

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**Çok Katmanlı Küresel Derinlik Parametreleri ve Derin Öğrenme Yöntemleri İle  
3B Şekil Tamamlama ve Üretme**



**YÜKSEK LİSANS TEZİ**

**Abdüllatif AĞCA**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Tez Danışmanı: Doç. Dr. F. Betül ATALAY SATOĞLU**

**Ağustos 2021**



# ÖZET

Yüksek Lisans Tezi

Çok Katmanlı Küresel Derinlik Parametreleri ve Derin Öğrenme Yöntemleri İle 3B Şekil Tamamlama ve Üretme

Abdüllatif AĞCA

TOBB Ekonomi ve Teknoloji Üniversitesi  
Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. F. Betül ATALAY SATOĞLU

Tarih: Ağustos 2021

Yapay sinir ağları ile 3B geometri üretimi uzun süredir üzerine çalışılan bir konudur. Yapay sinir ağları, yapıları gereği verileri kendilerine özgü formatlarda kabul etmekte olduğundan, 3B geometri verisi bu formatlarda temsil edilmek suretiyle yapay sinir ağlarına sağlanmalıdır. İki boyutlu evrişimsel sinir ağının tek veya üç boyutlu veri kabul etmeyişi bu duruma bir örnektir. Fakat, 3B veriler genelde nokta bulutu ve poligon ağ gibi tek boyutta sıralı, düzensiz ve anlamlandırılması zor formatlarda tutulduğu için ilgili verinin yapay sinir ağınca anlamlandırılabilir bir formata dönüştürülmesi gerekir. Bunu yaparken hem 3B şeklin özelliklerinin tümü veya çoğu korunmalı hem de verimli yaklaşımlar kullanılmalıdır, çünkü 3B veriler oldukça çok sayıda öznitelik içermektedir. Polinomyal anlamda derece değişimi, hem öğrenme süresini hem de test süresini büyük ölçüde düşürecektir.

Bu çalışmanın ilk iki kısmında, evrişimsel yapay sinir ağlarıyla ve çok katmanlı küresel derinlik parametreleri kullanılarak 3B şeklin eksik kısmının küresel derinlik uzayında tamamlaması yapılmış, elde edilen sonuç ise üç boyutlu uzaya dönüştürülmüştür. Çok katmanlı küresel derinlik parametreleri işleme  $O(n^2)$  zaman karmaşasına sahip olup,  $O(n^3)$  zaman karmaşasına sahip olan doluluk haritası gibi klasik yaklaşımlara nazaran çok daha verimli ve yüksek performanslı bir yöntemdir. Burada  $n$  terimi, öznitelik verisinin eksen başına aldığı örnekleme sayısıdır. 2B bir  $n \times n$  görselde  $n^2$  olası örnek

bulunurken  $n \times n \times n$  boyutlara sahip bir doluluk haritasında  $n^3$  olası örnek bulunur. Örnek sayısı, olası sayıdan az olabilir, bazı pikseller ve vokseller boş olabilir. Yöntemimiz  $O(n^2)$ 'lik "state of the art" yaklaşımlarla da kıyaslanabilir bir yöntemdir. Tamamlama işlemi 2B küresel derinlik görselleri üzerinden sürdürüleceği için kullanılacak olan yöntemler de yine 2B görüntü tamamlamada kullanılan yöntemler olacaktır. Evrişimsel sinir ağı modeli olarak regresyon yapan U-Net tabanlı tek bir evrişimsel sinir ağı kullanılmıştır. Elde edilen sonuçlar, ustalık derecesinde bir çalışmanın aynı örneklerle çalıştığıında elde edilen sonuçlarla detaylı bir şekilde karşılaştırılıp sonuçlar analiz edilmiştir. Çalışmanın ikinci kısmında ise çok katmanlı küresel derinlik parametreleri dörtlü ağaca (quad tree) bölünüp yalnızca her katmandaki yaprak boğumlar evrişim işlemine tâbi tutulmuştur. Bu şekilde, evrişim işlemi detay gerektiren bölgelere daha çok sayıda uygulanırken detay gerektirmeyen yerlere az sayıda uygulanmıştır. Şekil tamamlama işlemi de bu çerçevede optimize edilmiştir. Çalışmanın üçüncü kısmında ise koşullu çekişmeli üretken ağlar kullanılarak rastgele küresel derinlik görselleri üretilmiş, üretilen rastgele görseller öğrenilen sınıflara ait rastgele ve özgün şekiller oluşturmuştur.

**Anahtar Kelimeler:** Küresel derinlik parametreleri, Evrişimsel sinir ağları, Çekişmeli üretken ağlar.

## ABSTRACT

Master of Science

3D Shape Completion and Generation using Multilayer Spherical Depth Parameters  
with Deep Learning Methods

Abdüllatif AĞCA

TOBB University of Economics and Technology  
Institute of Natural and Applied Sciences  
Department of Computer Engineering

Supervisor: Assoc. Prof. F. Betül ATALAY SATOĞLU

Date: August 2021

3D geometry generation with artificial neural networks has been a topic worked on for a long time. Because the artificial neural networks accept their inputs in specific formats, 3D geometry information has to be in the related formats before passing them as a parameter to the artificial neural networks. A two dimensional convolutional neural network does not accept input in one or three dimensions, which is an example to that case. However, because the 3D geometry information is stored in point clouds and polygon meshes most of the time which are stored in one dimension in an irregular way, the data has to be converted to a format which is meaningful to a neural network. This conversion must preserve most or all of the information the 3D geometry contains and it has to be optimal in terms of memory/time complexity because 3D geometry information contains a lot of features. Decrease in the polynomial degree decreases both the learning time and the testing time of the neural network drastically.

In the first two chapters of this work, completion of missing parts of the 3D shape is done using Multilayer Spherical Depth Parameters with convolutional neural networks, and the resulting images is converted to the 3D space. Processing time complexity of the multilayer spherical depth parameters is  $O(n^2)$ , which is a lot efficient and having better performance than the traditional  $O(n^3)$  time complexity approaches like 3D occupancy maps. Here the  $n$  term implies the number of possible samples in each of the axes. In an  $n \times n \times n$  voxel map there are  $n^3$  possible samples while in a 2D  $n \times n$  image,

there are  $n^2$  possible samples. Number of samples can be different than the number of possible samples. Some of the pixels and voxels may not store information. Our method is also comparable to the  $O(n^2)$  state of the art approaches. Because the inpainting will take place in 2D image space, the methods going to be used will be 2D image processing and computer vision techniques. Multilayer spherical depth parameters were inpainted with a U-Net based regression network. The results of that approach were also analyzed and compared to a state of the art 3D shape inpainting approach. In the second chapter, multilayer spherical depth parameters were divided into an octree and the convolution operations were applied on that octree representation to reduce the number of operations by eliminating unnecessary details. This way, the inpainting operation has been optimized. In the third and the last chapter, spherical depth parameters were generated by a conditional generative adversarial network and this way, authentic samples of the learned classes has been generated.

**Keywords:** Spherical depth parameters, Convolutional neural networks, Generative adversarial networks.

## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖZET</b> . . . . .	ii
<b>ABSTRACT</b> . . . . .	iv
<b>İÇİNDEKİLER</b> . . . . .	vi
<b>ŞEKİL LİSTESİ</b> . . . . .	viii
<b>ÇİZELGE LİSTESİ</b> . . . . .	xi
<b>KISALTMALAR</b> . . . . .	xii
1. GİRİŞ . . . . .	1
2. Literatür Araştırması . . . . .	4
2.1 Doluluk Haritası Temelli Yaklaşımlar . . . . .	5
2.2 Tek veya Çok Açıdan 2B İzdüşümsel Yaklaşımlar . . . . .	6
2.3 Tek Katmanlı Küresel Parametrizasyon Temelli Yaklaşımlar . . . . .	8
2.4 Sekizli Ağaç Temelli Yaklaşımlar . . . . .	10
2.5 Geometri Resimleri . . . . .	11
3. Çok Katmanlı Küresel Derinlik Parametreleriyle 3B Şekil Tamamlama . . . . .	13
3.1 Çok Katmanlı Küresel Derinlik Parametreleri . . . . .	16
3.2 Evrişimsel Sinir Ağları, U-Net ve Kısmi Evrişimsel Sinir Ağları . . . . .	20
3.2.1 Evrişimsel Sinir Ağları . . . . .	20
3.2.2 U-Net . . . . .	24
3.2.3 Kısmi Evrişimsel Sinir Ağları (Partial Convolutional Neural Networks) . . . . .	24
3.3 Kısmi Evrişimsel Sinir Ağı ve U-Net Tabanlı Maske Tahmin Edici ile Çok Katmanlı Küresel Derinlik Parametreleri Tamamlama . . . . .	28
3.3.1 Eğitim Verisi Hazırlama . . . . .	29
3.3.2 Sinir Ağı Modeli . . . . .	30
3.3.3 Deney . . . . .	32
3.4 U-Net Tabanlı Sinir Ağıyla Çok Katmanlı Küresel Derinlik Parametreleri Tamamlama . . . . .	32
3.4.1 Eğitim Verisi Hazırlama . . . . .	32
3.4.2 Sinir Ağı Modeli . . . . .	33
3.4.3 Önem Tabanlı Masraf Hesaplaması . . . . .	35
3.4.4 Deney . . . . .	37
3.4.5 Chamfer Mesafesi . . . . .	39
3.4.6 Sonuçlar ve Analiz . . . . .	40
4. Adaptif Küresel Derinlik Parametreleriyle 3B Şekil Tamamlama . . . . .	45
4.1 Eğitim Verisi Hazırlama . . . . .	45
4.2 Sinir Ağı Modeli . . . . .	48
4.3 Deney . . . . .	49

5. Koşullu Çekişmeli Üretken Sinir Ağları ve Küresel Derinlik Paramet- releriyle Öğrenilmiş Sınıf Özgün 3B Şekil Üretimi . . . . .	51
5.1 Çekişmeli Üretken Sinir Ağları ve Koşullu Çekişmeli Üretken Si- nir Ağları . . . . .	51
5.2 Koşullu Çekişmeli Üretken Ağ Eğitimi . . . . .	53
5.3 Deney . . . . .	56
5.4 Sonuçlar . . . . .	58
6. Sonuçlar ve Öneriler . . . . .	60
<b>Kaynakça . . . . .</b>	<b>63</b>





## ŞEKİL LİSTESİ

Şekil 2.1: 3B Üçgenleştirilmiş Geometrik Şekil . . . . .	4
Şekil 2.2: Doluluk Haritası ile Modellenmiş 3B Şekil . . . . .	5
Şekil 2.3: Tek Katmanlı Küresel Derinlik Parametreleri . . . . .	8
Şekil 2.4: Doluluk Haritası ve Sekizli Ağaç Kesiti Kıyaslaması. Bu örnekte, 64 yerine yalnızca 13 tane veri kullanılarak şeklin temsil edilmesi örneklenmiştir. . . . .	10
Şekil 2.5: OCNN'in Çoklu Girdi Kodlama Mekanizması . . . . .	11
Şekil 3.1: Çok katmanlı küresel derinlik parametreleri çıkarımı ve üç katman için küresel derinlik görselleri. [21] çalışmamızda kullandığımız şeklin değiştirilmiş versiyonudur. (© 2021 IEEE) . . . . .	16
Şekil 3.2: Çok katmanlı küresel derinlik parametrelerinin 3B Öklit uzayında karşılığı (katmanlara göre renklendirilmiş nokta bulutu). Görselleştirme MeshLab [2] ile elde edilmiştir. . . . .	17
Şekil 3.3: ÇKKDP Öznitelik Çıkarıcı Sözde Kod. [21] çalışmamızda kullandığımız şekildir. (© 2021 IEEE) . . . . .	18
Şekil 3.4: Doluluk Haritası - Küresel Derinlik Parametreleri Örnek Hassasiyeti Kıyaslaması . . . . .	19
Şekil 3.5: Yukarıdaki şekilde $[ix, iy, z]$ boyutlarında bir görsel ve $[kx, ky, z]$ boyutlarında iki adet evrişim filtresi bulunmaktadır. Z eksenindeki örnekleme sayısının aynı olması önemlidir. . . . .	21
Şekil 3.6: Yukarıdaki şekilde $[ix, iy, z]$ boyutlarında bir görsel üzerine $[kx, ky, z]$ boyutlarında iki adet evrişim filtresi uygulandıktan sonra ortaya çıkan şeklin hangi pikselinin hangi evrişim operasyonundan geldiği gösterilmektedir. Kırmızı çıktı kanalı kırmızı filtrenin sonucu olup mavi çıktı kanalı mavi filtrenin sonucudur. Koyu kırmızı ile gösterilen piksel kırmızı filtrenin ilgili andaki operasyonunun sonucunun 1. kanalın hangi pikseline yazıldığını gösterirken koyu mavi ile gösterilen piksel mavi filtrenin ilgili andaki operasyonunun sonucunun 2. kanalın hangi pikseline yazıldığını göstermektedir. . . . .	22
Şekil 3.7: İleri besleme ve geri dağılım evrişim işlemleri . . . . .	22
Şekil 3.8: U-Net . . . . .	25
Şekil 3.9: [22] çalışmasında kullanılan masraf bileşenleri. . . . .	26
Şekil 3.10: Kısmi Evrişimsel Sinir Ağları [22] . . . . .	28
Şekil 3.11: Kısmi Evrişimsel Tamamlayıcı Ağ Kompleksi . . . . .	29
Şekil 3.12: Kısmi Evrişim Deneyi: Eksik, Tamamlanmış ve Taban Doğru (Ground Truth) İmajlar . . . . .	31
Şekil 3.13: Kısmi Evrişim Deneyi: Tahmin Edilmiş ve Taban Doğru Maskeler . . . . .	31

Şekil 3.14: Kümülatif Maskeleye İşlemi: Bu işlem, eğitim için gerekli eksik geometri bilgisini tutarlı bir şekilde hazırlamak için uygulanır. Öncelikle her bir katmandan ayrı ayrı veri silme işlemi gerçekleşir. Fakat, en dıştaki katmandan bir nokta silindiği zaman, bir alt katmanda aynı doğrultuda yani aynı $(u, v)$ koordinatında bir nokta varsa, bu alt katmandaki nokta artık en dıştaki nokta olacağından onu alıp bir üst imaja yazmak gerekir. Bir alt katmanda yoksa şayet, iki alt katmana bakılır. Eğer orada varsa aynı işlem o nokta için uygulanır. Bu işlem bütün katmanlara ve her piksele uygulandığı takdirde her bir pikselin hangi katmanda hangi değere sahip olacağı tekrar sıralanmış olur. Bu işlem, sentetik eksik verinin tutarlı bir küresel derinlik görseli paketi oluşturması için elzemdir. Kümülatif maskeleye, adını buradan alır. En üstteki katman yalnızca kendi maskesine muhatap iken, bir alt katman hem kendi maskesine göre veri kaybeder, hem de üsttekinin maskesine göre üst katmana alınması gereken verileri üst katmana verir. Dolayısıyla hem üst katmanın hem kendisinin maskesine muhatap olur. İç katmanlara indikçe bu sayı artmaktadır. . . . .	34
Şekil 3.15: Değiştirilmiş U-Net . . . . .	35
Şekil 3.16: Küresel Derinlik İmajının Kenar Tespiti (Üç ile çarpılmış değerler). Kenar tespit aracı olarak Alex Clark ve diğer katılımcıların, Fredrik Lundh ve diğer katılımcılar tarafından geliştirilen PIL kütüphanesi üzerine geliştirilen Pillow Python kütüphanesi kullanılmıştır [25]. . . . .	37
Şekil 3.17: Soldaki şekilde mavi ve kırmızı iki farklı nokta bulutu bulunmaktadır. Bu iki farklı nokta bulutu arasındaki Chamfer Mesafesi'ni hesaplamak için iki nokta bulutu arasında en yakın nokta çiftleri bulunur, eşleşme yapıldıktan sonra mavi nokta bulutunun kırmızıya olan ortalama mesafesi hesaplanır, kırmızı mavi nokta bulutunun mavi nokta bulutuna olan ortalama mesafesi hesaplanır ve ikisi toplanır. . . . .	39
Şekil 3.18: $5 \times 10^{-3}$ öğrenme oranıyla modelin eğitim esnasındaki eğitim ve validasyon verisi masraf değeri grafiği . . . . .	41
Şekil 3.19: $5 \times 10^{-4}$ öğrenme oranıyla modelin eğitim esnasındaki eğitim ve validasyon verisi masraf değeri grafiği . . . . .	42
Şekil 3.20: Otomobil Şekli Tamamlama ( $\sim 38$ eksik geometri ile). Soldan Sağa Doğru: Taban Doğru, Eksik Geometri, OCNN with OGSC, ÇKKDP $lr=10^{-3}$ , ÇKKDP $lr=10^{-4}$ . Görselleştirme MeshLab [2] ile elde edilmiştir. . . . .	43
Şekil 3.21: Otomobil Şekli Tamamlama ( $\sim 52$ eksik geometri ile). Soldan Sağa Doğru: Taban Doğru, Eksik Geometri, OCNN with OGSC, ÇKKDP $lr=10^{-3}$ , ÇKKDP $lr=10^{-4}$ . Görselleştirme MeshLab [2] ile elde edilmiştir. . . . .	43
Şekil 3.22: Sandalye Şekli Tamamlama ( $\sim 40$ eksik geometri ile). Soldan Sağa Doğru: Taban Doğru, Eksik Geometri, OCNN with OGSC, ÇKKDP $lr=10^{-3}$ , ÇKKDP $lr=10^{-4}$ . Görselleştirme MeshLab [2] ile elde edilmiştir. . . . .	43

Şekil 3.23: Sandalye Şekli Tamamlama ( $\sim 56$ eksik geometri ile). Soldan Sağa Doğru: Taban Doğru, Eksik Geometri, OCNN with OGSC, ÇKKDP $lr=10^{-3}$ , ÇKKDP $lr=10^{-4}$ . Görselleştirme MeshLab [2] ile elde edilmiştir. . . . .	43
Şekil 3.24: Kanepeler Şekli Tamamlama ( $\sim 42$ eksik geometri ile). Soldan Sağa Doğru: Taban Doğru, Eksik Geometri, OCNN with OGSC, ÇKKDP $lr=10^{-3}$ , ÇKKDP $lr=10^{-4}$ . Görselleştirme MeshLab [2] ile elde edilmiştir. . . . .	44
Şekil 3.25: Kanepeler Şekli Tamamlama ( $\sim 55$ eksik geometri ile). Soldan Sağa Doğru: Taban Doğru, Eksik Geometri, OCNN with OGSC, ÇKKDP $lr=10^{-3}$ , ÇKKDP $lr=10^{-4}$ . Görselleştirme MeshLab [2] ile elde edilmiştir. . . . .	44
Şekil 4.1: Tek Katmanlı Küresel Derinlik Parametrelerinin Dörtlü Ağaca (Quad Tree) Bölünmüş Görüntüsü . . . . .	46
Şekil 4.2: Adaptif Çok Katmanlı Küresel Derinlik Parametreleri . . . . .	47
Şekil 4.3: Adaptif Küresel Derinlik Tamamlayıcı U-Net . . . . .	49
Şekil 4.4: Olması Gereken Şekil - Overfit Testi Sonucu Kıyaslaması . . . . .	50
Şekil 5.1: Koşullu Çekişmeli Üretken Ağ Örneği: Üretken ağın girdilerinden $r$ vektörü rastgele gizli vektörü temsil ederken $s$ girdisi ilgili verinin hangi sınıfa ait olduğunu veya olması gerektiğini gösterir. Şeklin yukarıda kalan kısmı eğitimin ayırt edici ağı eğitime kısmını gösterirken aşağıda kalan kısmı üretken ağın eğitimini gösterir. . . . .	54
Şekil 5.2: Çekişmeli Üretken Ağlar ile Üretilen Bir Otomobil Modeli ve Eğitim Veri Kümesinde Kendisine En Çok Benzeyen Otomobil Modelinin Karşılaştırılması (Üretilen model turuncu renkte iken eğitim verisi sarı renktedir). Resimlerde pikseller $[0, 255]$ aralığında değer almaktadır. İki resim arasındaki ortalama karesel hata 24.634 olarak ölçülmüştür. Görsel olarak da sayısal olarak da iki şeklin farklı olduğu görülmektedir. Üretilmiş şekildeki dikiz aynası, rüzgarlık, tampon çizgileri gibi detaylar çekişmeli üretken ağların ne denli detaylı modelleme yapabildiğini gösterirken, bu detayların şekle en benzeyen eğitim şeklindekilerden farklı oluşu da ağın mod çöküşü yaşamadığını göstermektedir. . . . .	55
Şekil 5.3: Çekişmeli Üretken Ağlar ile Üretilmiş Rastgele Otomobil Şekilleri . . . . .	56
Şekil 5.4: Çekişmeli Üretken Ağlar ile Üretilmiş Rastgele Sandalye Şekilleri . . . . .	56
Şekil 5.5: Çekişmeli Üretken Ağlar ile Üretilmiş Rastgele Kanepeler Şekilleri . . . . .	57
Şekil 5.6: Çekişmeli Üretken Ağlar ile Üretilmiş Hatalı Koltuk Şekli . . . . .	59
Şekil 5.7: Kanat ve Dikey Sabitleyici Üretimi Bozuk ve Eksik Olan Uçak Şekli (3B görüntü Blender [35] programı ile elde edilmiştir) . . . . .	60



