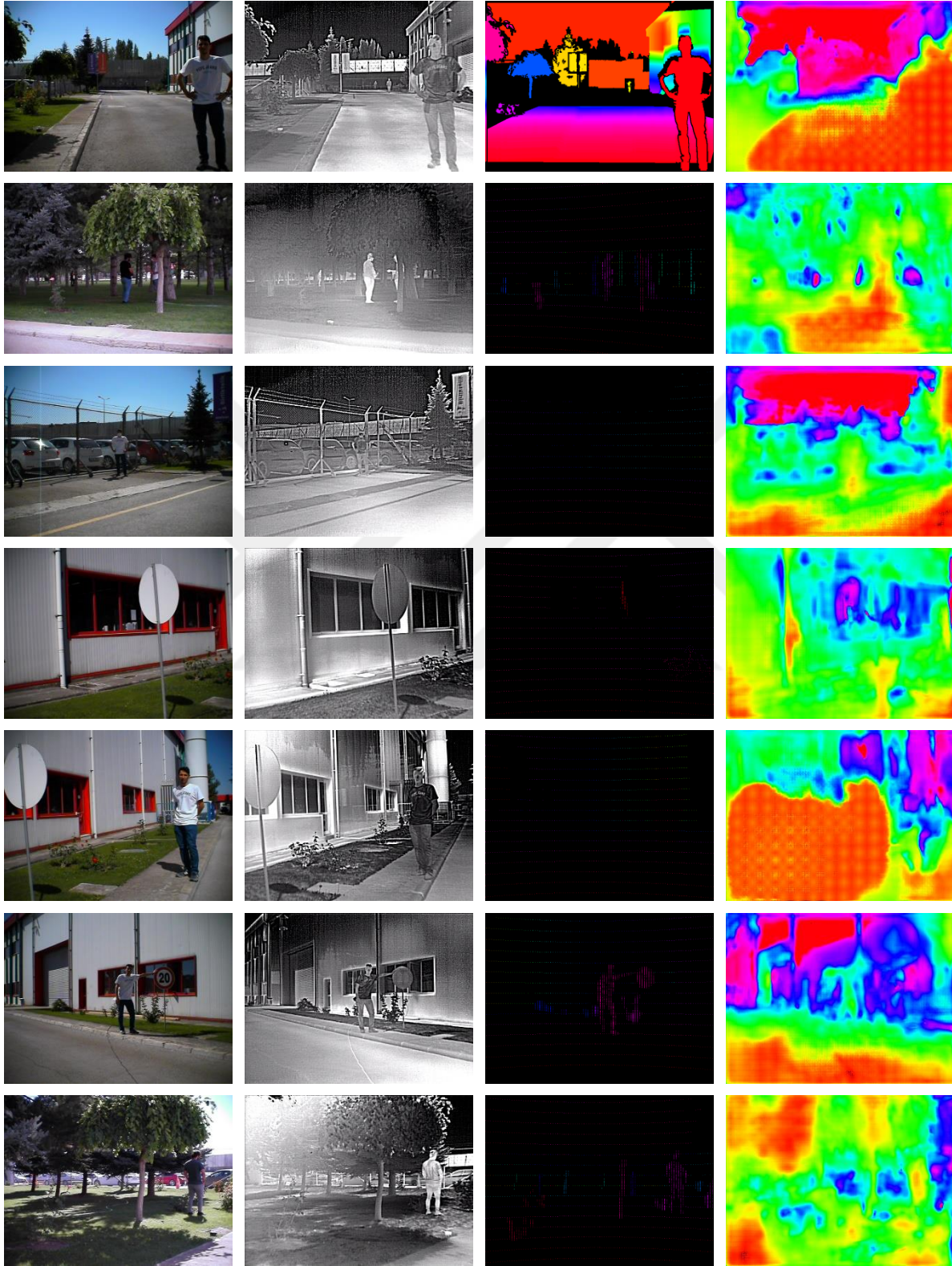


EK-1. Oluşturulan veri kümesi ile eğitilen FADNet ağının test çıktıları