## ÇİZELGE LİSTESİ

Çizelge 3.1: CHAMFER MESAFESİ SONUÇLARI . . . . .	40
Çizelge 5.1: AZAMI BENZERLİK SONUÇLARI . . . . .	58





## KISALTMALAR

- ÇKKDP** : Çok Katmanlı Küresel Derinlik Parametreleri  
**GAN** : Generative Adversarial Networks  
**OCNN** : Octree Convolutional Neural Networks  
**OGSC** : Output Guided Skip Connections







## 1. GİRİŞ

3B şekil bilgisinin yapay sinir ağları tarafından işlenmesi, 3B uzayda eksik bilgilerin tahmini, sınıflandırılması ve özgün 3B şekil üretilmesi gibi konularda kritik rol oynar. Yapay sinir ağları, girdi verileri arasındaki ilişkiyi girdi verisinin formatı yardımıyla kurduğundan, 3B şeklin yapay sinir ağı tarafından öğrenilebilir bir formatta sağlanması önemlidir. 3B eksik şekil tamamlama, eksik üç boyutlu geometrinin aslına uygun bir şekilde tamamlanmasıdır. Geometrinin eksik olması, pek çok nedenden kaynaklanabilir. 3B şekilleri dijital ortamlara dönüştüren veya bu bilgiyi dijital ortama dönüştürmeye yardımcı olan araçlara radar, lidar ve derinlik kameraları örnek gösterilebilir. Bu cihazlar 3B bilgiyi dijital ortama dönüştürürken sensörleri aracılığıyla gerçek dünyadan veri örnekleri toplarlar. Toplanan bu veriler dijital ortamda sayılarla temsil edilir. Bu veriler toplanırken çeşitli engellerle karşılaşılabilir. Bunlardan en temelleri, taranacak geometrinin önüne geçen başka geometrik şekillerdir. Sensör ile taranacak veri arasına giren veirler, geometrinin ilgili kısmının dijital ortamda eksik kalmasına yol açar. Geometrinin eksik çıkmasının bir diğer nedeni de fiziksel deformasyon olabilir. Mimari unsurlar, bitkiler ve günlük hayatta karşılaşılabileceğimiz, sabit duran unsurlar fiziksel koşulların da etkisiyle yıllar geçtikçe yıpranır ve deforme olur. Deformasyon bazen geometriden büyük bölümlerin kaybolması kadar şiddetli olabilir. Bir kısmı yıkılmış tarihi bir köprü, bir bina veya ağaç eksik 3B şekillere örnek gösterilebilir. Bu gibi unsurlar taranırken herhangi bir engelle karşılaşılmasa dahi dijital ortamdaki veri eksik olacaktır. Bu durumda ilgili eksik verinin tahmini için çeşitli yöntemlere ihtiyaç duyulacaktır. Analitik yöntemler bu tür tahmin uygulamalarında yeterli olmadığından öğrenme tabanlı yaklaşımlar ilgili verinin tahmininde önemli rol oynayacaktır. Bu sebeple 3B şekil tamamlama, radar ve lidar uygulamalarında, 3B tarayıcı ve modelleme uygulamalarında kullanıcıya kolaylık sağlayabilecek ek özellikler olup derin öğrenme kullanılarak bu görevler oldukça yüksek performanslı ve başarılı bir şekilde yerine getirilebilir.

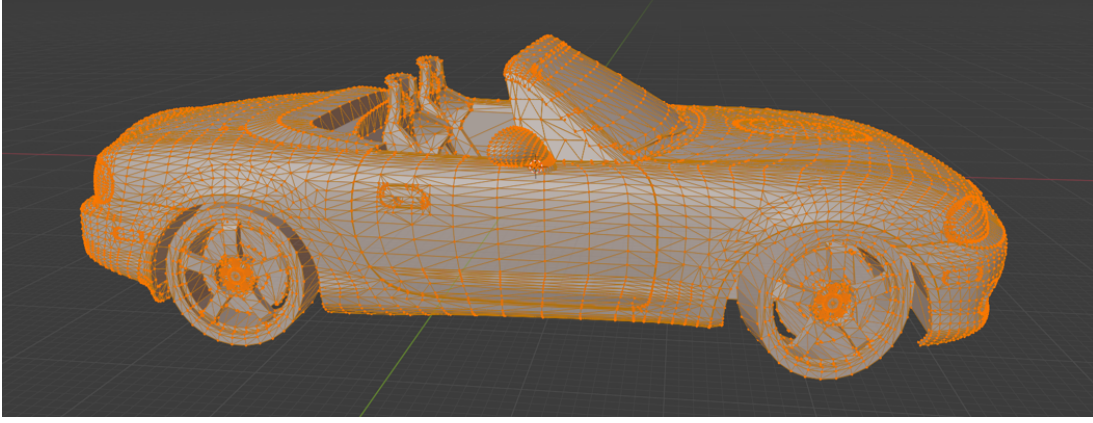
3B modelleme, sanatçılar, mimarlar ve daha pek çok meslek grubunun profesyonel uğraşının bir parçasıdır. 3B modelleme halihazırda var olan şekillerin dijital ortama akta-

rılmasını içerdiği gibi özgün şekillerin tasarlanmasını ve modellenmesini de içerebilir. 3B özgün tasarım, sanatçıların ortaya daha önceden görülmemiş ürünler koymasını ve mimarların daha önce eşine rastlanmamış tasarımlar yapmasına olanak tanır. Fakat 3B tasarım süreçleri oldukça zahmetlidir. 3B tasarım için oldukça iyi eğitilmiş 3B çizim profesyonelleri olmazsa olmazdır. Bunun yanısıra, 3B tasarım oldukça vakit alan bir süreçtir. Çizim yapılan evren 3B uzay olduğu için, noktaların uzayda dizilmesi ve bunlar arasına poligonlar çizilmesi, on binlerce poligon için oldukça çok vakit alan işlerdir. Bununla birlikte özgünlük genelde sanatçının veya mimarın tecrübesiyle ve hayal gücüyle doğrudan ilgilidir. Bireyler yaşamları boyunca edindikleri tecrübe ile özgün ürünler ortaya koyarlar. Fakat bu tecrübeyi edinmek çok uzun yıllar alır. Dolayısıyla 3B özgün tasarım yapmak ciddi özveri isteyen bir iştir. Bu bağlamda özgün tasarım konusunda tasarımcıya yardımcı olacak araçlar sürece dair büyük kolaylık sağlayacaktır. Özel bir tür yapay sinir ağı olan Çekişmeli Üretken Ağlar (Generative Adversarial Networks) tıpkı bir insan beyni gibi birtakım örnek veri ile kendini eğitip neticesinde bu örneklerle benzer örnekler üretebilir hale gelebilmektedir. Çekişmeli Üretken Ağlar'ın avantajı, eğitim süresinin insanlarda olduğu gibi seneler değil günler, nadiren haftalar oluşudur. Öte yandan, eğitim tamamlandıktan sonra özgün örnek üretim süresi çoğu zaman milisaniyelerle sınırlıdır, saniyede onlarcasını üretmesi mümkündür. Üretebileceği toplam örnek sayısı, referans aldığı girdi verisine göre üstel (exponential) yani test edilmesi dahi güç denebilecek kadar çok olduğu için her seferinde diğerlerinden farklı, özgün bir örnek üretmesi oldukça olasıdır. Bu yapay zeka modeli ile sınıfsal veri üretimi de mümkündür, tek bir ağ ile farklı sınıflara ait örnek modeller üretmek mümkündür. Dolayısıyla bu yaklaşım, 3B modelleme işiyle uğraşan sanatçılar, mimarlar ve diğer pek çok meslek grubu için referans model veya tasarım fikri üretmek açısından oldukça yardımcı olabilecek bir yaklaşımdır.

Bu çalışmada yukarıda bahsi geçen görevleri tamamlamak için geliştirilen tarafımızca geliştirilen yapay sinir ağı modelleri, bu modellerin artıları ve eksileri, uygulanan deneyler ve bu deneylerin sonuçları ele alınacaktır. Çalışmanın ilk kısmında eksik üç boyutlu geometri tamamlama işlemi için çok katmanlı küresel derinlik parametreleriyle eğitilmiş U-Net tipi bir yapay sinir ağı kullanılmıştır. İkinci başlıkta ise çok katmanlı

küresel derinlik parametreleri dörtlü ağaç yapısıyla optimize edilmiş, yeni öznitelik vektörü (feature vector) yine U-Net tipi bir yapay sinir ağını eğitmede kullanılmıştır. Üçüncü kısım ise koşullu çekişmeli üretken ağların küresel derinlik parametreleri ile eğitilmesini ve neticesinde eğitilen sınıflardan özgün örnekler üretilmesini ele alır.

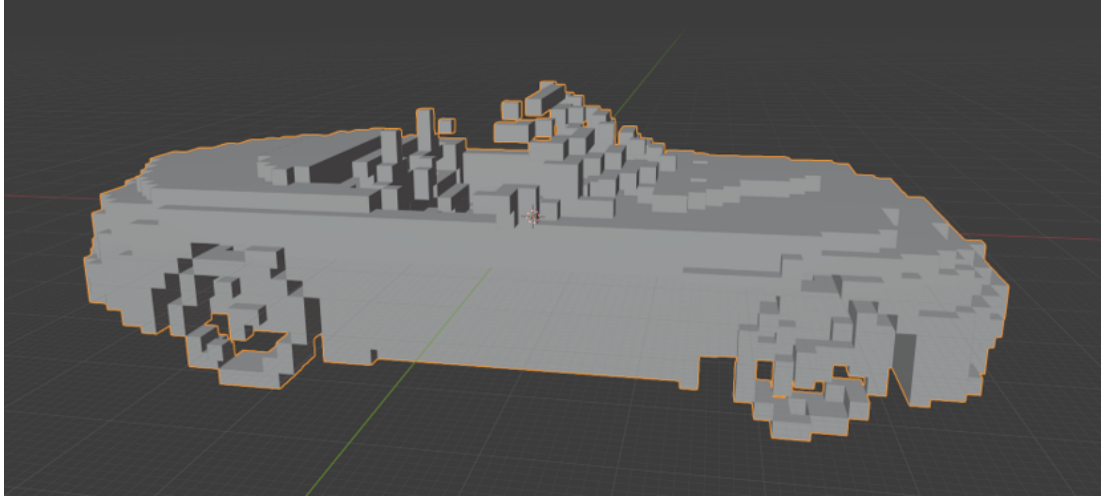




Şekil 2.1: 3B Üçgenleştirilmiş Geometrik Şekil

## 2. Literatür Araştırması

3B şekiller, genelde sanatçıların tasarımına veya 3B tarayıcıların tarama örüntüsüne bağlı yapılar olduğu için, 2B görsellerdeki düzenli ızgara (grid) biçiminde yapı 3B geometri bilgilerinde varsayılan olarak bulunmaz. Şekil 2.1’de buna örnek olarak poligonları sanatçının istediği şekilde ve düzensiz yerleştirilmiş bir üçgenleştirilmiş poligon ağı bulunmaktadır. Düzenli bir veri formu ve veriler arasındaki ilişkilerin bilgisi halihazırda bulunmadığı için bu bilgiler doğrudan yapay sinir beslediği takdirde yapay sinir ağının bu veride bir örüntü bulması oldukça zordur. Bunun sebebi, geometrik verilerin birbirinden bağımsız olmayışı, tahmin edilecek verinin civardaki diğer verilerle ilişkili veya onlara benzer bir değer alacak olması ve sağlanacak olan formatta diğer verilerle olan ilişkinin güzel gösterilmeyişinin bu veriler arasında bağıntı kurup değer tahmin etmeyi zorlaştırmasıdır. Bu yüzden bu verileri yapay sinir ağlarına beslemeden önce birtakım ön işlemlere tâbi tutup verileri anlamlı, örüntü bulunabilir ve şekil itibarıyla uygun bir formda yapay sinir ağlarına beslemek gerekmektedir. Bu amaç için çeşitli yaklaşımlar denenmiştir, bu yaklaşımlar avantajları ve dezavantajları, birbirlerine karşı üstünlükleri ve zayıflıkları ile birlikte literatür taramamızda yer almış olup birkaç ana başlık altında ele alınacaktır.



Şekil 2.2: Doluluk Haritası ile Modellenmiş 3B Şekil

## 2.1 Doluluk Haritası Temelli Yaklaşımlar

Doluluk haritası, 3B geometri bilgisinin tek bir küpün bütün hacmini dolduracak şekilde eş küplere bölünmesiyle elde edilir. Şekil 2.2’de doluluk haritası örneği bulunmaktadır. Bu alt küplerin herbirinde tutulmak istenen öznitelik bilgileri tutulur. Genelde bu bilgi sadece ilgili küpün 3B şeklin uzayda kapladığı hacmin içinde olup olmadığıdır. Bu veri yapısında tutulan veri, üç boyutta sürekli olduğu için  $O(n^3)$  karmaşaya sahiptir. Bu yaklaşım, akla ilk gelen yaklaşımlardan biri olduğu gibi bellek ve zaman karmaşası cinsinden optimal sayılmayacak bir yaklaşımdır.

Bu konudaki örnekleri kavrayabilmek adına çekişmeli üretken ağların temel anlamda ne olduğunu ve nasıl çalıştığını bilmek önemlidir. Çekişmeli üretken ağlar, üretken ağ ve ayırt edici ağ olarak iki alt ağ içerir. Üretken ağ, kendisine girilen gizli vektörü bir takım operasyonlara tâbi tutarak üretilmesi gereken formatta veri üretir. Ayırt edici ağ ise üretilen verinin gerçek veri veya üretken ağ tarafından üretilen veri olup olmadığını ayırt etmeye çalışır. Dolayısıyla iki ağdan herhangi birinin başarısı bir diğerinin başarısızlığı anlamına gelir. Bu ağlar, birbirlerinin başarısını masraf edinerek eğitilir. Eğitim tamamlandığında ayırt edici ağlar da üretken ağlar da kendisini geliştirmiş olur. Üretken ağlar gerçeğe yakın şekiller üretilip ayırt edici ağı yer yer kandırabilir hale gelmişken ayırt edici ağlar da yer yer üretken ağın ürettiği sonuçların üretim olduğunu

anlayabilir hale gelir. Bir ağın diğerine baskın gelmesi, baskın gelinen ağın üretimini durdururken eğitim başarısız sonuçlanır.

Wang ve arkadaşlarının [11] çalışmalarında 3B şekil tamamlamak için kullandığı öz-nitelik veri formatı, doluluk haritasına örnektir. Bu çalışmada Wang ve arkadaşları, 3B eksik şeklin tahmini için önce 3B bir çekişmeli üretken sinir ağı kullanmış, sonrasında da elde edilen tahminin yukarı örneklenmesi için evrimsel sinir ağı [31] tabanlı bir Uzun Kısa Vadeli Hafıza (Long Short Term Memory) sinir ağı [32] modeli kullanmıştır. 3B GAN girdi olarak 3B bir doluluk haritası alırken, Uzun Kısa Vadeli Hafıza sinir ağı girdi olarak 3B doluluk haritasının her biri 2B olan ardışık kesitlerini alıp, bunları yan yana koyarak nihai bir 3B doluluk haritası elde etmiştir. Elde edilen nihai doluluk haritasının çözünürlüğü, girdi doluluk haritasınınkinden yüksek olup, girdi haritasında eksik olan noktalar da çıktı haritasında tahmin edilmiş durumdadır.

Wu ve arkadaşları, çalışmalarında [13] üç boyutlu özgün şekiller üretmek için çekişmeli üretken sinir ağları kullanmışlardır. Bu çalışmada çekişmeli sinir ağlarında üç boyutlu evrimsel katmanlar kullanmışlardır. Öznitelik formatı olarak doluluk haritası kullanmışlardır. Üretken ağa verilen gizli vektörden itibaren yeniden şekillendirme katmanı (reshape) ve evrimsel katmanlar (convolutional layers) ile nihai bir doluluk haritası elde edilir. 3B ayırt edici ağ ise ilgili sonucun gerçek mi yoksa üretken ağın üretimi mi olduğunu ayırt etmeye çalışır. Bu iki ağ birbiriyle mücadele ederken üretken ağ özgün üç boyutlu şekil üretmeyi öğrenir. Üretilen şekil doluluk haritası formatındadır.

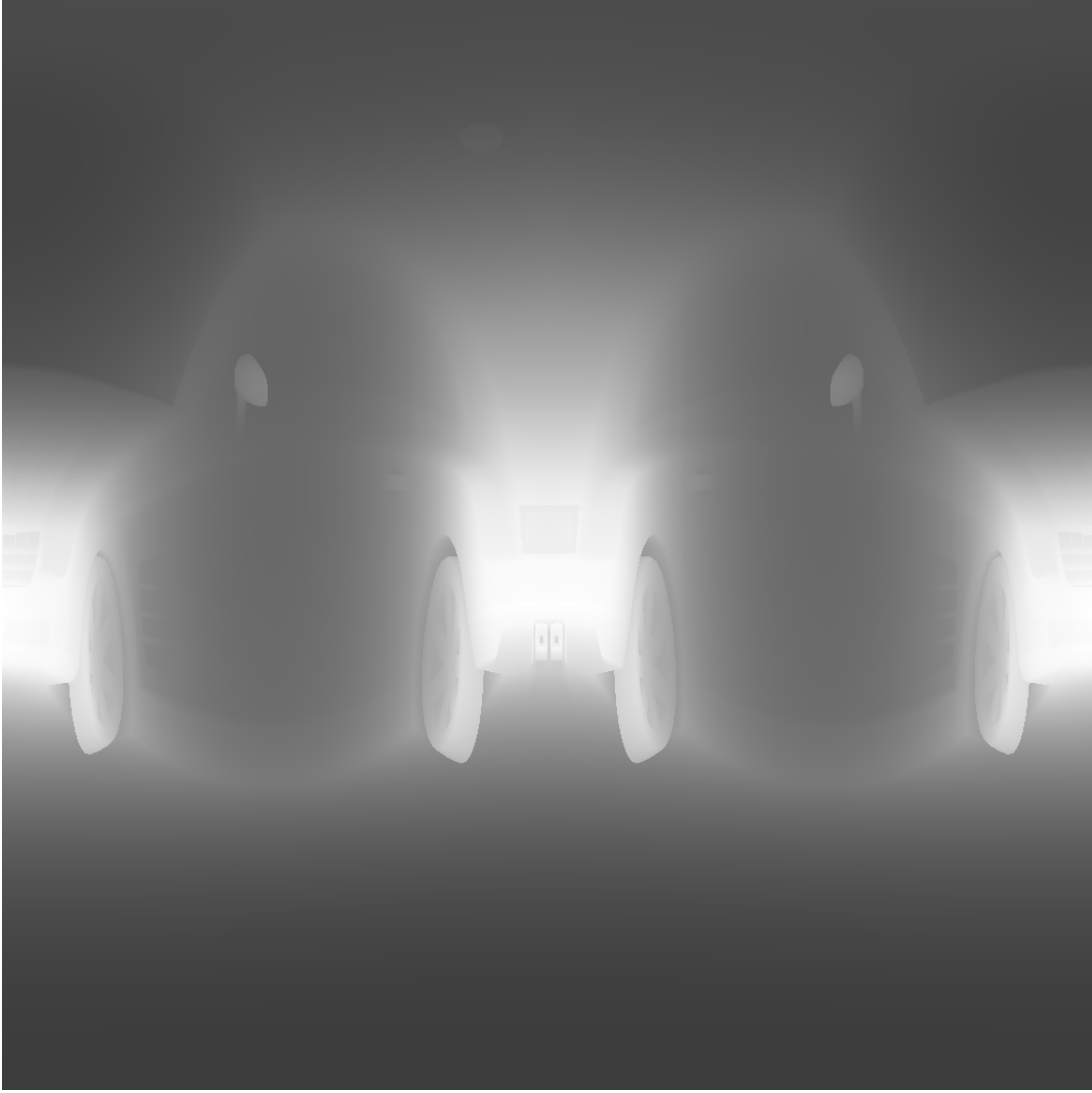
## **2.2 Tek veya Çok Açıdan 2B İzdüşümsel Yaklaşımlar**

Tek veya çok açıdan 2B izdüşümsel yöntemlerde şeklin derinlik veya renk bilgisi bir veya daha fazla 2B görselde tutulur, bu görsellerden yola çıkılarak şeklin 3B geometri bilgisi ile ilgili çıkarımlarda bulunulur. Bir veya daha fazla derinlik görseli üzerinde çalışan yaklaşımlar, bu görseller üzerindeki eksik bilgileri tamamlayarak çalışır. Birden fazla görsel üzerinde çalışan yöntemlerden bir kısmı farklı görsellerdeki derinlik

bilgisinin 3B uzaya dönüştürülmüş haldeki bilgilerini sentezleyerek nihai 3B şekli ortaya çıkarır. Tek derinlik görseli üzerinden çalışan yöntemlerden bir kısmı ise derinlik bilgisini 3B uzay bilgisine dönüştürüp 3B tamamlama sinir ağına bu bilgiyi besler.