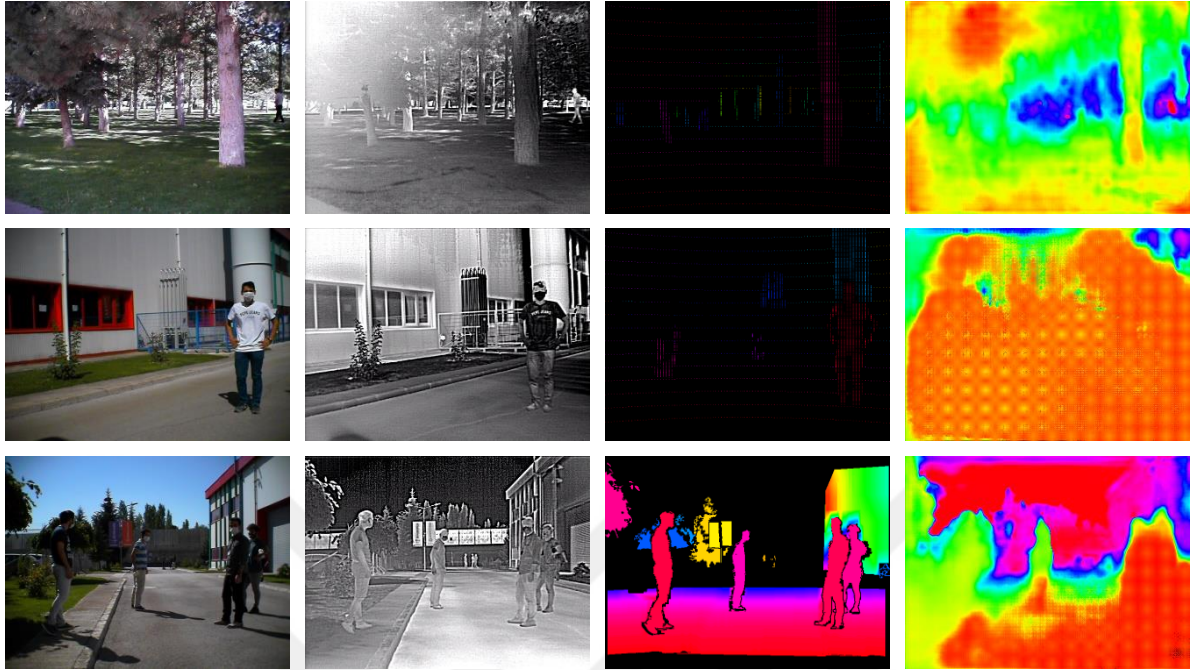
Resim 1.1. Oluşturulan veri kümesi ile eğitilen FADNet ağının test çıktıları