Çok açıdan izdüşüm ile ilgili yönteme örnek olarak Hu ve arkadaşlarının [1] çalışması verilebilir. Bu çalışmada Hu ve arkadaşları eksik şeklin birden fazla 2B derinlik görselini tamamlamak üzere koşulsal üretken ve darboğaz şekilli bir 2B yapay sinir ağına beslemiş, her bir görsel için çıkan sonucu 3B uzaya dönüştürüp ortaya çıkan kısmi 3B şekli tutarlılık masraf hesaplama işlemine tâbi tutarak ve ortaya çıkacak olan hatayı minimize ederek birleştirmiştir.

Tek açıdan izdüşümle çalışan yöntemlere örnek olarak da Yang ve arkadaşlarının [15] çalışması örnek gösterilebilir. Bu çalışmada Yang ve arkadaşları, eksik geometrik şeklin tek açıdan derinlik izdüşümünü 3B uzaya çevirip doluluk haritasında tutmuş, bu doluluk haritasını ise 3B şekil tamamlayıcı koşulsal çekişmeli üretken ağlara beslemiş ve ağın çıktısı olarak da tamamlanmış şekli elde etmiştir. Kullandıkları sinir ağı, temel çekişmeli üretken ağlardan farklı olarak atlama/birleştirme ve yoğun katmanlar içermektedir.



Şekil 2.3: Tek Katmanlı Küresel Derinlik Parametreleri

### 2.3 Tek Katmanlı Küresel Parametrizasyon Temelli Yaklaşımlar

Tek katmanlı küresel parametrizasyon temelli yaklaşımlar, küresel parametrizasyon kullanarak 3B şeklin küresel uzaydaki derinlik bilgisini, her bir küresel açı örneği iki boyutta bir uv piksel koordinatına tekabül edecek şekilde 2B bir görsele izdüşerek modeller. Bu modelleme neticesinde elde edilen 2B görsel, 3B şeklin merkezinden atılan ışınların şekli kestiği noktalardan merkeze en uzak olanın merkeze uzaklık bilgisini tutar. Şekil 2.3'te bu durum gösterilmiştir. Bu uygulamada sıkıntılı olan, ışınların çoğu zaman birden fazla noktayı kesmesi ve bu noktalardan yalnızca en uzakta olanın



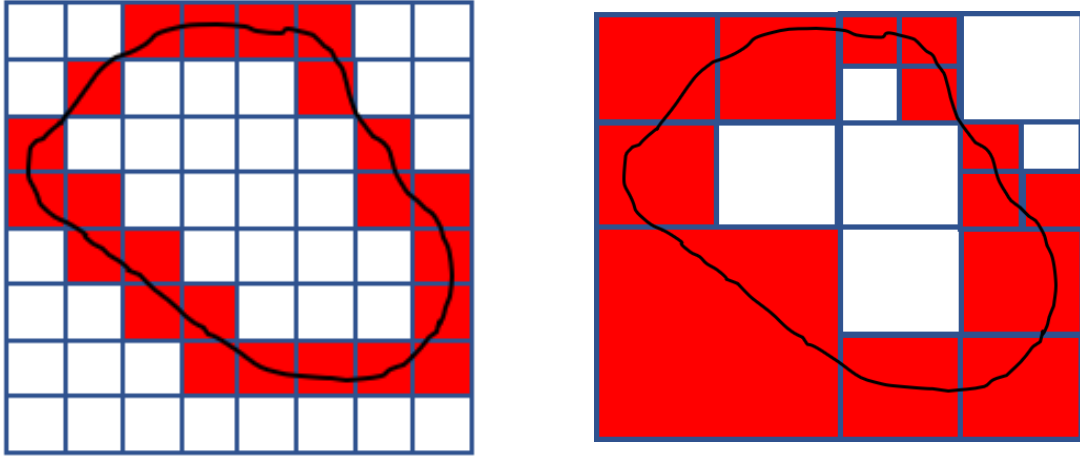
modellenmesinin şeklin çok büyük bir kısmını görselde yer almamasıdır. Bu şekilde yapılacak olan bir modellemede 3B geometri bilgisinin çok ciddi bir kısmı kaybolacaktır, tamamlanacak olan geometri bilgisinin yanında çok sayıda bilgi de kaybolacaktır. Buna örnek olarak Zhang ve arkadaşlarının [16] çalışması örnek gösterilebilir. İlgili çalışmada iki boyutlu RGB görselden derinlik haritası çıkarımında bulunulur. İlgili derinlik haritasının üç boyutlu uzayda tekabül ettiği noktaların küresel derinlik parametreleri çıkarılmıştır. İlgili parametreler iki boyutlu görsel tamamlayıcı sinir ağı ile tamamlanmış, elde edilen sonuç tekrar üç boyuta dönüştürülmüştür. Üç boyuttaki sonuç ile baştaki derinlikten elde edilen üç boyutlu veri birleştirilmiş ve bir doluluk haritası iyileştirme ağına girdi olarak verilmiştir. Bu ağ ise doluluk haritası işleyen üç boyutlu bir evrişimsel sinir ağıdır. Küresel derinlik parametreleri kullanılmak suretiyle tamamlama işleminin zaman karmaşası (time complexity)  $O(n^2)$  seviyesinde tutulduysa da sonradan kullanılan üç boyutlu doluluk haritası iyileştirme ağı ile zaman karmaşası  $O(n^3)$  seviyesine çekilmiştir. Küresel derinlik uzayında yapılan tamamlama işlemine ilaveten doluluk haritasına dönüşüm ve doluluk haritası üzerinden iyileştirme işlemi yapılması, tek katmanlı küresel derinlik parametrelerinin tek katmanlı oluşu nedeniyle atladığı detayları yeniden oluşturabilmek içindir. Wang ve arkadaşlarının GSIR [17] çalışmasında kullandığı yöntem de buna benzerdir. Öncelikle tek resimden derinlik tahmini yapmışlar, sonrasında öğrenilmemiş sınıfa dair şekil interpretasyonu yapmışlar, ilgili interpretasyon üzerinden tek katmanlı küresel derinlik parametresi çıkarmışlar ve nihayetinde de tıpkı [16] çalışmasında olduğu gibi doluluk haritası iyileştirme modülü kullanmışlardır. Benzer limitasyonlar GSIR için de geçerlidir. Cao ve arkadaşları [12] çalışmasında tek katmanlı ve dilimli küresel derinlik parametreleriyle 3B sınıflandırma yapmışlardır. Shi ve arkadaşlarının [4] çalışmasında kullandıkları yöntem de küresel parametrizasyona oldukça benzerdir. Bu çalışmada küresel parametrizasyon yerine silindirik parametrizasyon kullanılmıştır, bir sınıflandırma çalışmasıdır. Öz-kapanma gibi unsurlar bu çalışma için de geçerlidir. Shi ve arkadaşları bu çalışmada yalnızca tek katmanlı silindirik parametrizasyon tabanlı 2B sınıflandırma yapmıştır. Öznitelik çıkarım yöntemi itibarıyla benzer bir çalışmadır. Bu iki çalışmada esas amaç sınıflandırma olduğu için, tek katmanlı küresel derinlik parametreleri şekli

tanımak için yeterli bir öznitelik uzayı oluşturmuştur.

## 2.4 Sekizli Ağaç Temelli Yaklaşımlar

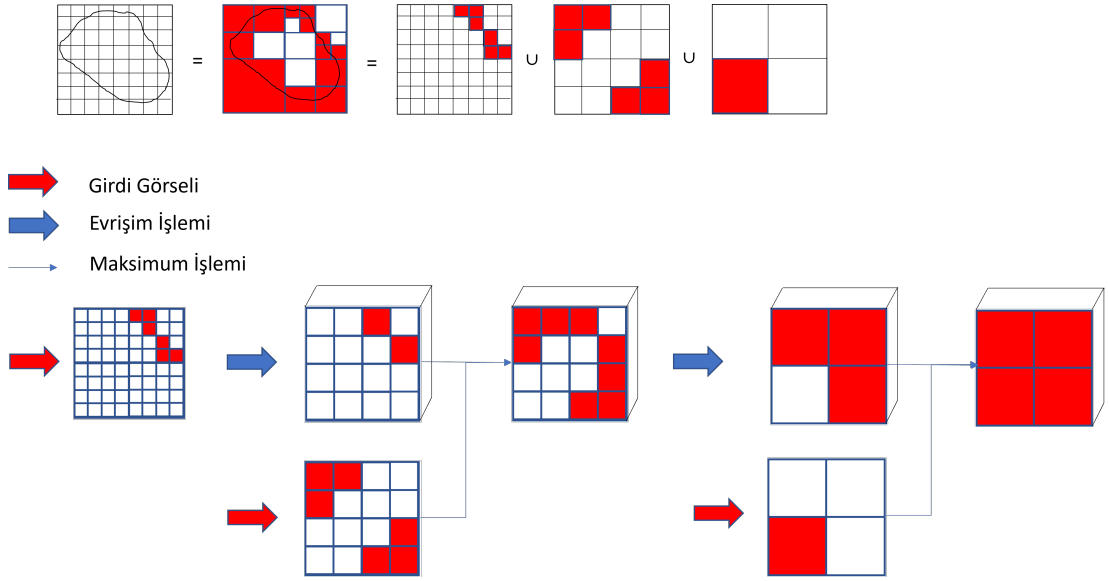
Sekizli ağaç temelli yaklaşımlar doluluk haritasının çok detay gerektiren kısımları detaylı olarak modellenmiş, az detay gerektiren kısımları da yüzeysel modellenmiş, ağaç yapısındaki formudur. Bu yaklaşımlardan bazıları Wang ve arkadaşlarının [8, 9, 14] çalışmalarıdır.

OCNN çalışmasında [8] 3B geometri kapsayıcı kutu içerisine alınır, her bir yaprak hücrenin (leaf cell) doluluk bilgisi ve bu hücreye tekabül eden yüzeyin yüzey normali sekizli ağaç (octree) yapısı formatında işaretlenir. 3B evrişim operasyonu da yalnızca ağacın veri tutan yaprak hücrelerine uygulanır. Eksik geometri de bu 3B evrişimlerle tahmin edilir. Yaprak hücrelerdeki yüzey normalleri doğrultusunda dörtgen yamalar oluşturulur ve tüm bu yamalar birleştirilip üç boyutlu şekil oluşturulur. Doluluk haritası kullanıldığında  $O(n^3)$  olan gereken 3B evrişim işlemi yalnızca  $O(n^2)$  karmaşa ile tamamlanmış olur. Şekil 2.4'te iki boyutlu kesit üzerinden aradaki bellek karmaşası farkı özetlenmiştir.



Şekil 2.4: Doluluk Haritası ve Sekizli Ağaç Kesiti Kıyaslaması. Bu örnekte, 64 yerine yalnızca 13 tane veri kullanılarak şeklin temsil edilmesi örneklenmiştir.

Adaptive OCNN çalışmasında [9] ise OCNN'den farklı olarak doluluk ve yüzey nor-



Şekil 2.5: OCNN'in Çoklu Girdi Kodlama Mekanizması

mali bilgisi yalnızca yaprak hücrelerde değil yüzey ile örtüşen bütün hücrelerde bulunur ve bu bilgi yapay sinir ağının farklı kodlama (encoding) katmanlarında ağa beslenir. Diğer bir deyişle, darboğaz tipte bir evrişimsel sinir ağının, darboğaza kadar gitmekte olan aşağı örnekleme/kodlama katmanlarına, farklı çözünürlükteki katmanlar için ilgili detay seviyesindeki sekizli ağaç hücreleri beslenir ve böylelikle evrişim operasyonu sekizli ağacın farklı oktantları arasındaki komşuluk ilişkisinden de beslenerek öğrenimini tamamlamış olur. Şekil 2.5'te detaylı akış diyagramı mevcuttur. Prosedürün kalan kısmı ve karmaşası OCNN ile aynıdır.

Output Guided Skip Connected OCNN ise bunlara ilaveten adından da anlaşılacağı üzere kodlama (encoding) ve deşifre (decoding) katmanları arasında atlama ve birleştirme bağlantıları kurar, bu yöntem de ağ bünyesindeki daha fazla sayıda parametre arasında bağlantı kurulmasını sağlar ve tahmindeki isabet oranını daha da artırır.

## 2.5 Geometri Resimleri

Geometri resimleri (Geometry Images), Gu ve arkadaşlarının [10] çalışmasında ele aldıkları yöntemdir. Küresel derinlik parametrelerinin geniş tanımıdır. Bu çalışmada 3B ağ (3D Mesh), üzerinde sınır kenar (boundary edge) olmayan kenarlar ayrılır ve sınır

kenar olan arkadaş kenarlar (mate edges) üretilir. ayrılan kenarlar kümesi bir grafikte bir kesi (cut) oluşturur. Bu şekilde 3B ağ (3D Mesh) disk topolojisine sahip forma dönüştürülür. Elde edilen ağ, sınır ve iç parametrisasyon yöntemleriyle 2B bir resime izdüşülür. Üç kanallı olan bu resim her bir kanalda sırasıyla nihai ağın  $[x, y, z]$  koordinatlarını tutar. İsteğe bağlı olarak yüzey normali haritası da çıkarılabilir.

Geometri resimlerinin derin öğrenme yöntemleriyle birlikte kullanılmasına örnek olarak Sinha ve arkadaşlarının [5] çalışması gösterilebilir. Bu çalışmada Sinha ve arkadaşları, 2B RGB resimden geometri resmi çıkarımı yapan ve parametrik temsilden (sınıf numarası ve poz bilgisi) geometri resmi çıkarımı yapan çalışmalar sunmuştur. Geometri resimlerinin bir sınırlaması, yalnızca cins-0 (genus-0) ağlar (meshes) ile çalışıyor olmasıdır. Daha yüksek sayıda cins (genus) için uygulamak adına kullanılacak yardımcı yöntemler olsa da, belirli bir seviyeden sonra bu yöntemler de uygulanabilir olmaktan çıkmaktadır. Öte yandan, şekil tamamlaması gibi görevleri doğrudan eksik imaj tahmini yapan yöntemlerle başarmak geometri resimlerini kullanarak nispeten zordur çünkü girdi ve çıktı resimlerinin ikisi de, yani eksik ve tam geometri resimlerinin ikisi de resim karesinin tümünü kullanıyor (100% utilization) olacaktır. Bu da, girdi ve çıktı resimleri arasında yani eksik ve bütün resimler arasında birebir ilişki kurmayı zorlaştıracaktır.

### 3. Çok Katmanlı Küresel Derinlik Parametreleriyle 3B Şekil Tamamlama

Bu çalışmada farklı bir öznitelik temsil yöntemi olarak çok katmanlı küresel derinlik parametreleri önerilmektedir. 3B şekil tamamlama işlemi için bu yöntemin tercih edilmiş olma nedeni, polinomyal anlamda optimal bir yaklaşım olması ( $O(n^2)$  zaman ve bellek karmaşıklığına sahiptir) ve geometriyi kapsayıcı bir şekilde modelleyebiliyor olmasıdır. Örnek uzaydaki şekiller daha fazla detay gerektirdikçe katman sayısı gerekli detayı destekleyecek kadar çok sayıda artırılabilir. Deneyler otomobil, kanepeler ve sandalye şekilleri için üç katmanın yeterli olduğunu gösterirken, uçak şekillerinde altı katman yeterli bulunmuştur. Çok katmanlı küresel derinlik parametreleri iki boyutlu ızgara formunda veri dizilimine sahiptir ve bu dizilimi işlemek için en ideal yaklaşımlardan biri de 2B evrişimsel sinir ağlarıdır. 2B evrişimsel sinir ağları görüntü işleme uygulamalarında yüksek performanslı olduğu ispatlanmış, sıkça kullanılan bir yöntemdir ve 2B regresyon işi için hem yapı itibarıyla uygundur, hem başarılı olma olasılığı yüksektir.

2B evrişimsel sinir ağları kendi içerisinde de farklı türlere ayrılır ve bu türler hiper-parametrelerine göre farklı derecede başarımları gösterebilir. Hiper-parametreler, öğrenilebilir parametrelerin aksine eğitim esnasında değil, farklı eğitim denemeleri arasında değiştirilen parametrelerdir. Parçalı değerlerle denemeler yapıldığı için farklı hiper-parametre kombinasyonlarından yalnızca en önemli bilgi içerenler not alınır ve en iyi sonuç veren model nihai olarak kullanılır.

Çok katmanlı küresel derinlik parametreleriyle 3B şekil tamamlama işlemi için üç farklı çalışmamız olmuştur.

1) Bunlardan birincisi kısmi evrişimlerle eksik görsel tamamlamadır [22]. Bu yöntemde evrişim operasyonuna eksik görselin yanısıra görselin neresinin eksik olduğunu gösteren maske görseli de beslenir. Maske görselinde derinlik bilgisi içermeyen piksellerin değeri sıfır iken derinlik bilgisi içeren piksellerin değeri birdir. Bütün ağ boyunca görselin verisinin yanısıra bu maske verisi de ileri beslenir fakat geri dağılmaz. Masraf hesaplamasına maske kanalının sonucu dahil edilmez, bu sonuç yalnızca esas görse-

lin masrafının hesaplanmasında öğrenilmiş bir maske olarak ve hesaplama esnasında farklı katsayılarla çarpılarak kullanılır. Bu şekilde masraf fonksiyonunun sonucu maskenin sınırladığı alan ve görselin kalan kısmı için ayrı ayrı ve önem tabanlı hesaplanır. Böylelikle eksik olan kısmın doldurulması için daha fazla önem biçilir. Çok katmanlı küresel derinlik parametreleri ilk olarak bu model ile denenmiştir fakat bu modelde insan etkileşimi gerekmektedir çünkü görselin eksik olan kısmını gösteren maske de modele girdi olarak beslenmelidir.