EK-1. (devam) Oluşturulan veri kümesi ile eğitilen FADNet ağının test çıktıları



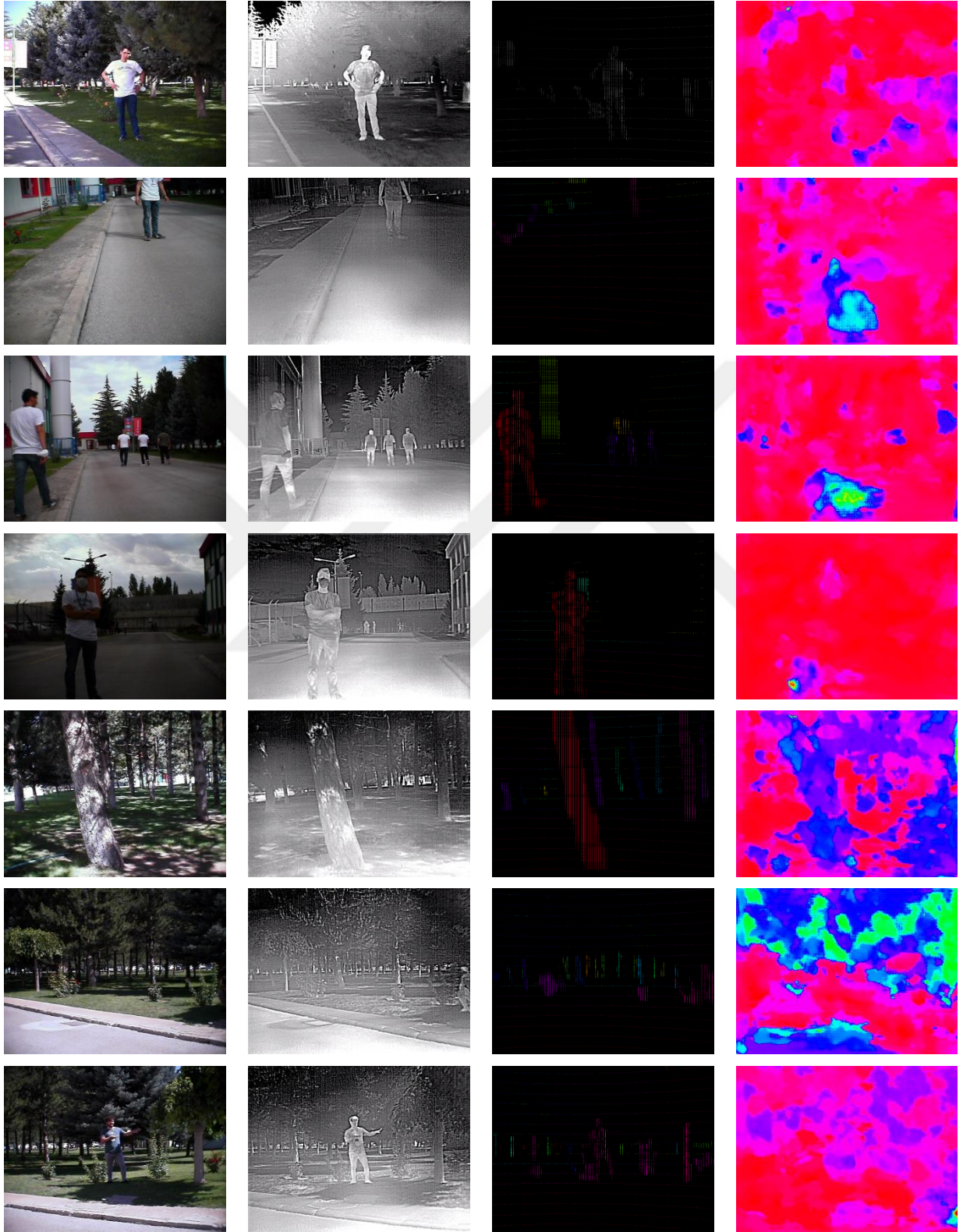
Resim 1.1. (devam) Oluşturulan veri kümesi ile eğitilen FADNet ağının test çıktıları