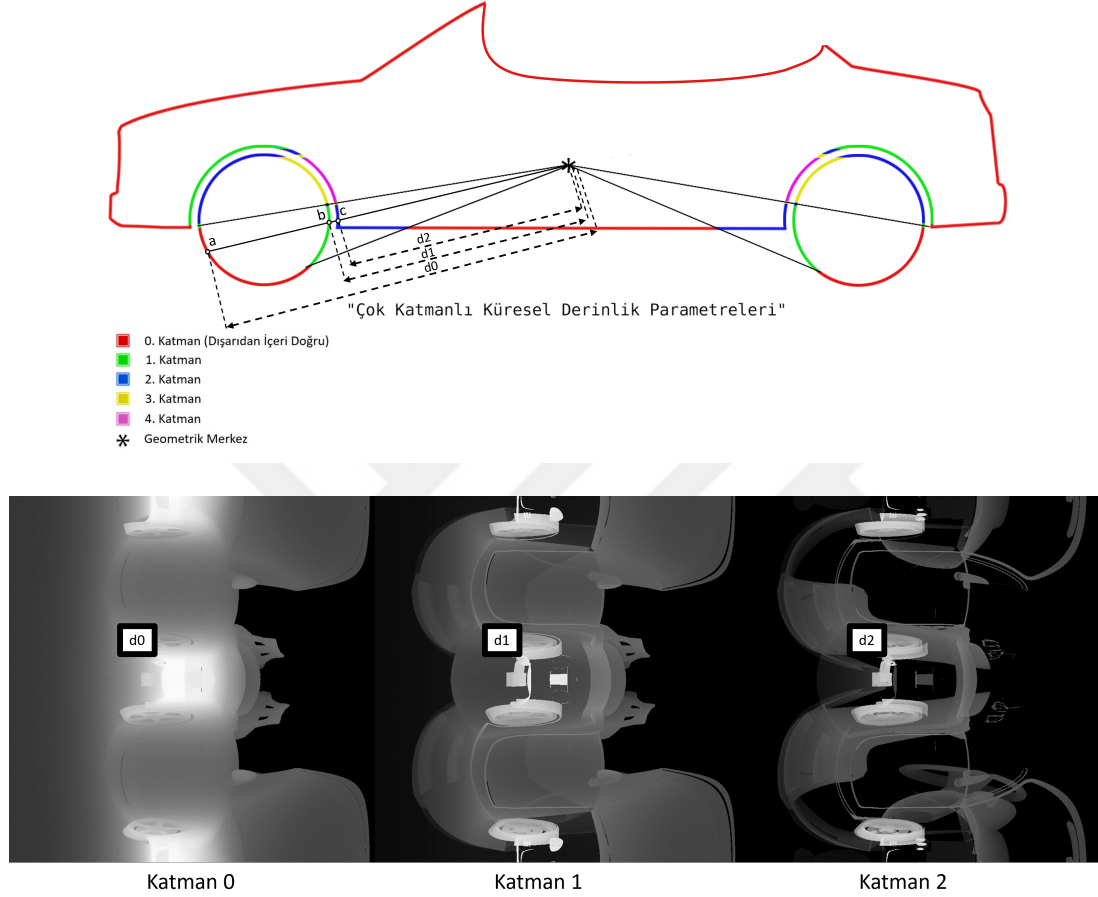
2) Bu sorunu aşmak için bu maskeyi tahmin eden ayrı bir model kullanılmıştır. Bu model, U-Net [6] tabanlı segmentasyon modelidir. U-Net'ten farklı olarak Kolla ve arkadaşlarının uyguladığı bir değişiklik de modelde mevcuttur [18]. Bu değişiklik, her aşağı ortaklama (max pooling) ve yukarı örnekleme (upsampling) işleminden önce yığın normalizasyonu (batch normalization) [19] katmanı kullanılmasıdır. Bu yöntem, verinin ağ boyunca öğrenilen normal değer aralığında seyretmesini sağlar. Segmentasyonda iki sınıf kullanılmıştır, birincisi şeklin eksik olduğu yerler, ikincisi ise şeklin dolu olduğu yerlerdir. Şeklin nerede dolu nerede boş olduğunu gösteren maske görseli de bir yapay sinir ağının buluşsal (heuristic) sonucu olduğu için bu sonuçta biraz gürültü olacaktır. Halihazırda gürültülü olan bir veri başka bir yapay sinir ağına beslendiğinde gürültü miktarı geometrik olarak artmış olacaktır. Bu yüzden bu deneme yüksek başarımla sonuçlanmamıştır.

3) Son çalışmada tek bir U-Net benzeri evrimsel sinir ağı ile maske kullanmadan regresyon yapılmış, iki farklı hiper-parametre ve iki farklı eksik nokta yüzdesi için sonuçlar raporlanmıştır. Raporlanan sonuçlar, 3B şekil tamamlama işini ustalık derecesinde yapan OCNN with Output Guided Skip Connections [14] çalışmasının eğitilmiş modeliyle kıyaslanmıştır. Bu kıyaslama için, önceki denemede kullanılmış olan veriler (iki farklı eksiklik yüzdesi ile), eğitilmiş olan OCNN with OGSC modeline beslenmiş, sonuçlar önceki denemeye aynı ölçüğe çekilmiş ve iki çalışma için de spesifik bir geometrik hata metriği kullanılarak orijinal şekiller ve tamamlanan şekiller arasındaki fark kıyaslanmıştır. Sonuçlar, çalışmamızın ustalık derecesindeki çalışmalar sınıfında olduğunu göstermiştir.

Bu çalışmaların tümünde derin öğrenme kütüphanesi olarak TensorFlow [28] kullanılmıştır. Çalışmada veri kümesi olarak [3] ve [7] çalışmalarından faydalanılmıştır.



### 3.1 Çok Katmanlı Küresel Derinlik Parametreleri

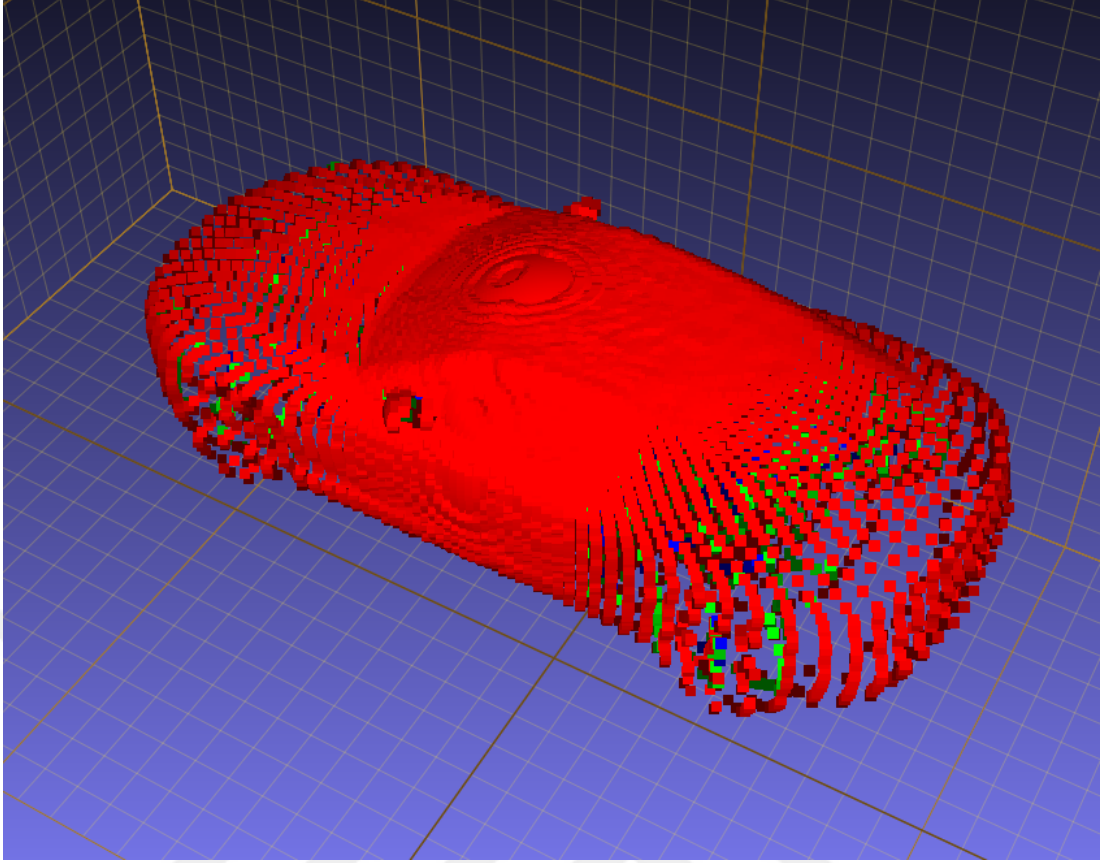


Şekil 3.1: Çok katmanlı küresel derinlik parametreleri çıkarımı ve üç katman için küresel derinlik görselleri.

[21] çalışmamızda kullandığımız şeklin değiştirilmiş versiyonudur. (© 2021 IEEE)

Çok katmanlı küresel derinlik parametrelerini elde etmek için 3B şeklin geometrik merkezinden dışarı doğru, ileri-geri ve yukarı-aşağı eksenleri etrafında 360 derece parçalı örneklenecek şekilde ışınlar gönderilir. Gönderilen ışınların her biri için, ilgili ışının kestiği noktalar geometrik merkeze olan uzaklıklarına göre sıralanır. Bu noktalar, farklı küresel derinlik görsellerine yazılır. En dışta kalan nokta 0 endeksli görsele yazılırken görsel endeksi dışarıdan içeri doğru ilerlerken artar. Aynı ışın üzerindeki tüm





Şekil 3.2: Çok katmanlı küresel derinlik parametrelerinin 3B Öklit uzayında karşılığı (katmanlara göre renklendirilmiş nokta bulutu). Görselleştirme MeshLab [2] ile elde edilmiştir.

noktalar, farklı geometri görsellerinin aynı  $(u, v)$  koordinatına yazılır.  $(u, v)$  koordinatı, 2B görsel üzerinde her pikselin görselin sol altına izafi olarak hangi konumda bulunduğunu gösterir. Alabileceği değer aralığı  $[0, 1]$ 'dir. Buna göre, atılan her bir ışının her iki eksenindeki açıları  $360'$ a bölünür, böylelikle her iki eksen için  $[0, 1]$  aralığında  $(u, v)$  koordinatı elde edilmiş olur. Aynı ışının kestiği tüm noktalar, yukarıda anlatılan yöntemle göre farklı küresel derinlik görsellerinin aynı  $(u, v)$  koordinatına yazılır. Derinliğin görselde tutulduğu değer aralığı, kullanılan veri türüne göre değişir. Virgüllü sayı kullanılması durumunda genelde  $[0, 1]$  aralığında derinlik bilgisi tutulurken tam sayı tutulması durumunda tam sayının  $n$  bit olması durumunda  $[0, 2^n]$  aralığında tutulur. Tüm bitlerin pozitif ekseninde kullanılması, derinliğin negatif olmayacağındandır. Her görsel formatı için piksel değerleri ilgili formata göre ölçeklendirilebilir, önemli olan bunun tutarlı bir biçimde yapılmasıdır. Bu varsayımla birlikte negatif olmayan

sayı şeklinde ilgili derinlik bilgisi tutulabilir. Şekil 3.1, 3.2 ve 3.3'te çok katmanlı küresel derinlik parametrelerinin nasıl elde edildiği anlatılmaktadır. Şekil 3.1'deki farklı renkler farklı katmanları simgelemektedir. Şekildeki a, b ve c noktaları, ilgili açıdan gönderilen ışının şeklin yüzeylerini kestiği noktalardır. Dışarıdan içeri doğru sıralandığında a 0. sırada yer alırken c 2. sırada yer alır. Bu sebeple, a noktasının derinlik değeri olan  $d_0$  0. küresel derinlik görseline yazılırken c noktasının derinlik değeri olan  $d_2$  2. küresel derinlik görseline yazılır. Her üç katman için de ilgili noktaların imajın hangi bölgesinde yer aldığı yine şekilde katman katman gösterilmiştir. 3B şekil, bu işlemden önce iki birimlik kapsayıcı küre içerisine koyulur. Bu da, merkezden en uzak noktanın merkeze bir birim uzakta olmasını sağlar.

```

ÇKKDPÖznelikÇıkarıcı(Şekil M, şekilÖlçeği, n $\theta$ , n $\phi$ , nKatman)
çıkıtıResimler  $\leftarrow$  [ ]
for k  $\in$  range(nKatman)
    çıkıtıResimler.ekle(yeni boş resim)
for i  $\in$  range(n $\theta$ )
    for j  $\in$  range(n $\phi$ )
         $\phi \leftarrow (j/n\phi) \times (2 \times \pi)$ 
         $\theta \leftarrow ((i/n\theta) \times 2 - 1) \times \pi$ 
        doğrultu  $\leftarrow$  normalizeEt(vec3( $\theta$ ,  $\phi$ ))
        ışın  $\leftarrow$  Işın(vec3(0), doğrultu)
        kesişimler  $\leftarrow$  ışın.kesitir(M)
        for k  $\in$  range(nKatman)
            u, v  $\leftarrow$  i/n $\theta$ , j/n $\phi$ 
            çıkıtıResimler[k][u][v]  $\leftarrow$  kesişimler[k]/şekilÖlçeği
return çıkıtıResimler

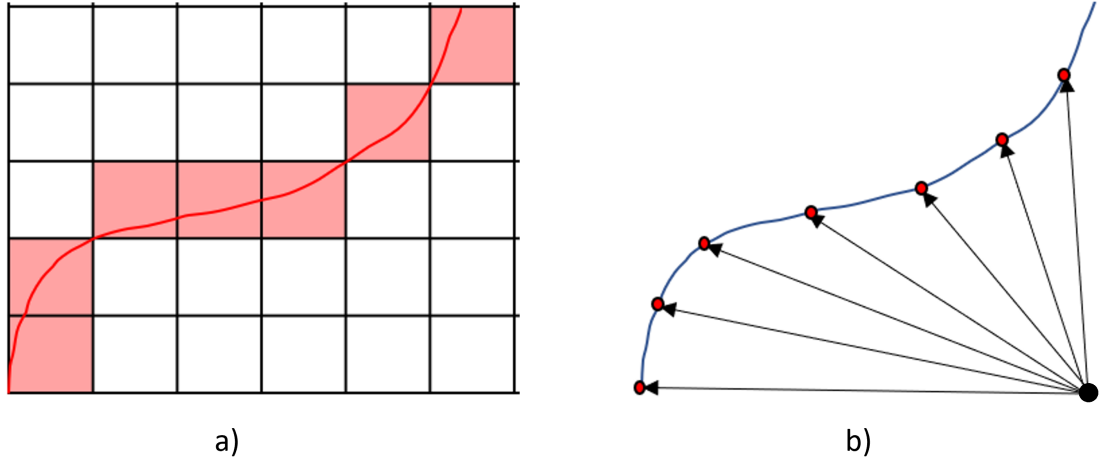
```

Şekil 3.3: ÇKKDP Öznelik Çıkarıcı Sözde Kod.

[21] çalışmamızda kullandığımız şekildir. (© 2021 IEEE)

Çıktı görseli sigmoid aktivasyondan ötürü [0, 1] aralığında tutulur. Girdi görseli ise [0, 255] değer aralığına ölçeklendirilir. Çok katmanlı küresel derinlik gösteriminin azami katman sayısı, veri kümesindeki en fazla değere göre belirlenebileceği gibi, önceden belirlenmiş ve şekillerin geometrik verisinin çok büyük bir kısmını temsil edebildiği tespit edilmiş sabit bir değer de olabilir. Bu şekilde tek katmanlı küresel derinlik parametrelerinin uygulamasındaki eksik kısımları tahmin etmek için kullanılan  $O(n^3)$ 'lük doluluk haritası tamamlayıcıları yerine, sabit sayıda katman ile  $O(n^2)$ 'lik karmaşa ile şeklin eksik kısmını tahmin etmek mümkündür.

Doluluk haritası temelli yaklaşımların  $O(n^3)$  zaman karmaşasına sahip olması dışında



Şekil 3.4: Doluluk Haritası - Küresel Derinlik Parametreleri Örnek Hassasiyeti Kıyaslaması

bir dezavantajı da özniteliklerin küplerle modellenmesinden kaynaklı yüzeyi sıkı bir şekilde sarmamasıdır. Doluluk haritasında küplerin yüzeye çok yakın bir hal alabilmesi için üç boyutta da çözünürlüğün ciddi miktarda artırılması gereklidir. Çok katmanlı küresel derinlik parametrelerinde ise çözünürlük ne olursa olsun, özniteliklerin yüzeye olan mesafesi virgüllü sayıların (genelde 32 bit olanların) hassasiyetinde olacaktır. Şekil 3.4 her iki yöntemle alınan örneklerin yüzeye yakınlık açısından hassasiyetini göstermektedir. Şekilde aynı şeklin hem daha az sayıda örnekle temsil edilmesi hem de alınan örneklerin tümünün yüzeye virgüllü sayı hassasiyetinde yakın olması gösterilmektedir. Zaten çok sayıda örnek toplamak zorunda olan doluluk haritası ise ancak sonsuz detaya bölündüğünde yüzeye sıfır mesafede olacaktır. Çözünürlüğü artırmak ise yüzeye dair detay sayısını artıracak, ve bu artırma işlemi yalnızca iki boyutta yapılacağından doluluk haritasındaki kadar verimsiz olmayacaktır. Sekizli ağaç yapısında ise zaman karmaşası çok katmanlı küresel derinlik parametreleri ile aynı olsa da, özniteliklerin peçlerle temsil ettiği yüzeylerin gerçek yüzeye yeterince yakın olması için ağacın derinliğini ciddi anlamda artırmak gerekecektir. Bu da çok katmanlı küresel derinlik parametrelerini diğer yöntemlere nazaran daha cazip bir hale getirir.

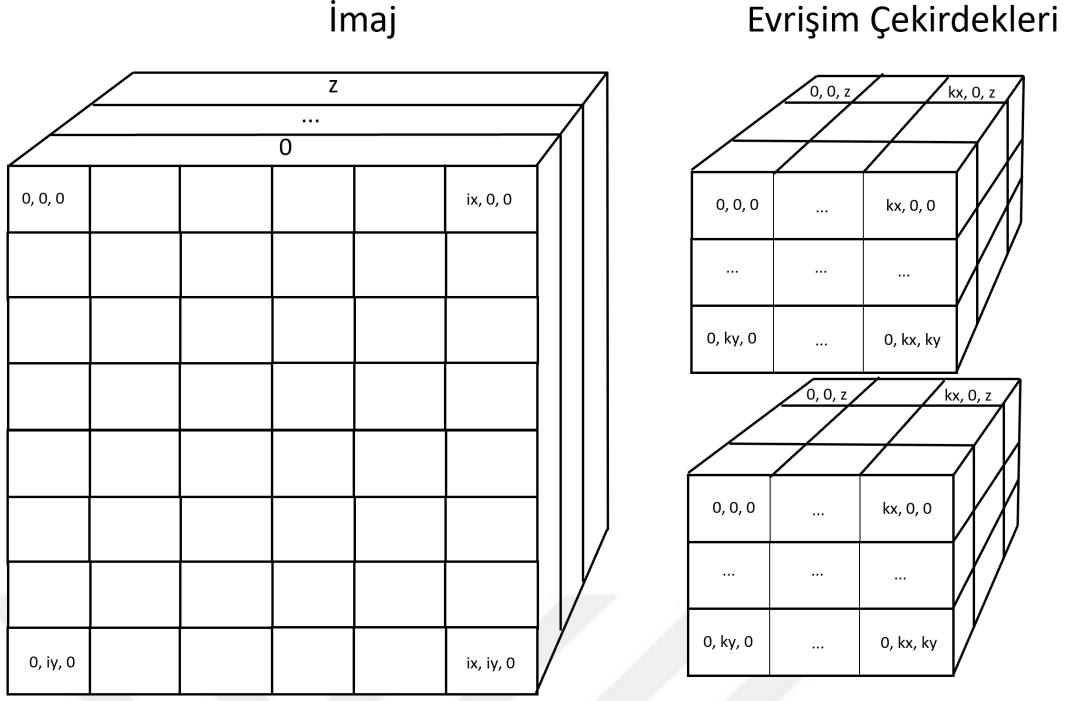
## 3.2 Evrişimsel Sinir Ağları, U-Net ve Kısmi Evrişimsel Sinir Ağları

### 3.2.1 Evrişimsel Sinir Ağları

Evrişim operasyonu, evrişim çekirdeği (convolution kernel) adı verilen sayı kümesinin ızgara yapıdaki veriye uygulanmasıyla sağlanır. Evrişim çekirdeği bir filtredir, bu filtre  $[x_k, y_k, z_k]$  boyutundadır. Bu filtre, bir pencere gibi  $[x_i, y_i, z_i]$  boyuttaki veri üzerinde gezdirilir. Çekirdek, merkezine tek bir veriyi (piksel, voksel veya daha çok boyutlu) alacak şekilde filtre sınırları içerisinde kalan bütün verilere uygulanır. Uygulama, filtre içerisindeki verinin görsel üzerinde tekabül ettiği veriyle çarpılması, filtredeki tüm sayılar için bu işlemin uygulanması ve çekirdeğin tek bir iterasyonunun sonucu olarak bu sonuçların toplanmasıdır.  $[x_k, y_k]$  boyutları, uygulanacak filtrenin ne denli kapsayıcı olacağı, tek seferde ne denli çok sayıda veriyi değerlendirmesi gerektiğine bağlı olarak operasyonun mimarisini belirleyen kimse tarafından belirlenir.  $z_k$  boyutu ise, evrişim operasyonunun üzerine uygulanacağı ızgara yapıdaki verinin  $z_i$  boyutu ile aynıdır. Bu senaryo, Şekil 3.5'te gösterilmiştir.

Filtrenin gezdirilirken kaç sıra atlayarak gideceği ve görselden taşan kısımlarda boşluğa gelen veri yerine ne koyacağı opsiyoneldir. Filtrenin bir kısmının boşluğa gelmesi durumunda, görsel sınırındaki değer tekrarlanabilir, sabit bir değer koyulabilir veya modüler bir biçimde görselin diğer ucundan veri okunabilir. Filtreyi görselden taşırılmayacak şekilde yerleştirmek de mümkündür fakat bu işlem, uygulama sonrasında görselin  $x_i$  ve  $y_i$  boyutlarını filtrenin  $x_k$  ve  $y_k$  boyutlarının yarısı kadar düşürür. Bu uygulanmaz da diğer üç yöntem uygulanırsa evrişim işlemi sonucunda  $[x_i, y_i, 1]$  boyutta yeni bir görsel elde edilir. Eğer  $n$  adet evrişim işlemi uygulanırsa, her bir işlemin sonucu  $z$  ekseninde arka arkaya eklenir ve  $[x_i, y_i, n]$  boyutunda bir çıktı görseli elde edilir. Şekil 3.6'da bu durum görselleştirilmiştir.

Evrişimsel sinir ağları, evrişim operasyonlarının filtre değerlerini rastgele veya öğrenilmesi kolay bir durumdan başlatarak ileri besleme ve geri dağılım yöntemleriyle optimal değerlere yakınsamasını sağlayan, bu öğrenilen parametreleri sınıflandırma veya sürekli değer tahmini gibi işlerde kullanabilen bir yapay sinir ağı modelidir. Evrişimsel

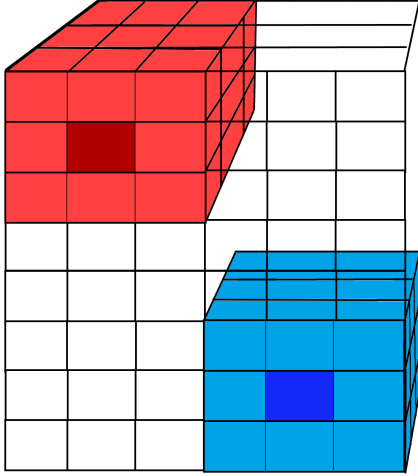


Şekil 3.5: Yukarıdaki şekilde  $[ix, iy, z]$  boyutlarında bir görsel ve  $[kx, ky, z]$  boyutlarında iki adet evrişim filtresi bulunmaktadır. Z eksenindeki örnekleme sayısının aynı olması önemlidir.

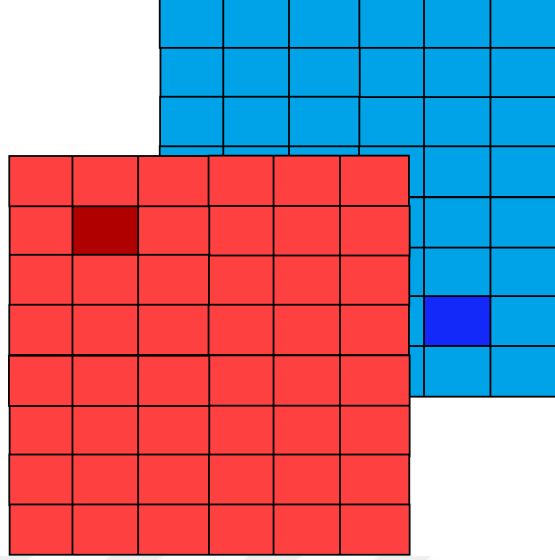
sinir ağları genelde derin yapıda olup çok sayıda katmandan oluşur. Evrişimsel sinir ağı modellerinde genelde yığın normalizasyonu (batch normalization), güç sönümü (dropout), azami ortaklama (max pooling), yukarı örnekleme (up-sampling) ve birleştirme (concatenation) katmanları da evrişim katmanlarıyla birlikte kullanılır. Bu katmanların bazıları öğrenilebilir parametreler içerirken bazıları yalnızca veriyi organize etme ve verinin önemli kısımlarının öne çıkarılmasını sağlamada görevlidir.

Evrışimsel sinir ağlarında eğitim ardışık iki basamaktan oluşur; ileri besleme (feeding forward) ve geri dağılım (back propagation). İleri besleme safhasında evrişim işlemi her bir katmandan geçmekte olan girdi görseline uygulanır. Her bir katmanın çıktısı, diğer bir veya birden fazla katman (birleştirme katmanı kullanılması durumunda) için girdi mahiyetindedir. Sinir ağının son katmanı da çalışıp önceden belirlenmiş olan bir masraf fonksiyonuna bağlı olarak (örn. ortalama karesel hata) mevcut masraf değeri hesaplandıktan sonra ilgili masraf değerinin çıktıya göre birinci dereceden türevi alınıp her piksel için azalma doğrultusu bulunur. Bu doğrultu, masraf fonksiyonu üzerindeki

Evrişim İşlemi Örnekleri

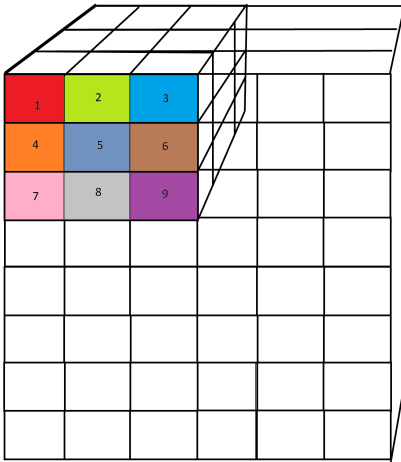


Çıktı Katmanları

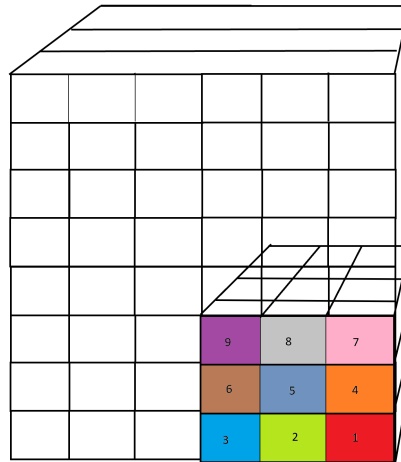


Şekil 3.6: Yukarıdaki şekilde  $[ix, iy, z]$  boyutlarında bir görsel üzerine  $[kx, ky, z]$  boyutlarında iki adet evrişim filtresi uygulandıktan sonra ortaya çıkan şeklin hangi pikselinin hangi evrişim operasyonundan geldiği gösterilmektedir. Kırmızı çıktı kanalı kırmızı filtrenin sonucu olup mavi çıktı kanalı mavi filtrenin sonucudur. Koyu kırmızı ile gösterilen piksel kırmızı filtrenin ilgili andaki operasyonunun sonucunun 1. kanalın hangi pikseline yazıldığını gösterirken koyu mavi ile gösterilen piksel mavi filtrenin ilgili andaki operasyonunun sonucunun 2. kanalın hangi pikseline yazıldığını göstermektedir.

İleri Besleme



Geri Dağılım



Şekil 3.7: İleri besleme ve geri dağılım evrişim işlemleri

en yakın yerel çukur noktayı gösteren bir teğet vektördür. Bu vektör doğrultusunda ilerlendiği takdirde masraf, ilgili örnek için azaltılmış olacaktır.

Grafik üzerinde sekmelere sebebiyet vermemek için düşüm (gradient) vektörü, öğrenme oranı adı verilen sabit veya değişken bir değere bağlı olarak aşağı ölçeklendirilecektir. Daha sonra her piksel için bulunan azalma doğrultusu çıktı katmanından girdi katmanına doğru, ileri besleme işleminin tersi yönde ilerleyecektir. Bir katmanın dereceli azalma görseli, diğeri için masraf değeri anlamında olacaktır. Aynı işlem girdi katmanına kadar devam edecektir. Bu işlem her görsel için tek tek veya görsel yığını için tek seferde yapılabilir. Tek tek yapılması durumunda daha düşük masraf değerlerine yakınsaması daha olasıyken görsel yığını için yapıldığında yığının görsel adedine bağlı olarak işlemin hız anlamında daha yüksek performansta yapılması olasıdır. Her bir görsel veya yığın için biraz daha optimal değere yakınsanan masraf değeri, çok sayıda iterasyon sonucunda sezgisel olarak düşük bir değere yakınsayacaktır. Sezgisel yaklaşımlarda bulunan değerlerin mutlak minimum olduğunu bilmek mümkün değildir. Farklı parametrelerle mümkün olan en ufak değere yakınsamak en önemli hedeftir.

Eğitim esnasında eğitim verisinin yanısıra başarı oranı eğitimde kullanılmayan validasyon verisi ile de test edilir. Eğitimin seyri açısından en objektif veri de ağı eğitim esnasında validasyon verisi üzerindeki başarıdır. Eğitim bittikten sonra yine eğitim esnasında görülmemiş olan test verisi ile yapılacak olan test, eğitimin başarısıyla ilgili nihai sonucu verir. Bu verilerin dağılımı farklılık gösterebilmekle birlikte genelde 60% eğitim, 20% validasyon ve 20% test verisi şeklindedir. Yüzdeler bütün veri kümesinin boyutuna oranlardır.

Evrişimsel sinir ağlarında her bir evrişim katmanında  $[z_c, k_x, k_y, z_g]$  boyutlarında bir evrişim çekirdekleri matrisi bulunur.  $z_c$  hedeflenen çıktı filtre sayısıdır,  $z_g$  ise beklenen girdi filtre sayısıdır.  $k_x$  ve  $k_y$  ise uygulanacak olan her bir evrişim çekirdeğinin x ve y boyutlarıdır. İleri besleme işlemi evrişim çekirdeklerinin girdi görseli üzerinde gezdirilmesiyle yapılırken, geri dağılım işlemi evrişim çekirdeklerinin 180 derece döndürülmüş halinin katman üzerinden geçmekte olan dereceli azalma görseli üzerine uygulanmasıyla yapılır. Bu uygulama, Şekil 3.7'de görselleştirilmiştir. Geri dağılımda girdinin katmanının aktivasyon fonksiyonunun türevine beslendiğinde elde edilen değer, 180 derece döndürülmüş evrişim çekirdeklerinin dereceli azalma görseli üzerine uygulanma-

sından sonra bulunan sonuçla çarpılır. Nihayetinde katman, kendisine beslenmiş geri dağılım değerinin türevini alıp bulduğu değeri sonraki katmana aktarmış olur. Geri dağılmakta olan dereceli azalma görseli ve girdi görselinin evrişim çekirdeği girdi görseli üzerinde gezdirilirken ilgili doğrultuda çarpılmasıyla da evrişim çekirdeklerinin dereceli azalma değerleri bulunur. Bu değer de yine öğrenme oranıyla çarpılarak evrişim çekirdeğinden çıkarılır, böylece evrişim çekirdeğinin sezgisel değeri de güncellenmiş olur.

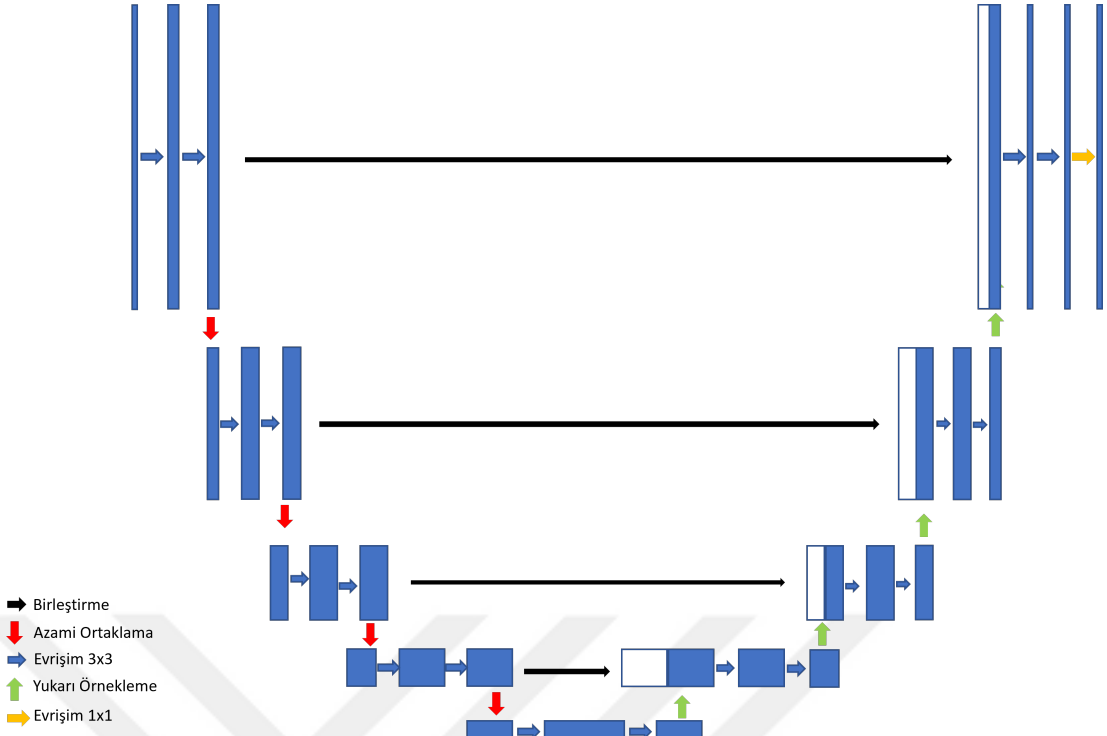
### 3.2.2 U-Net

U-Net [6], özel bir tür evrişimsel sinir ağıdır. Esasında tıbbi segmentasyon amaçlı kullanım için tasarlanmıştır fakat öğrenme kabiliyeti çok yüksek olduğundan farklı varyasyonları farklı amaçlarla da kullanılmıştır. Darboğaz yapılı bir evrişimsel sinir ağıdır. Darboğaz adı verilen, görselin  $x$  ve  $y$  boyutlarının minimum  $z$  boyutunun ise maksimum olduğu bir evre içerir. Darboğaza kadar bir dizi kodlama, yani  $x$  ve  $y$  boyutunu azaltırken  $z$  boyutunu artıran evrişim operasyonları uygulanır. Darboğazdan sonra ise  $x$  ve  $y$  boyutlarını artırırken  $z$  boyutunu azaltan deşifre işlemleri uygulanır. Kodlama ve deşifre katmanları arasında atlama bağlantıları bulunur, veri sadece bir boru hattı üzerinden gitmez, bir grafik gibi ardışık olmayan katmanlar arasında da bağ bulunur. Şematik gösterimi "U" harfine benzediği için adına UNET denmiştir. Biyomedikal segmentasyon amaçlı kullanılan U-Net piksel bazında sınıflandırma görevi için kullanılıyordu. Son katmanının aktivasyon fonksiyonu ve modelin masraf fonksiyonu değiştirilerek bu mimari aynı zamanda sürekli veri tahmin işleri için de kullanılabilir. Şekil 3.8'de orijinal U-Net'in veri akış diyagramı gösterilmiştir.

### 3.2.3 Kısmi Evrişimsel Sinir Ağları (Partial Convolutional Neural Networks)

Kısmi evrişimsel sinir ağları, evrişimsel sinir ağlarının özel bir türüdür. Liu ve arkadaşları [22] ilgili çalışmada U-Net tipi bir yapay sinir ağının evrişimsel katmanlarını kendi geliştirdikleri kısmi evrişimsel katmanlarla değiştirmişlerdir.





Şekil 3.8: U-Net

Bu özel türden evrişimsel katman, girdi olarak yalnızca işlenecek olan görseli değil, bunun yanısıra tahmin edilecek olan kısımların maskesini de alır. Sinir ağı böyle bir katmanla başladığı için girdi olarak görsel ve maske verileri verilir. İleri besleme basamağında kısmi evrişimsel sinir ağı ileri doğru hem görseli besler, evrişim operasyonuna tâbi tutar, hem de maskeyi evrişim işlemi uygulandıktan sonraki şekline uygun bir şekilde getirir. Katman, çıktı olarak evrişime tâbi tutulmuş ve yeniden şekillendirilmiş maskeyi verir. Bu işlem son katmana kadar devam eder. Bunlardan maske çekirdekleri öğrenilebilir parametreler değilken görselin evrişim çekirdekleri öğrenilebilir parametrelerdir. Maske, her kısmi evrişim katmanında görsel ile çarpılıp görselin yalnızca anlamlı kısmının evrişimden faydalanmasını sağlar. Burada dikkat edilmesi gereken, evrişim işleminin yine tüm görsel uygulanıyor oluşudur, yalnızca maskenin anlamlı olmadığını söylediği yerlerin değeri sabit sıfırdır. İşlem sayısı cinsinden bir optimizasyon söz konusu değildir.

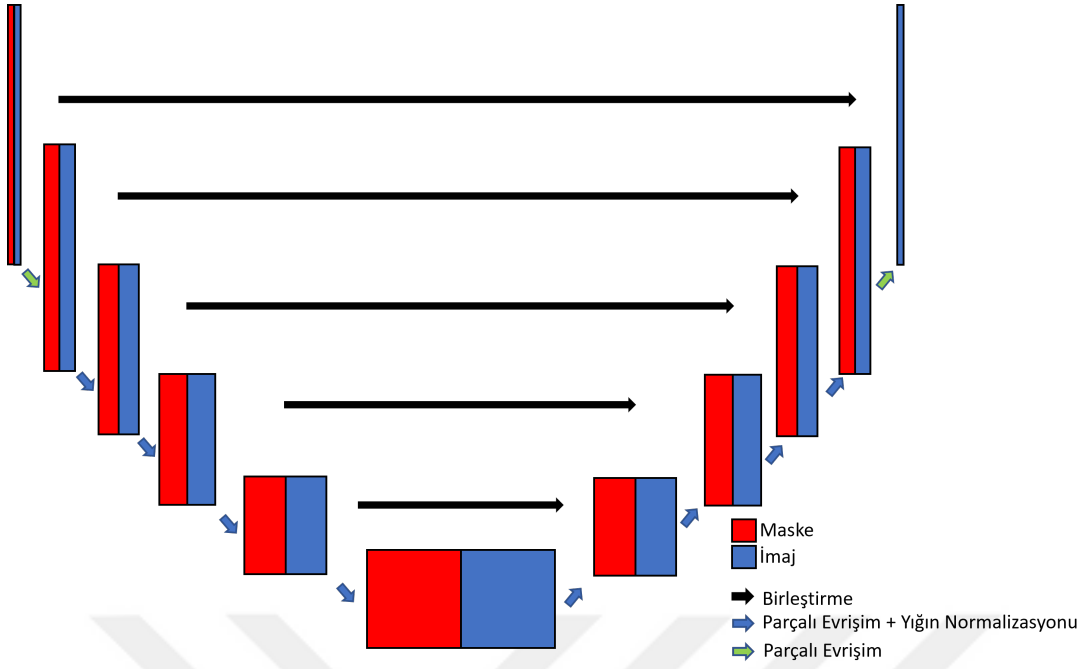
Hata hesaplama safhasında son katmana kadar getirilmiş ve çıktı görselinin şeklini almış maske, çıktı katmanına belirli katsayılarla uygulanır ve farklı bileşenlerle hata

$$\begin{aligned}
L_{valid} &= \frac{1}{N_{I_{gt}}} \|M \times (I_{out} - I_{gr})\|_1 \\
L_{hole} &= \frac{1}{N_{I_{gt}}} \|(1 - M) \times (I_{out} - I_{gr})\|_1 \\
L_{perceptual} &= \sum_{p=0}^{p-1} \frac{\|\psi_p^{I_{out}} - \psi_p^{I_{gt}}\|_1}{\psi_p^{I_{gt}}} + \sum_{p=0}^{p-1} \frac{\|\psi_p^{I_{comp}} - \psi_p^{I_{gt}}\|_1}{\psi_p^{I_{gt}}} \\
L_{style_{out}} &= \sum_{p=0}^{p-1} \frac{1}{C_p C_p} \|K_p((\psi_p^{I_{out}})^T (\psi_p^{I_{out}}) - (\psi_p^{I_{gt}})(\psi_p^{I_{gt}}))\|_1 \\
L_{style_{comp}} &= \sum_{p=0}^{p-1} \frac{1}{C_p C_p} \|K_p((\psi_p^{I_{comp}})^T (\psi_p^{I_{comp}}) - (\psi_p^{I_{gt}})(\psi_p^{I_{gt}}))\|_1 \\
L_{tv} &= \sum_{(i,j) \in R, (i,j+1) \in R} \frac{\|I_{comp}^{i,j+1} - I_{comp}^{ij}\|_1}{N_{I_{comp}}} + \sum_{(i,j) \in R, (i+1,j) \in R} \frac{\|I_{comp}^{i+1,j} - I_{comp}^{ij}\|_1}{N_{I_{comp}}} \\
L_{total} &= L_{valid} + 6L_{hole} + 0.05L_{perceptual} + 120(L_{style_{out}} + L_{style_{comp}}) + 0.1L_{tv}
\end{aligned} \tag{3.1}$$