EK-1. (devam) Oluřturulan veri kümesi ile eğitilen FADNet ağıının test çıktıları



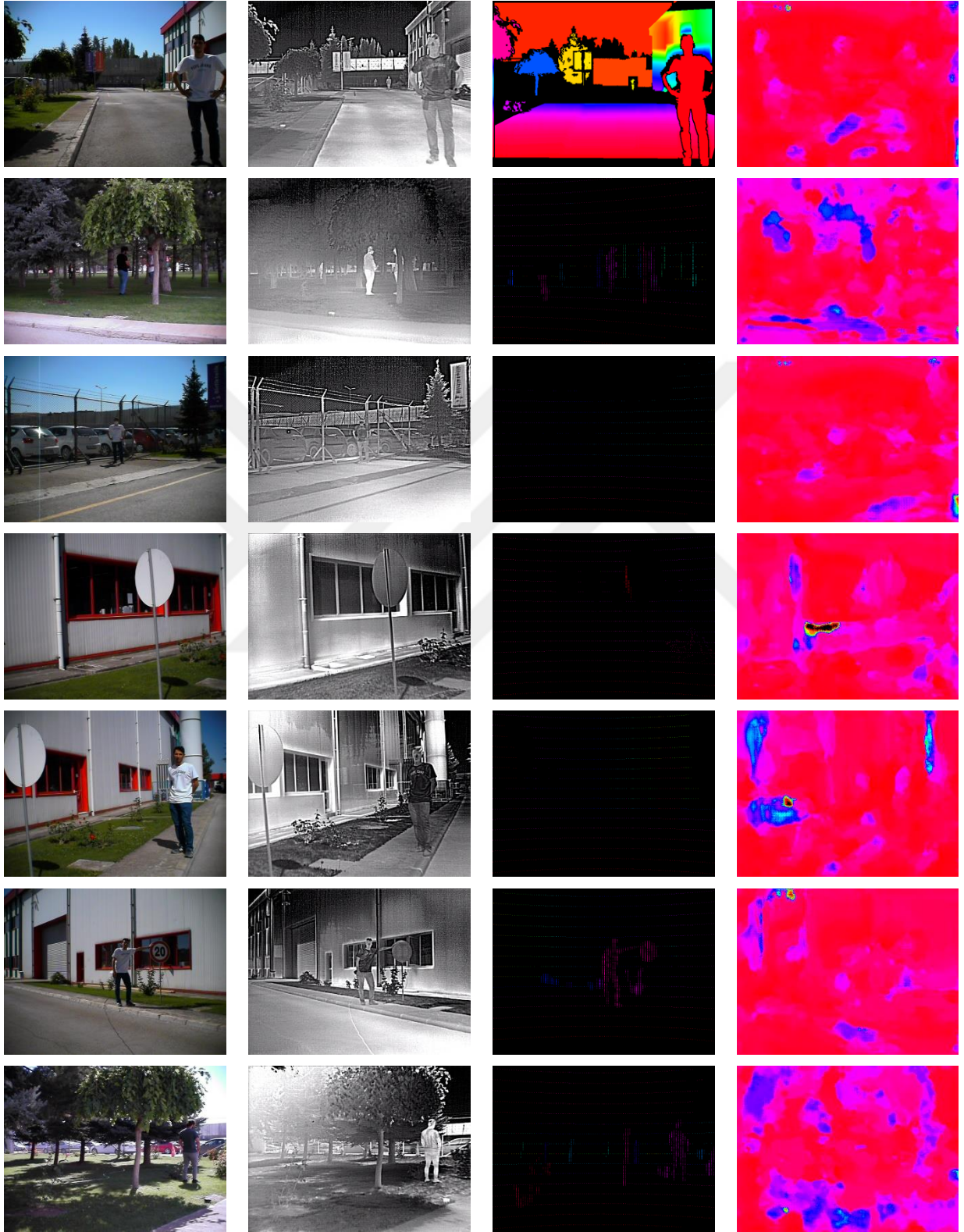
Resim 1.1. (devam) Oluřturulan veri kümesi ile eğitilen FADNet ağıının test çıktıları