Şekil 3.9: [22] çalışmasında kullanılan masraf bileşenleri.

hesabı yapılır. Bileşenlerin bir kısmı kısmi evrişimsel sinir ağının çıktıları üzerinden hesaplanırsa da, bir kısmı VGG [23] sinir ağının ilk birkaç basamağını öznetelik çıkarım yöntemi olarak kullanır ve orijinal veri ile tahmin edilmiş verinin bu birkaç katman için verdiği çıktının masrafını kullanır. Tüm masraf bileşenleri, 3.1’de gösterilmiştir [22]. Şekilde  $I_{in}$  girdi görselini,  $M$  maskeyi,  $I_{out}$  çıktı görselini,  $I_{gt}$  taban doğru görseli temsil eder.  $I_{comp}$  maskelenmemiş pikselleri taban doğruya eşitlenmiş  $I_{out}$  görselini temsil eder.  $N_{\psi^{I_{gt}}}$  ise  $\psi_p^{I_{gt}}$ ’deki eleman sayısını gösterir. "Perceptual Loss" yani sezgisel masraf  $I_{out}$  ve  $I_{comp}$  ve taban doğru görsellerinin daha yüksek bir öznetelik uzayında aralarındaki farkı temsil eder, yani VGG-16 ile eğitilmiş ImageNet’teki masrafı temsil eder [23] [22].  $L_{tv}$  ise toplam varyans masrafıdır, delik bölgenin 1 piksel açığından atılmış ceza yumuşatmasıdır [tv loss referansı] [partial convolution referansı]. Katsayılar ilgili çalışmalardan elde edilmiş hiper-parametrelerdir. Şekildeki  $L_{valid}$  masraf hesabı, çıktı görselinin girdideki eksiltilmemiş koordinatlarının tahmin masrafıdır.  $L_{hole}$  çıktı görselinin girdideki eksiltilmiş kısımlarının tahmin masrafıdır.  $L_{perceptual}$ , Gatys ve arkadaşlarının [24] çalışmalarında kullandıkları bir masraf hesap yöntemidir. VGG’ye beslenen ve ilk birkaç katmandan sonra çıktısı alınan taban doğru (ground truth) ve tahmin görsellerin bu birkaç katmanın sonundaki çıktıların tâbi tutulduğu masraf fonksiyonudur.  $L_{style}$  ise yine VGG’nin ilk birkaç katmanından sonra tahmin ve taban doğru görseller için verdiği çıktıya göre hesaplanır. Bu masraf fonksiyonu ise yine aynı çalışmadandır.  $L_{style_{out}}$  ham tahmin için hesaplanır.  $L_{style_{comp}}$  ise tahminin maskeli kısımlarının tahminden, maskesiz kısımlarının taban doğru görselden alındığı tahmin görseline göre hesaplanır. Son olarak  $L_{tv}$ , toplam varyans masrafıdır. Bu masraf fonksiyonu, eksik kısımların sınırlarını yumuşatmak içindir. Eksik kısımlar görselden keskin çizgilerle ayrıldığı için, tahmin edilen eksik kısım ile eksil olmayan kısım arasında bir sınır çizgisi (seam) görünme ihtimali yüksektir. Bunu azaltmak için toplam varyans masrafı uygulanır ve eksik kalan kısımların biraz daha genişletilmiş, yani deliklerin sınırlarının daha ileriye çekilmiş halina masraf uygulanır. Tüm bu masraf fonksiyonlarının deneme yanılma yoluyla elde edilen katsayılarla ağırlıklı toplamı alınca toplam masraf ortaya çıkar.

Kısmi evrişimsel sinir ağlarının akış diyagramını Şekil 3.10’da görebilirsiniz. Temelde

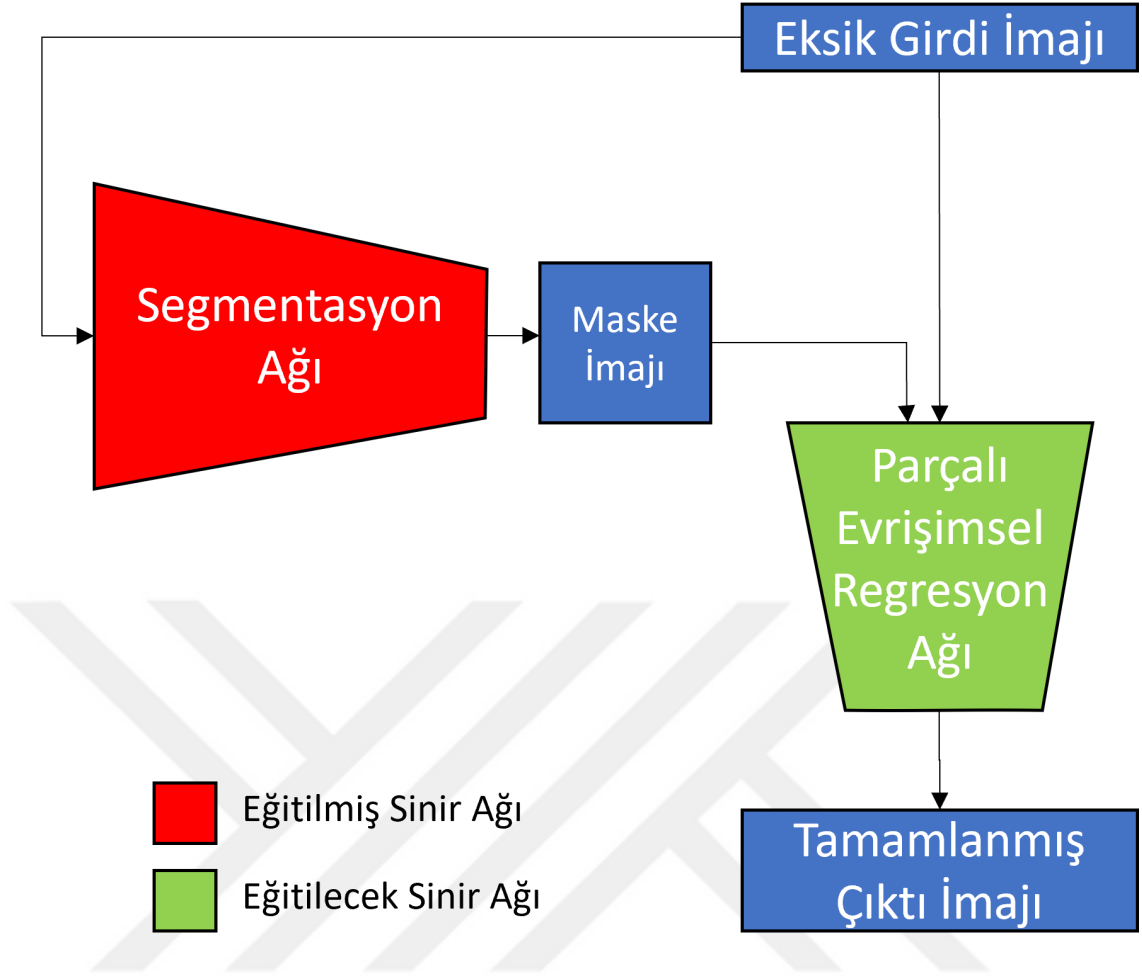


Şekil 3.10: Kısmi Evrişimsel Sinir Ağları [22]

U-Net ile aynı akış diyagramına sahip olduğu söylenebilir. Tek farkı, evrişim katmanının değiştirilmiş olması ve masraf hesaplamasının hibrit bir şekilde birden fazla yöntemle dayanılarak yapılmasıdır, yani standart bir masraf fonksiyonu kullanılmamasıdır.

### 3.3 Kısmi Evrişimsel Sinir Ağı ve U-Net Tabanlı Maske Tahmin Edici ile Çok Katmanlı Küresel Derinlik Parametreleri Tamamlama

Bu çalışmada iki farklı yapay sinir ağı tek bir ağ altında birlikte kullanılmıştır. Bunlardan bir tanesi önceden eğitilmiştir ve diğer sinir ağına eğitim esnasında çıktısını veri olarak besleyebilecek hale getirilmiştir. Diğer sinir ağı ise hem girdi görselini, hem de bu görsel üzerinde eksik olan noktaların konumunu belirten maske görselini girdi olarak alır. Maske görseli elle verilmez, ilk sinir ağı bu maskeyi tahmin eder ve ikinci sinir ağına besler. İlk sinir ağı segmentasyon ağı iken ikincisi regresyon ağıdır. Şekil 3.11 ilgili modelin akış diyagramını göstermektedir.



Şekil 3.11: Kısmi Evrimsel Tamamlayıcı Ağ Kompleksi

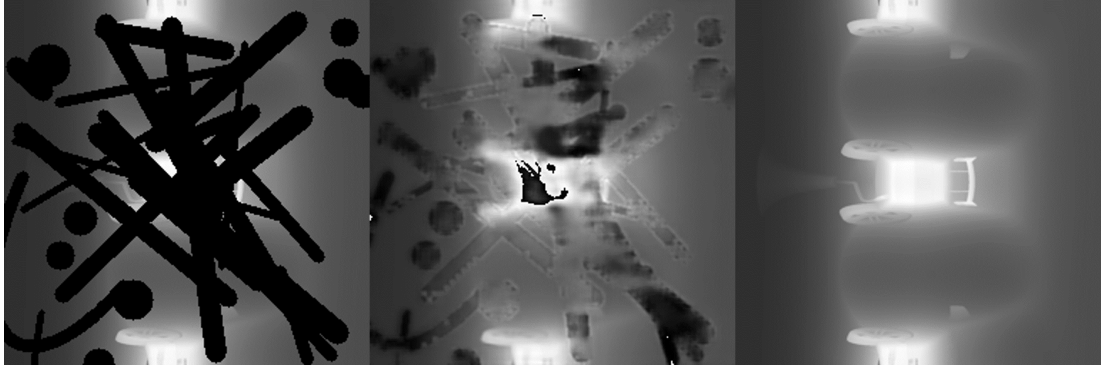
### 3.3.1 Eğitim Verisi Hazırlama

Bu deneyde eğitim verisi olarak üç katmanlı küresel derinlik parametreleri kullanılmıştır. Üç Katmandan rastgele noktalar silinmiş, öncelikle ağın performansını ölçmek adına katmanlar arası sıralama ve yeniden organizasyon yapılmamıştır. Bunun nedeni, maske tahmin edicinin performansının ne denli yeterli olacağını gözlemlemek istememizdir.

### 3.3.2 Sinir Ağı Modeli

U-Net tabanlı bir segmentasyon ağı, küresel derinlik parametrelerinin eksik kısımlarını tahmin edecek şekilde eğitilmiştir. Sinir ağının çıktı katmanı softmax aktivasyon fonksiyonuna ve categorical cross entropy masraf fonksiyonuna sahiptir. Bunun sebebi, her bir piksel için ilgili pikselin küresel derinlik parametresinin yüzeye tekabül edip etmediğini sınıflandırmak istememizdir. İki sınıf yerine tek parametreyle de temsil edilebilir, binary cross entropy masraf fonksiyonu da kullanılabilirdi. Bu durumda da aşağı yukarı aynı performansı verecektir, toplam parametre sayısı diğerine nazaran biraz daha az olacaktır. Sinir ağında Adam Optimizer kullanılmış, öğrenme oranı  $10^{-4}$ 'te tutulmuştur. Ağ yapısı 31 milyon parametrelilik standart bir U-Net'tir. U-Net'te Kolla ve arkadaşlarının Eye In The Sky [18] çalışmasında kullandıkları yığın normalizasyon katmanları her aşağı ortaklama ve yukarı örnekleme katmanlarından önce kullanılmıştır.

Yine U-Net şeklinde fakat evrişim yerine kısmi evrişim kullanan, masraf fonksiyonu da yine 3.1 formülündeki gibi olan bir sinir ağı da regresyon yapmak üzere eğitilmiştir. Bu sinir ağının görevi, kendisine verilen eksik resmin eksik kısımlarını, yine kendisine prosedürel olarak sağlanan maske görselini kullanarak tahmin etmesidir. Kısmi evrişimsel sinir ağının toplam parametre sayısı 18 milyon civarındadır. Toplamda altı adet kodlama, bir adet darboğaz ve altı adet de deşifre katman kompleksi kullanılmıştır. Kodlama kompleksinde kısmi evrişim katmanı ve ilk katman haricinde yığın normalizasyon katmanı kullanılmıştır. Kullanılan aktivasyon fonksiyonu ReLu'dur. Bu fonksiyonun tercih edilmesi, oldukça derin olan bu sinir ağının gradyanlarının geri dağıtılırken başlangıç katmanlarına doğru sıfıra yakınsamasını önlemek içindir. Deşifre kompleksinde ise kısmi evrişim ve son katman hariç yığın normalizasyonu katmanı kullanılmıştır. Aktivasyon fonksiyonu olarak yine gradyanların sıfıra yakınsamasını önleme amaçlı Leaky ReLu aktivasyon fonksiyonu kullanılmıştır. Leaky ReLu'nun ReLu'dan farkı, negatif toplam değerleri için aktivasyon fonksiyonu 0 dönmez, fakat doğrunun eğimi pozitif değerlerinkine nazaran, ki pozitif doğrunun eğimi birdir, çok daha azdır. Genelde 0.1 tercih edilir. Modelde Adam Optimizer kullanılmıştır, öğrenme



Şekil 3.12: Kısmi Evrişim Deneyi: Eksik, Tamamlanmış ve Taban Doğru (Ground Truth) İmajlar



Şekil 3.13: Kısmi Evrişim Deneyi: Tahmin Edilmiş ve Taban Doğru Maskeler

oranı  $2 \times 10^{-4}$  olarak belirlenmiştir.

Bu iki ağ Şekil 3.11'deki gibi birlikte kullanılır. Çok katmanlı küresel derinlik parametreleri üç katman olarak eksiltilmiş geometri ile sinir ağına beslenir, sinir ağı çıktı olarak tamamlanmış görseli verir. Üç görsel de ağa ayrı ayrı beslenir. Deneyin sonunda çıktı görselleri görsel açıdan değerlendirilir, görsellerde görsel bozukluklar olup olmadığı görülür.

### 3.3.3 Deney

Deneyde önce segmentasyon ağı eğitilmiş, eğitilmiş segmentasyon ağı kullanılarak kısmi evrişimsel sinir ağı da eğitilmiştir. Eğitilmiş sinir ağının ortalama karesel hata değerleri hesaplanmış, ve çıktı görselleri görsel olarak incelenmiştir. Görsel görsellerdeki kusurlar nedeniyle sonuçlar doğrudan kullanılmamış, diğer deneylere geçilmiştir. Eksik görsel, tamamlanmış görsel ve taban doğru görsellerin örneği Şekil 3.12’de mevcuttur. Taban doğru maske ve tahmin edilmiş maskenin örneği de Şekil 3.13’te mevcuttur. Bu sonuçlar ışığında maskeyi tahmin edip kısmi evrişim uygulamak yerine tek bir U-Net tabanlı sinir ağı modeli ile, küresel derinlik resminin her bir pikselinin regresyon yapılarak tahmin edilmesi daha uygun görülmüştür.

### 3.4 U-Net Tabanlı Sinir Ağıyla Çok Katmanlı Küresel Derinlik Parametreleri Tamamlama

Bu çalışmada kısmi evrişimsel sinir ağı ile yapılan çalışmadan farklı olarak herhangi bir maskeleme işlemi kullanılmamış, doğrudan tek bir masraf fonksiyonu ile ve tek bir çıktı ile 2B bir görsel üzerinden piksel bazında regresyon yapılmıştır. Maskeleme işlemine gerek olmadığı için maske tahmin etmede kullanılan segmentasyon ağı da bu çalışmada yer almamaktadır. Bu durum, bu çalışmayı öncesine nazaran daha sade ve yüksek performanslı (lightweight) yapmaktadır.

#### 3.4.1 Eğitim Verisi Hazırlama

Eksik şekil tamamlama işlemini öğrenebilmek için ilgili 3B şekillerin hem eksiltilmiş hem de taban doğru hallerine ihtiyaç duyulur. Eksiltilmiş şekil tutarlı ve gerçeğe uygun olmalıdır. Hazırlanacak eksik verilerin gerçekçi senaryolar olması önem arz etmektedir. Üçgenleştirilmiş 3B şekil üzerinden veri eksiltmek için poligon bölme ve yeniden oluşturma işlemleri yapılması gerekir, yüzey parçalı bir uzay olarak üçgenlerden oluşturulduğu için yüzeye sürekli ve rastgele boşluklar açmak zor bir görevdir. Yapay sinir

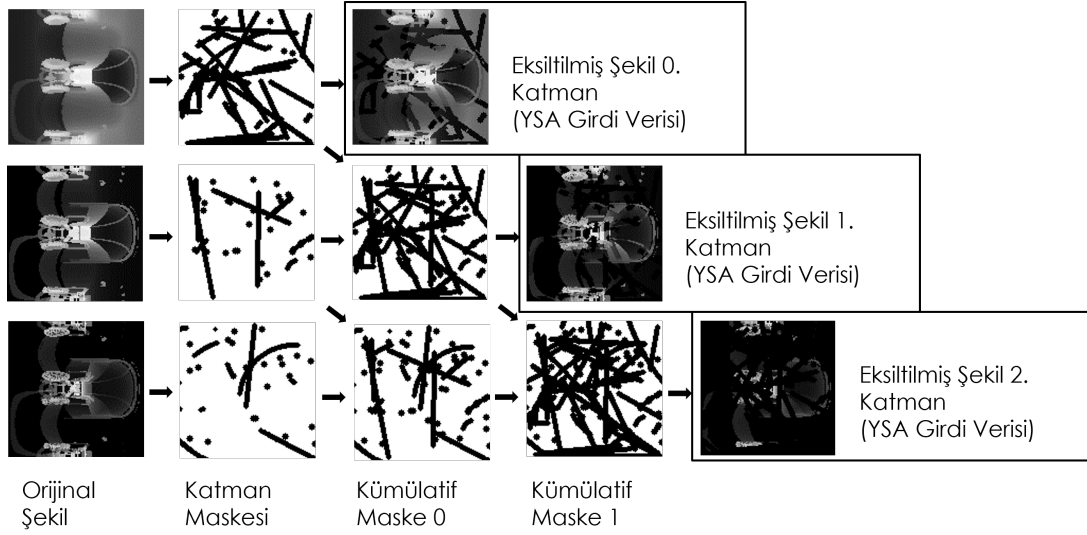


ağı çok katmanlı küresel derinlik parametreleri ile çalışacağı için veri eksiltme işleminin küresel derinlik uzayında yapılmasında sakınca yoktur, neticede eksik veri de bu uzaya dönüştürüldüğünde aynı derinlik görseline sahip olacaktır. Fakat veri eksiltirken dikkat edilmesi gereken bir nokta vardır. Küresel derinlik parametreleri birden fazla katmanda ele alındığında ilgili noktaların derinlik görsellerine dağıtımı geometrik şeklin merkezine olan mesafeye göre ters sıralıdır. En uzak noktalar ilk görselde yer alırken en yakın olanlar son görselde yer alır. Aradan bazı noktaları sildiğimizde ilgili açı için noktaların sıralaması değişir ve bu noktaları yeniden sıralamak gerekir. Dolayısıyla, eğer en dış katmandan bir nokta siliyorsak, bir alt katmandaki ilgili açıya tekabül eden noktanın artık en üst katmanda değerlendirilmesi gerekir. Eğer bu açıya tekabül eden nokta yoksa sırayla diğer alt katmanlara bakılır. En başta da belirtildiği üzere sentetik verinin tutarlı olması için bu önemlidir. Üst katmana aktarılan verinin de eski katmanından silinmesi gerekir. Bu amaçla, her bir katmandan rastgele noktalar silindikten sonra dıştan içe doğru noktaları görsellere tekrar dağıtmak gerekir. Rastgele nokta silinmesi işlemi de rastgele ebatlarda oval geometrik şekillerin 2B görsel üzerinde delikler açacak şekilde kullanılmasıyla olur. Önce delikler açılır, sonra kümülatif maskeleyme işlemi yapılır. Şekil 3.14'te bu işlemin örneği görülmektedir.

Bu çalışmada çok katmanlı küresel derinlik parametrelerinin bir tür derin öğrenme modeli olan U-Net'e beslenerek eksik girdi şeklinin tamamlanması ele alınmıştır. Bunun için öncelikle 3B şekillerin 3B Öklit uzayından küresel uzaya dönüşümü yapılmış, ardından küresel uzayda bu verilere gerçeğe benzer delikler açılmıştır. Oluşturulan sentetik eksik veri yapay sinir ağının eğitiminde kullanılmış, nihayetinde eksik küresel derinlik görseli tamamlayabilen bir yapay sinir ağı elde edilmiştir. Bu sinir ağı, ilgili eğitim verisiyle eğitilmiş ve ağın test aşamasında çıktığı görselleri 3B Öklit uzayına dönüştürülmüş, taban doğru şekiller ile kıyaslanmış ve başarı oranı raporlanmıştır.

### **3.4.2 Sinir Ağı Modeli**

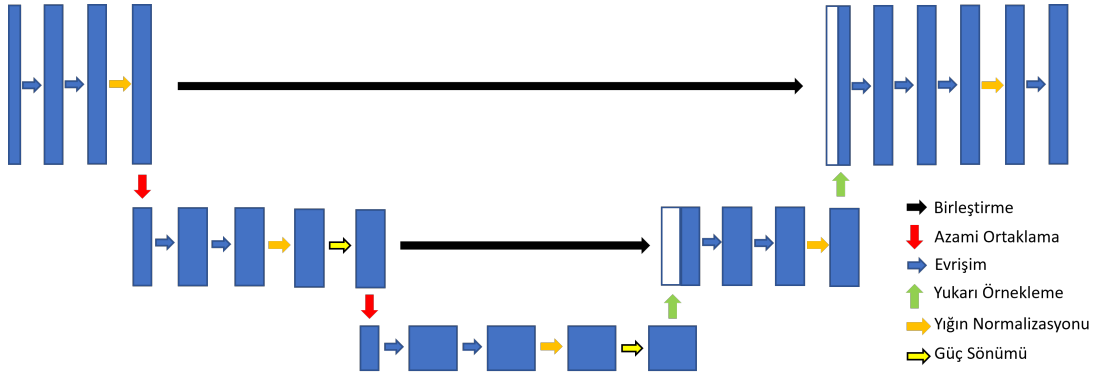
Kullandığımız sinir ağı, katman sayısı azaltılmış ve biraz da modifiye edilmiş bir U-Net. U-Net'ten farklı olarak da Kolla ve arkadaşlarının çalışmalarında [18] yaptığı gibi her



Şekil 3.14: Kümülatif Maskeleme İşlemi: Bu işlem, eğitim için gerekli eksik geometri bilgisini tutarlı bir şekilde hazırlamak için uygulanır. Öncelikle her bir katmandan ayrı ayrı veri silme işlemi gerçekleşir. Fakat, en dıştaki katmandan bir nokta silindiği zaman, bir alt katmanda aynı doğrultuda yani aynı  $(u, v)$  koordinatında bir nokta varsa, bu alt katmandaki nokta artık en dıştaki nokta olacağından onu alıp bir üst imaja yazmak gerekir. Bir alt katmanda yoksa şayet, iki alt katmana bakılır. Eğer orada varsa aynı işlem o nokta için uygulanır. Bu işlem bütün katmanlara ve her piksele uygulandığı takdirde her bir pikselin hangi katmanda hangi değere sahip olacağı tekrar sıralanmış olur. Bu işlem, sentetik eksik verinin tutarlı bir küresel derinlik görseli paketi oluşturması için elzemdir. Kümülatif maskeleme, adını buradan alır. En üstteki katman yalnızca kendi maskesine muhatap iken, bir alt katman hem kendi maskesine göre veri kaybeder, hem de üsttekinin maskesine göre üst katmana alınması gereken verileri üst katmana verir. Dolayısıyla hem üst katmanın hem kendisinin maskesine muhatap olur. İç katmanlara indikçe bu sayı artmaktadır.

azami ortaklama ve yukarı örnekleme işleminden sonra yığın normalizasyonu katmanı yerleştirilmiştir. Bu katman, verinin uzun bir ağ boyunca normalize değerlerde seyretmesini sağlarken, gradyanların ve çıktı değerlerinin üstel yüksek veya üstel düşük gelmesine de engel olmaktadır. Modelin mimarisi Şekil 3.15'teki gibidir.

Bu çalışmada modelin öğrenme oranı 0.0005 ve 0.005 olacak şekilde iki farklı konfigürasyondur. Modelin masraf fonksiyonu ortalama karesel hata olarak belirlenmiştir fakat bu eğitim esnasında küresel derinlik uzayı için belirlenen masraf fonksiyonudur. Şekil tamamlamanın başarısını ölçmek için kullanılan metrik ortalama karesel hata de-



Şekil 3.15: Değiştirilmiş U-Net

ğildir. Adam Optimizer varsayılan parametreleriyle beraber kullanılmıştır, eğitim esnasında 50 adım (epoch) ve her adımda 1000 yığın (batch) kullanılmıştır. Her bir yığın dört şekilden ve her şekil üç katmandan oluşmaktadır, dolayısıyla her yığın 12 görselden oluşmaktadır. Adım başı (per epoch) bütün eğitim verisi gezilmemiştir, rastgele 1000 yığın (batch) seçilmiştir. Gizli katmanların aktivasyon fonksiyonu olarak ReLu seçilmiştir. Bu tercih, negatif veri beslemesini engellediği için kaybolan gradyanlar sorununun olmaması adına önemli bir adım olmuştur. Çıktı aktivasyon fonksiyonu olarak Sigmoid fonksiyonu seçilmiştir. Sürekli değer tahmini uygulamalarında çıktı aktivasyon fonksiyonu olarak genelde doğrusal aktivasyon tercih edilir, bunun nedeni de sonsuz aralığı ve sürekli değeri olmasıdır. Fakat bu çalışmada, girdi ve çıktı görsellerinin her bir pikselinin değeri  $[0, 1]$  aralığında olmak zorundadır. 0 şeklin merkezini temsil ederken 1 şekle en uzak noktanın derinliğini temsil eder. Çıktı değer aralığını limitlemek adına Sigmoid fonksiyonu tercih edilmiştir. Modelin öğrenilebilir parametre sayısı 1,872,065 adettir, bu da emsallerine göre oldukça az bir sayıdır. Az sayıda katmandan ve her katmanda nispeten daha az sayıda filtreden oluşuyor olması da yine öğrenme ve çalışma sürelerini kısaltmaktadır.

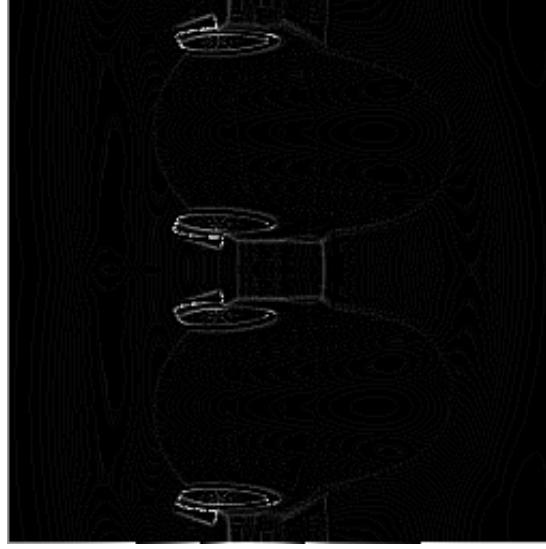
### 3.4.3 Önem Tabanlı Masraf Hesaplaması

Masraf fonksiyonu, çıktı görselindeki her bir piksel için hesaplanır ve masraf fonksiyonunun çıktıya göre birinci dereceden türevi çıktıdan geriye doğru dağıtılacak olan ilk gradyanları verir. Gradyanların büyüklüğü, önceki katmandaki ağırlıkların ne oranda

güncelleneceğini öğrenme oranıyla da bağlantılı olarak belirler. Sinir ağının daha iyi öğrenmesi istenilen öznelilikler varsa şayet, bunlara tekabül eden çıktılarının masraflarına ayrıca bir ceza uygulamak suretiyle bunların masrafı olduğundan fazla gösterilir ve böylece düzeltme oranında değişikliğe gidilmiş olunur [22, 24]. Çok katmanlı küresel derinlik parametrelerinin değerleri evrişim işlemine tabi tutulduktan sonra evrişim çekirdeklerinin komşu piksellerin değerlerini de öğrenilmiş ağırlıklarla çarpıp ilgili pikselin değerine yazması nedeniyle çok hafif de olsa blur benzeri bir etkiye sahip olur. Bu blur benzeri etki iki boyutta çok farkedilmese de üç boyutta bu işlemin sonucu iki geometrinin orta noktasında, geometrik bilgi olmaması gereken bir yerde geometrik bilgi oluşmasına neden olabilir. Bunun miktarı genelde azdır ve şeklin bütününe çok etki etmez, fakat düzeltilmesi için denemeler yapılabilir. Önem tabanlı masraf hesaplaması da bunlardan birisidir.

$$L_{onem} = L_{taban} + 3 \times I_{kenar} \times L_{taban} \quad (3.2)$$

Her taban doğru küresel derinlik parametresi için ilgili parametre görselini kenar tespit algoritmasına tâbi tutulur. Tespit edilen kenarlar, blurlanmanın en olumsuz etkileyeceği noktalardır. Basit bir keskinleştirme filtresinin etkisi bu işlem için yetersiz kalmaktadır. Öte yandan, sinir ağının eğitimini bunu düzeltecek şekilde tamamlamak da görseli sonradan işlemeye tercih edilir. Bu bağlamda, elde edilen kenar görseli ağa girdi olarak beslenir fakat ileri besleme veya geri dağılımda kullanılmaz, yalnızca masraf hesapları için masrafı manipüle etme amacıyla kullanılır. Kenar görselinde değeri yüksek olan yerlerin masraf değerleri kenar görselinin değeri ve bir katsayıyla çarpılarak artırılırken, kenar görselinde sıfır olan piksellerin masraf değerleri orijinal masraf değeri olarak tutulur. Böylelikle kenar noktalarının masrafına ayrıca bir ceza işlemi uygulanmış olur. Küresel derinlik görseli üzerine uygulanmış kenar tespit algoritmasının çıktısı Şekil 3.16'da görülebilir. Masraf fonksiyonu da 3.2 formülünde görülebilir. Bu yöntem, önem tabanlı olmayan masraf hesabına nazaran daha iyi bir sonuç vermediği için son konfigürasyonda kullanımı tercih edilmemiştir.



Şekil 3.16: Küresel Derinlik İmajının Kenar Tespiti (Üç ile çarpılmış değerler). Kenar tespit aracı olarak Alex Clark ve diğer katılımcıların, Fredrik Lundh ve diğer katılımcılar tarafından geliştirilen PIL kütüphanesi üzerine geliştirilen Pillow Python kütüphanesi kullanılmıştır [25].

#### 3.4.4 Deney

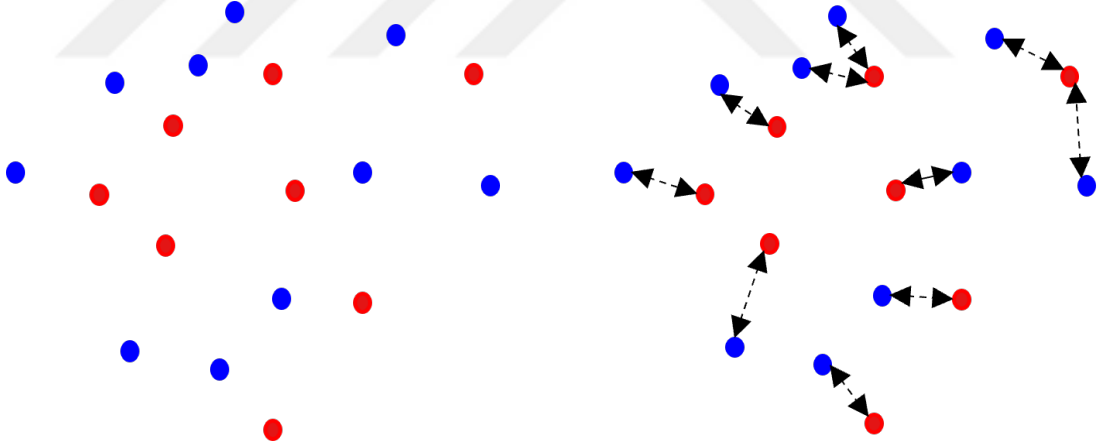
Deneyde eksik küresel derinlik görselleri ile 2B Unet görsel tamamlayacak şekilde eğitilmiştir. Eğitimde 4 farklı sınıfa ait (Otomobil, Sandalye, Kanepe ve Uçak) veriler kullanılmıştır. Bu sınıflara ait uçak için 6'şar, diğer sınıflar için 3'er adet küresel derinlik katmanı kullanılmıştır. Uçak için her bir katman 256x256 çözünürlükte modellenmişken diğer sınıflar için 128x128 modellenmiştir. Bunun nedeni, uçağın kontrol yüzeylerinin bulunduğu ince parçalarının (kanat ve dikey sabitleyici gibi) 128x128 küresel derinlik katmanı ile yeterince detaylı temsil edilemeyeşidir. Deney sonrasında yapılan gözlemde 256x256 çözünürlük için uçak küresel derinlik görsellerinden elde edilen taban doğru değerinde de bazı yüzeylerin detaylarının yeterince temsil edilemediği gözlemlenmiş, dolayısıyla uçak sınıfı için taban doğru referanslı kıyaslama yapılamayacağı kanısına varılmıştır. Eş ağırlıklı örneklerle yerine uçak gibi ince detay gerektiren sınıflarda önem örnekleme bir yöntem ihtiyacı duyulmuştur. Çok katmanlı küresel derinlik parametrelerinin ölçeklenebilir (scalable) oluşu sebebiyle çözünürlüğü artırmak problemi çözecek olsa da, önem örnekleme bir yaklaşım işlem süresi olarak daha yüksek performans sergileyecektir. Dörtlü ağaç yapılı adaptif küresel derinlik pa-

rametreleri çalışmamız da gerek olan bölgelerde gerekli detay seviyesini yakalayabilmek içindir. Diğer üç sınıfta ise böyle bir durum gözlemlenmemiştir. Eğitilen Unet ile bu diğer üç sınıfın eksik küresel derinlik görselleri tamamlanmış, tamamlanan görseller tekrar 3B Öklit uzayına dönüştürülmüş ve taban doğru küresel derinlik görsellerinin 3B Öklit uzayında tekabül ettiği nokta uzayıyla karşılaştırılmıştır. Eksik şekil tamamlama ve tamamlanan şeklin taban doğru şekil ile kıyaslanması işlemi, Çok Katmanlı Küresel Derinlik Parametreleri'nin yanısıra OCNN with Output Guided Skip Connections çalışması için de yapılmış, literatürde ustalık derecesindeki çalışmalara nazaran ÇKKDP'nin ne kadar başarılı olduğu incelenmiştir. Kıyaslama için OCNN with OGSC'nin tercih edilmiş olma nedeni, zaman ve bellek karmaşasının empirik olarak aynı polinomial seviyede olması ( $O(n^2)$ ) ve state of the art bir yöntem olmasındandır. OCNN çalışma serisinin Adaptive OCNN versiyonunda online notlarda zaman karmaşasına değinilmiştir [20]. Ayrıca ÇKKDP gibi geometrinin bütününe isabetli ve hassas bir şekilde temsil edebiliyor oluşu da önemli bir faktördür. Kıyaslama, iki çalışma için de x, y ve z eksenlerinde [0, 128] aralığına ölçeklendirilerek yapılmıştır.

OCNN with Output Guided Skip Connections çalışması veri yapısı anlamıyla kendisine özgü formatlar bulunduran bir çalışmadır. Eğitim ve deney için üç boyutlu geometri üzerinden rastgele örnekler alınır ve bu örnekler üzerinden eksiltme işlemi yapılır. Elde edilen nokta bulutu, çalışmanın kendi formatı olan Octree formatına dönüştürülür ve eğitim süreci de, test süreci de bu format üzerinden yürütülür. Çalışmanın araştırmacılar için sağlanan hazır eğitilmiş modeli ve kaynak kodu mevcuttur [26]. Kodun öncelikle derlenmesi ve sonra da kullanılması için yönergeler mevcut olsa da bazılarının kullanımı bu yönergelerle tam anlaşılabilir değildir. Detaylı anlamak için kodun, özellikle de eğitim ve test sürecini yöneten kodların dikkatle incelenmesi gerekmektedir. Çalışmada Python ve C++ dillerinden faydalanılmıştır. Python'da Tensorflow kullanılarak yapay sinir ağı modeli geliştirilmiş, gerekli olan özel sekizli ağaç katmanları için yüksek performanslı implementasyon C++ tarafında CUDA [27] ile yapılmıştır. Dolayısıyla, testi yürütebilmek için kodun C++ ve CUDA içeren kısmının derlenmesi gerekmektedir. Kod içerisinde nokta bulutu verilerini Octree formatına dönüştürecek yardımcı araçlar da mevcuttur. Fakat kodda birtakım varsayımlar olması,

kodu henüz incelememiş bir arařtırmacı için yardımcı araları kullanmayı zorlařtırabilmektedir. Octree dnüşüm araları, girdi verisi olarak  $[0, 128]$  deęer aralıęında üç ekseninde nokta bulutu istemektedir. Eęer daha düşük veya daha yüksek bir ölekte veri saęlanacak olursa, test kodu beklenmedik deęerler dnebilir. Öte yandan, yapay sinir aęının ıktı verisi olan Octree dosyaları yeniden nokta bulutuna dnüşürüldüęünde ise ıktı deęerleri girdinin aksine  $[0, 64]$  deęer aralıęında üç ekseninde bulunur. Eęer ıktı deęerinin taban doęru (ground truth) deęer ile kıyaslaması yapılacaksa, bu deęer aralıęına dikkat etmekte fayda vardır. Zira farklı ölekteki iki řekil arasında řekil 3.3'teki gibi Chamfer Mesafesi hesaplanırsa elde edilen deęer iki řeklin benzerlięi hakkında herhangi bir fikir vermez. řekiller tamamen aynı öleęe (scale), aynı dndürme miktarına (rotation) ve aynı öteleme miktarına (translation) sahip olmalıdır ki en yakın nokta kıyaslaması anlamlı bir hal alsın. Bu baęlamda deneyde OCNN with OGSC'nin ıktı deęerleri girdi ile aynı uzaya getirilmiřtir.

### 3.4.5 Chamfer Mesafesi



řekil 3.17: Soldaki řekilde mavi ve kırmızı iki farklı nokta bulutu bulunmaktadır. Bu iki farklı nokta bulutu arasındaki Chamfer Mesafesi'ni hesaplamak için iki nokta bulutu arasında en yakın nokta çiftleri bulunur, eřleşme yapıldıktan sonra mavi nokta bulutunun kırmızıya olan ortalama mesafesi hesaplanır, kırmızı mavi nokta bulutunun mavi nokta bulutuna olan ortalama mesafesi hesaplanır ve ikisi toplanır.