EK-2. Oluşturulan veri kümesi ile eğitilen PSMNET ağının test çıktıları



Resim 2.1. Oluşturulan veri kümesi ile eğitilen PSMNet ağının test çıktıları

EK-2. (devam) Oluşturulan veri kümesi ile eğitilen PSMNET ağının test çıktıları



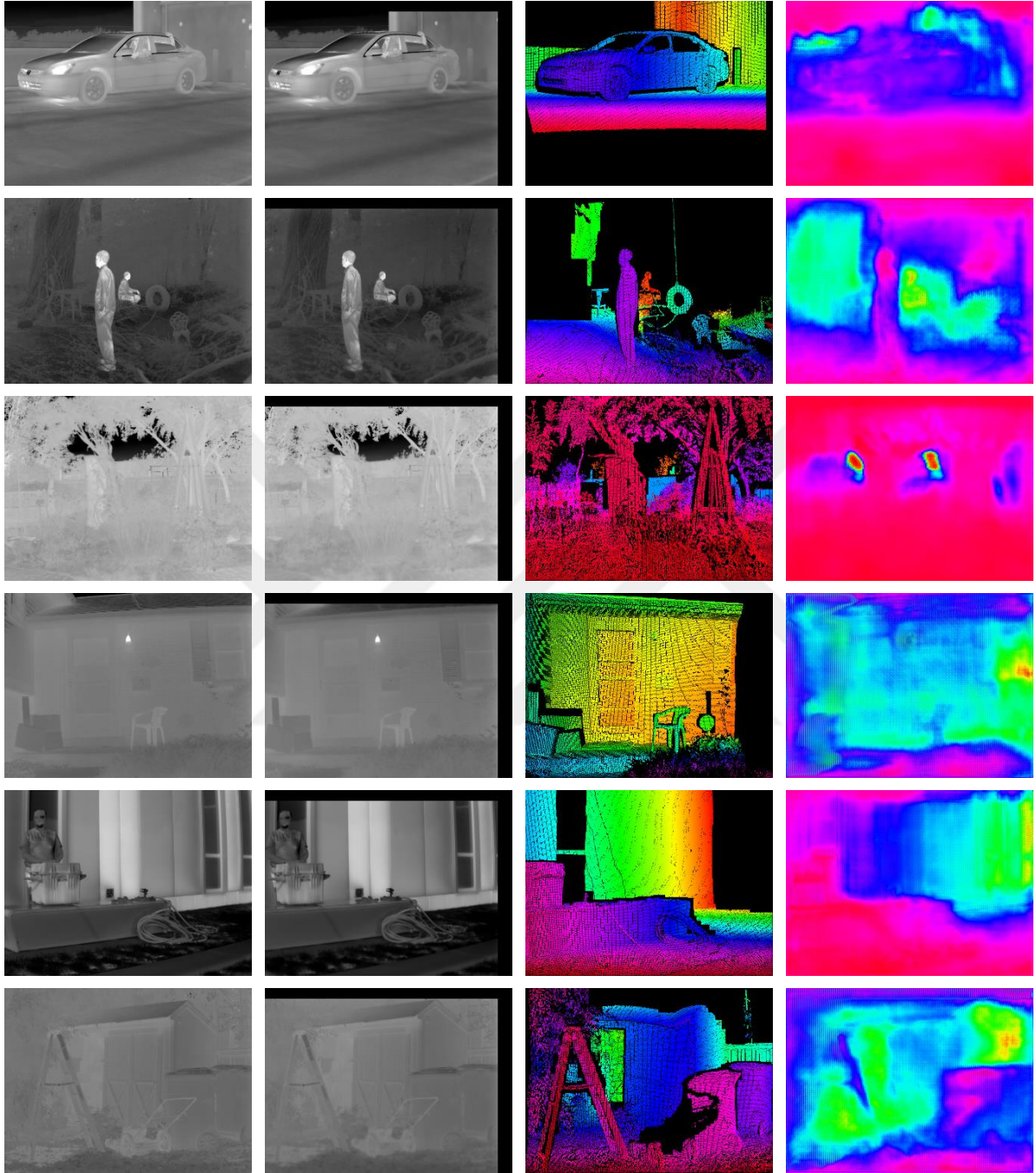
Resim 2.1. (devam) Oluşturulan veri kümesi ile eğitilen PSMNet ağının test çıktıları

EK-2. (devam) Oluşturulan veri kümesi ile eğitilen PSMNET ağının test çıktıları



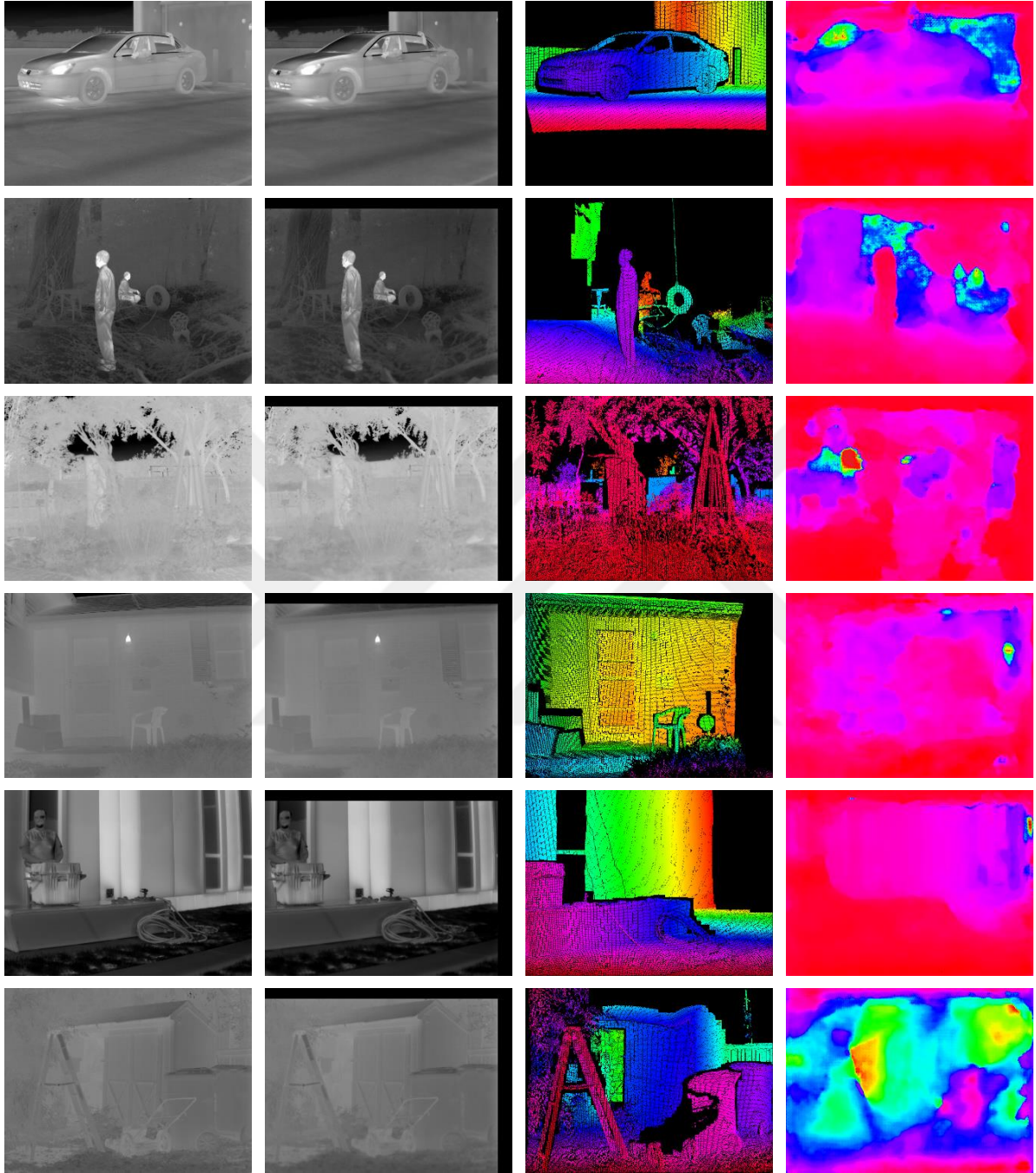
Resim 2.1. (devam) Oluşturulan veri kümesi ile eğitilen PSMNet ağının test çıktıları

EK-3. CATS veri kümesi ile eğitilen FADNet ağının test çıktıları



Resim 3.1. CATS veri kümesi ile eğitilen FADNet ağının test çıktıları

EK-4. CATS veri kümesi ile eğitilen PSMNet ağının test çıktıları



Resim 4.1. CATS veri kümesi ile eğitilen PSMNet ağının test çıktıları

EK-5. Kullanılan kavramlar ve İngilizce karşılıkları

Çizelge 5.1. Kullanılan kavramlar ve İngilizce karşılıkları

KAVRAMLAR	İNGİLİZCE KARŞILIKLARI
Artık bloklar	Residual blocks
Parti boyutu	Batch size
Gerçek referans değer	Ground truth
Kayıp değer	Loss
Fazla uyum	Overfitting
Yetersiz uyum	Underfitting
Eğitim turu	Epoch
Eğilmiş	Warped
Kaydırma adımı	Stride
Ters evrişim	Deconvolution
Bağlanım	Regression

EK-6. Diğer Görüntü Benzerlik Metrikleri ile Elde Edilmiş Sonuçlar

Çizelge 6.1. Diğer Görüntü Benzerlik Metrikleri ile Elde Edilmiş Sonuçlar

		FADNET					PSMNET				
		PSNR	RMSE	SSIM	FSIM	SRE	PSNR	RMSE	SSIM	FSIM	SRE
CATS	Eğitim	34,498	0,021	0,706	0,495	50,583	31,011	0,032	0,524	0,432	49,582
	Test	33,807	0,023	0,673	0,463	50,358	29,042	0,038	0,401	0,428	49,186
Oluşturulan Veri Kümesi	Eğitim	30,622	0,030	0,566	0,481	41,634	28,386	0,039	0,336	0,421	41,576
	Test	30,530	0,030	0,566	0,481	41,388	28,395	0,038	0,321	0,419	41,596
İdeal		Sonsuz	0	1	1	Sonsuz	Sonsuz	0	1	1	Sonsuz



GAZİ GELECEKTİR..