Karřılařtırma metrięi olarak Chamfer Mesafesi 3.3 kullanılmıřtır. Chamfer mesafesi, iki nokta bulutu arasındaki en yakın nokta çiftleri arasındaki ortalama uzaklıęı bulan

Çizelge 3.1: CHAMFER MESAFESİ SONUÇLARI

Eksik %	Sınıf	ÇKKDP		OCNN with OGSC
		$5 \times 10^{-4}$	$5 \times 10^{-3}$	
~ 38%	Otomobil	<b>3.066</b>	4.80	3.242
~ 52%	Otomobil	<b>3.153</b>	5.145	3.5032
~ 42%	Kanepe	<b>2.936</b>	4.271	3.881
~ 55%	Kanepe	<b>3.109</b>	4.414	4.246
~ 40%	Sandalye	<b>3.453</b>	4.100	3.701
~ 56%	Sandalye	<b>3.607</b>	4.690	4.079

bir yöntemdir. A ve B nokta bulutları arasındaki Chamfer Mesafesi aşağıdaki gibi hesaplanır:

$$D_{Ch} = \frac{1}{|A|} \times \sum_{p \in A, p_{min} \in B} \|p - p_{min}\|_2^2 + \frac{1}{|B|} \times \sum_{q \in B, q_{min} \in A} \|q - q_{min}\|_2^2 \quad (3.3)$$

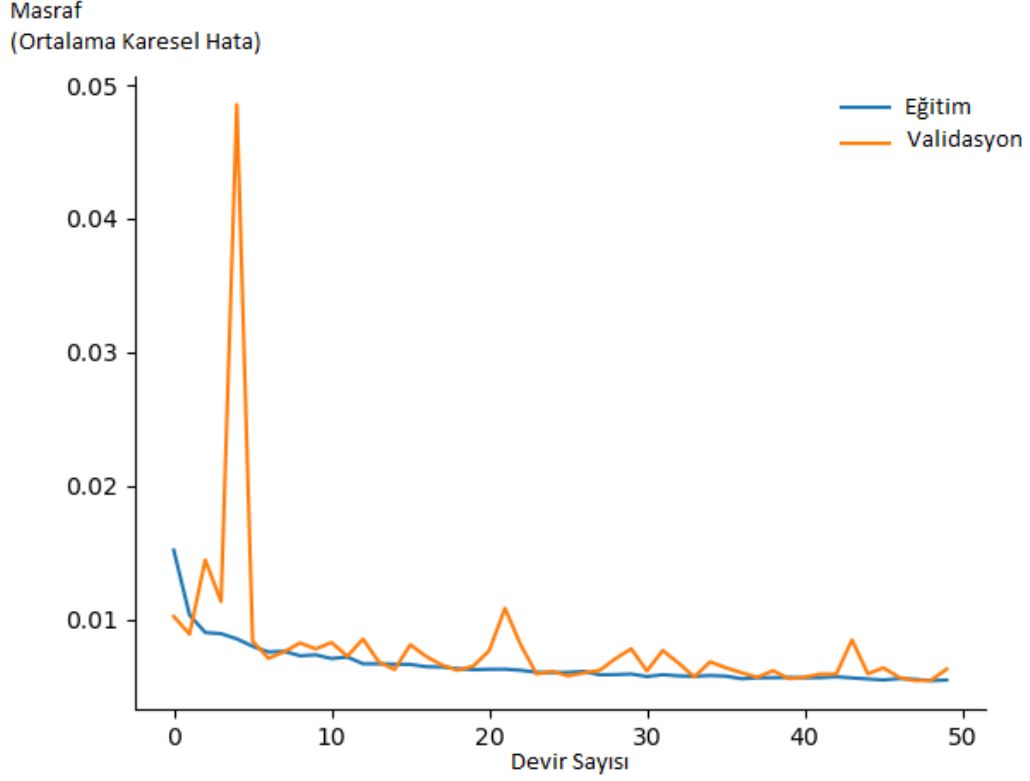
Şekil 3.17’de Chamfer Mesafesi hesaplaması görselleştirilmiştir.

- A nokta bulutundaki her bir nokta için B’deki en yakın nokta bulunur, A’daki her noktanın çiftine olan Öklit mesafesi bulunur ve bütün sonuçların ortalaması alınır.
- B nokta bulutundaki her bir nokta için A’daki en yakın nokta bulunur, B’deki her noktanın çiftine olan Öklit mesafesi bulunur ve bütün sonuçların ortalaması alınır.
- İki ortalama değer toplanır ve Chamfer Mesafesi bulunmuş olur. Formülde (3.3) matematiksel notasyonu gösterilmektedir.

### 3.4.6 Sonuçlar ve Analiz

Çizelge 3.1’de en soldaki sütun, ortalama eksiltelen nokta sayısını göstermektedir. Her sınıf için bu miktarın farklı olması, maskeleme işleminin tamamen rastgele küresel aç-

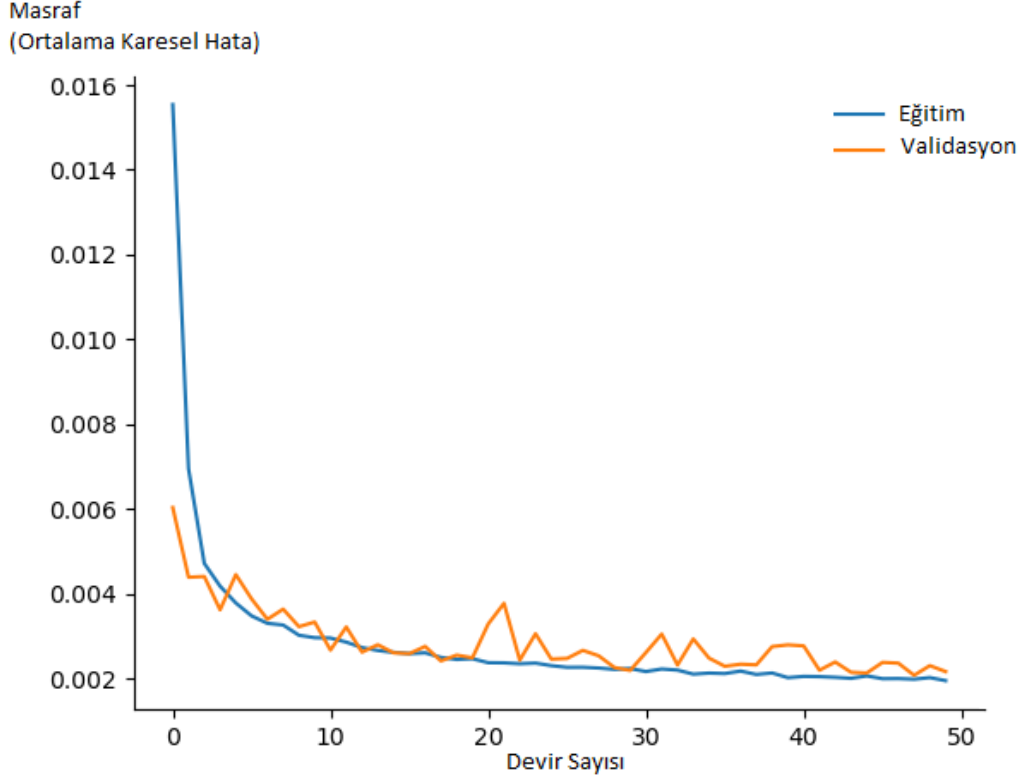




Şekil 3.18:  $5 \times 10^{-3}$  öğrenme oranıyla modelin eğitim esnasındaki eğitim ve validasyon verisi masraf değeri grafiği

larda yapılmasındandır. Her şekile aynı maskelenme uygulansa dahi bu miktar farklı çıkacaktır, çünkü her şeklin aynı küresel açısı bir yüzeye tekabül ediyor veya etmiyor olmak zorunda değil. ÇKKDP ( $lr = 5 \times 10^{-4}$ ) başlığı,  $5 \times 10^{-4}$  öğrenme oranıyla eğitilmiş modelin sonuçlarını listelerken ÇKKDP ( $lr = 5 \times 10^{-3}$ ) başlığı, öğrenme oranı  $5 \times 10^{-3}$  ile eğitilen modelin sonuçlarını listeler. OCNN with OGSC ise çalışmadaki test senaryosu için sağlanan varsayılan parametrelerle test edilmiştir. Deney sonuçları Şekillerde ve kıyaslama Şekillerinde gösterilmiştir. Sayısal ve görsel sonuçlara göre, Unet tabanlı model ÇKKDP ile ve  $5 \times 10^{-4}$  öğrenme oranıyla eğitildiği takdirde bütün testlerde OCNN with OGSC'den daha yüksek başarı oranı elde etmişken  $5 \times 10^{-3}$  öğrenme oranıyla daha düşük başarı oranı elde etmiştir.

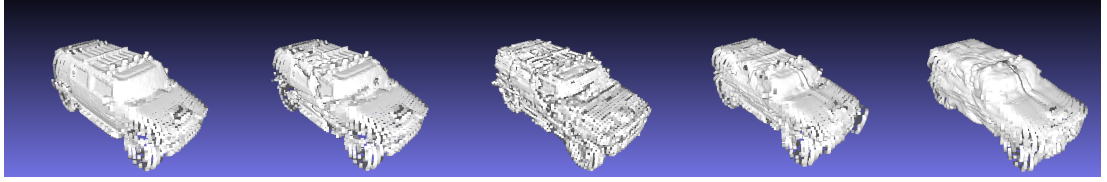
Öğrenme oranı, özellikle eğitimin ilerleyen aşamalarında yani daha yüksek dereceli azaltma hassasiyeti gerektiren yerlerde etkisini daha çok göstermektedir. Yüksek bir öğrenme oranı, tek seferde dendrit ağırlıklarını çok fazla oynatacağı için masraf eğri-



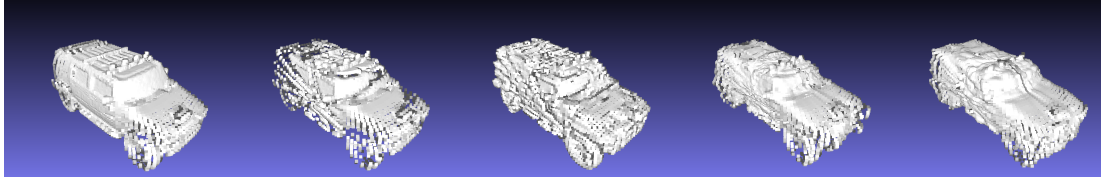
Şekil 3.19:  $5 \times 10^{-4}$  öğrenme oranıyla modelin eğitim esnasındaki eğitim ve validasyon verisi masraf değeri grafiği

sinde ileri geri gidildiği halde ortada, masrafın düşük olacağı noktada durmak mümkün olmayacaktır. Öğrenme oranının çok düşük olması durumunda da masrafın optimal değerine yakınsamak çok uzun vakit alacaktır. Şekil 3.19'da görüleceği üzere,  $5 \times 10^{-4}$  öğrenme oranıyla masraf eğrisi düzelme trendine girmiş ve hatta son devirlerde sabite yakın seyretmiştir. Bu da  $5 \times 10^{-4}$  öğrenme oranıyla tutarlı bir eğitim yapabildiğimizi, masraf eğrisinde sekmelere sebebiyet vermediğimizi gösterir. Öte yandan  $5 \times 10^{-3}$  öğrenme oranıyla grafikte zikzaklar, Şekil 3.18'de görülebildiği üzere daha büyük ölçektir (iki grafik arasındaki ölçek farkına dikkat edilmelidir). Bu da masraf değerinin artıp azaldığı son devirlerde arada başka bir optimal değer olabileceğini fakat öğrenme oranının bu değeri yakalayacak kadar hassas olmadığını gösterir.

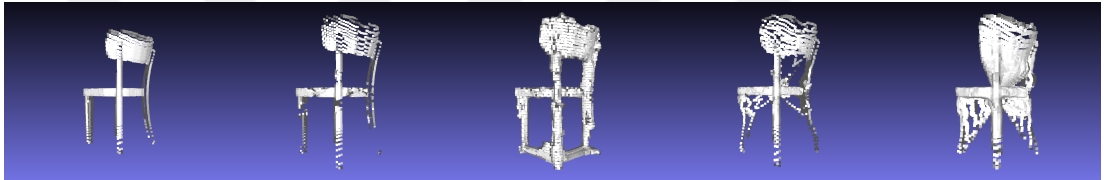
Deneyde eksiltilmiş nokta sayısı da değişiklik göstermektedir. İlk basamakta  $\sim 40\%$  dolaylarında veri silinip tamamlama işlemi gerçekleştirilmişken ikinci basamakta  $\sim 52\%$  dolaylarında veri silinip tamamlama işlemi gerçekleştirilmiştir. Hem ÇKKDP hem



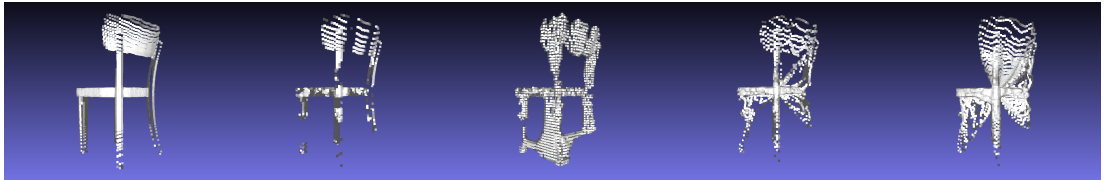
Şekil 3.20: Otomobil Şekli Tamamlama ( $\sim 38$  eksik geometri ile). Soldan Sağa Doğru: Taban Doğru, Eksik Geometri, OCNN with OGSC, ÇKKDP  $lr=10^{-3}$ , ÇKKDP  $lr=10^{-4}$ . Görselleştirme MeshLab [2] ile elde edilmiştir.



Şekil 3.21: Otomobil Şekli Tamamlama ( $\sim 52$  eksik geometri ile). Soldan Sağa Doğru: Taban Doğru, Eksik Geometri, OCNN with OGSC, ÇKKDP  $lr=10^{-3}$ , ÇKKDP  $lr=10^{-4}$ . Görselleştirme MeshLab [2] ile elde edilmiştir.

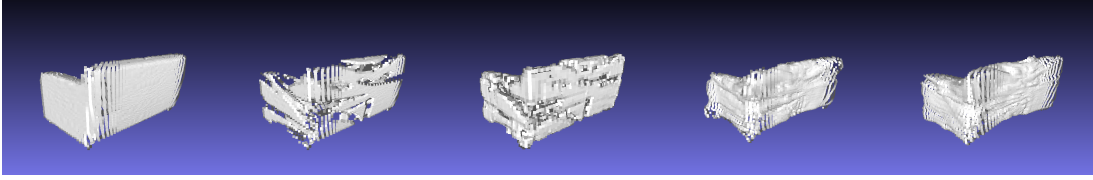


Şekil 3.22: Sandalye Şekli Tamamlama ( $\sim 40$  eksik geometri ile). Soldan Sağa Doğru: Taban Doğru, Eksik Geometri, OCNN with OGSC, ÇKKDP  $lr=10^{-3}$ , ÇKKDP  $lr=10^{-4}$ . Görselleştirme MeshLab [2] ile elde edilmiştir.

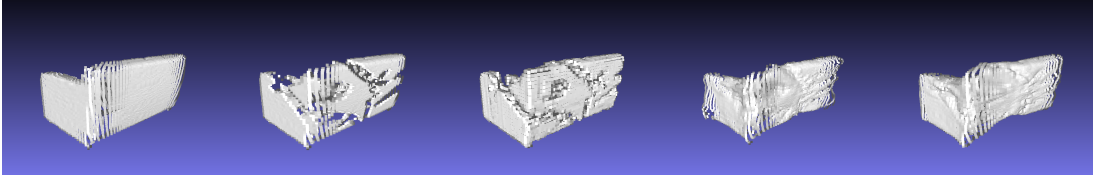


Şekil 3.23: Sandalye Şekli Tamamlama ( $\sim 56$  eksik geometri ile). Soldan Sağa Doğru: Taban Doğru, Eksik Geometri, OCNN with OGSC, ÇKKDP  $lr=10^{-3}$ , ÇKKDP  $lr=10^{-4}$ . Görselleştirme MeshLab [2] ile elde edilmiştir.

OCNN with OGSC, daha fazla sayıda nokta silindiği takdirde az sayıda nokta silindiğinde gösterdiği performansın aşağısında başarı oranı göstermiştir. Bunun en temel sebebi, evrişim işleminin komşuluk tabanlı çalışıyor olmasıdır. iki veya üç boyutta bir değer bir dizi evrişim işlemiyle tahmin edilecekse şayer, o verinin komşu lokasyonundaki verilerin o veriye dair anlam ima etmesi gerekir. Eğer komşu lokasyonlarda da yeterince veri yoksa, ilgili veriyi tahmin ederken kullanılan referans değerler azalacaktır.



Şekil 3.24: Kanepe Şekli Tamamlama ( $\sim 42$  eksik geometri ile). Soldan Sağa Doğru: Taban Doğru, Eksik Geometri, OCNN with OGSC, ÇKKDP  $lr=10^{-3}$ , ÇKKDP  $lr=10^{-4}$ . Görselleştirme MeshLab [2] ile elde edilmiştir.



Şekil 3.25: Kanepe Şekli Tamamlama ( $\sim 55$  eksik geometri ile). Soldan Sağa Doğru: Taban Doğru, Eksik Geometri, OCNN with OGSC, ÇKKDP  $lr=10^{-3}$ , ÇKKDP  $lr=10^{-4}$ . Görselleştirme MeshLab [2] ile elde edilmiştir.

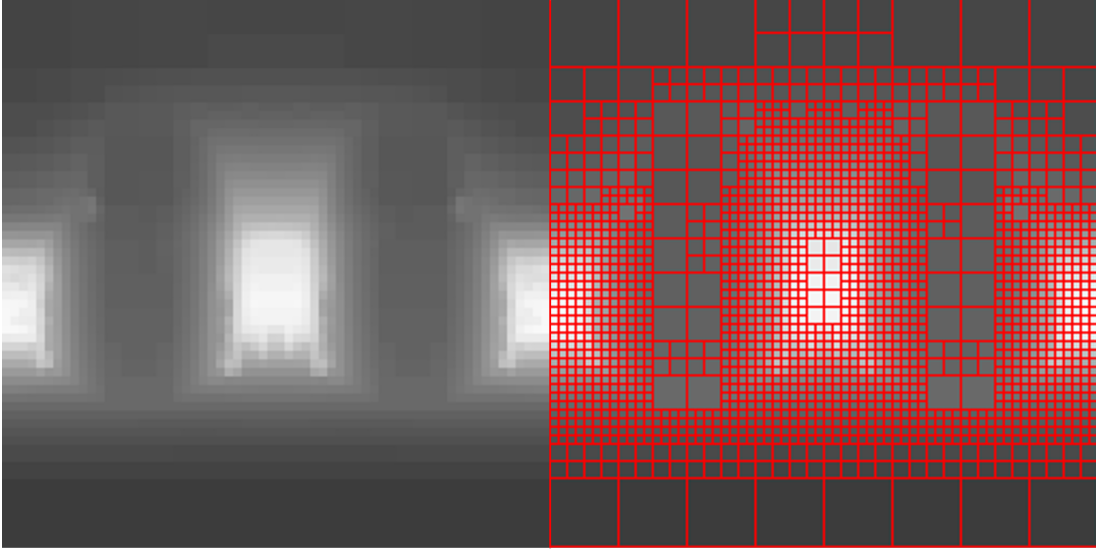
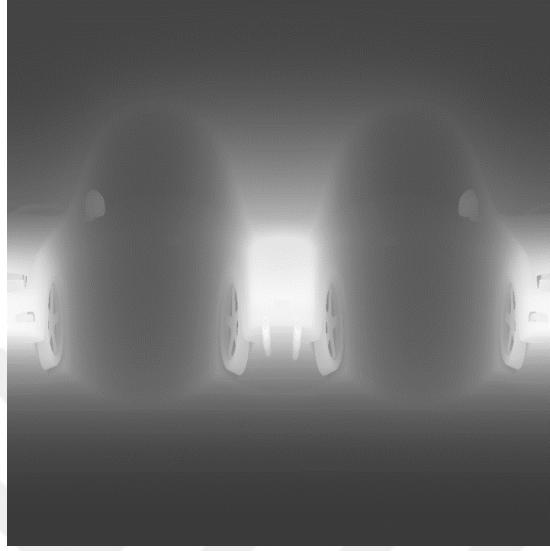
Elbette derin sinir ağlarında ilerleyen basamaklarda daha geniş çerçeveler de ilgili veri için göz önünde bulundurulacaktır fakat bunların hiçbiri en yakın lokasyondaki veriler kadar bilgi içermeyecektir. Dolayısıyla, ne kadar çok nokta azalırse komşuluk ilişkileri o ölçüde azalır. Ne kadar çok nokta azalırse komşulara bakarak yapılan tahminler o kadar isabetsiz hale gelir ve başarımları düşer. OCNN with OGSC’de de ÇKKDP’de de evrişim operasyonu kullanıldığından, eksiltelen nokta sayısının artmasına bağlı olarak ikisinin de performansında azalma gözlenmiştir. Şekil 3.20, 3.21, 3.22, 3.23, 3.24 ve 3.25’te eksiltelen nokta sayısının ve farklı öğrenme oranlarının Çizelge 3.1’de anlatılan etkisi görsel olarak da incelenebilir.

#### 4. Adaptif Küresel Derinlik Parametreleriyle 3B Şekil Tamamlama

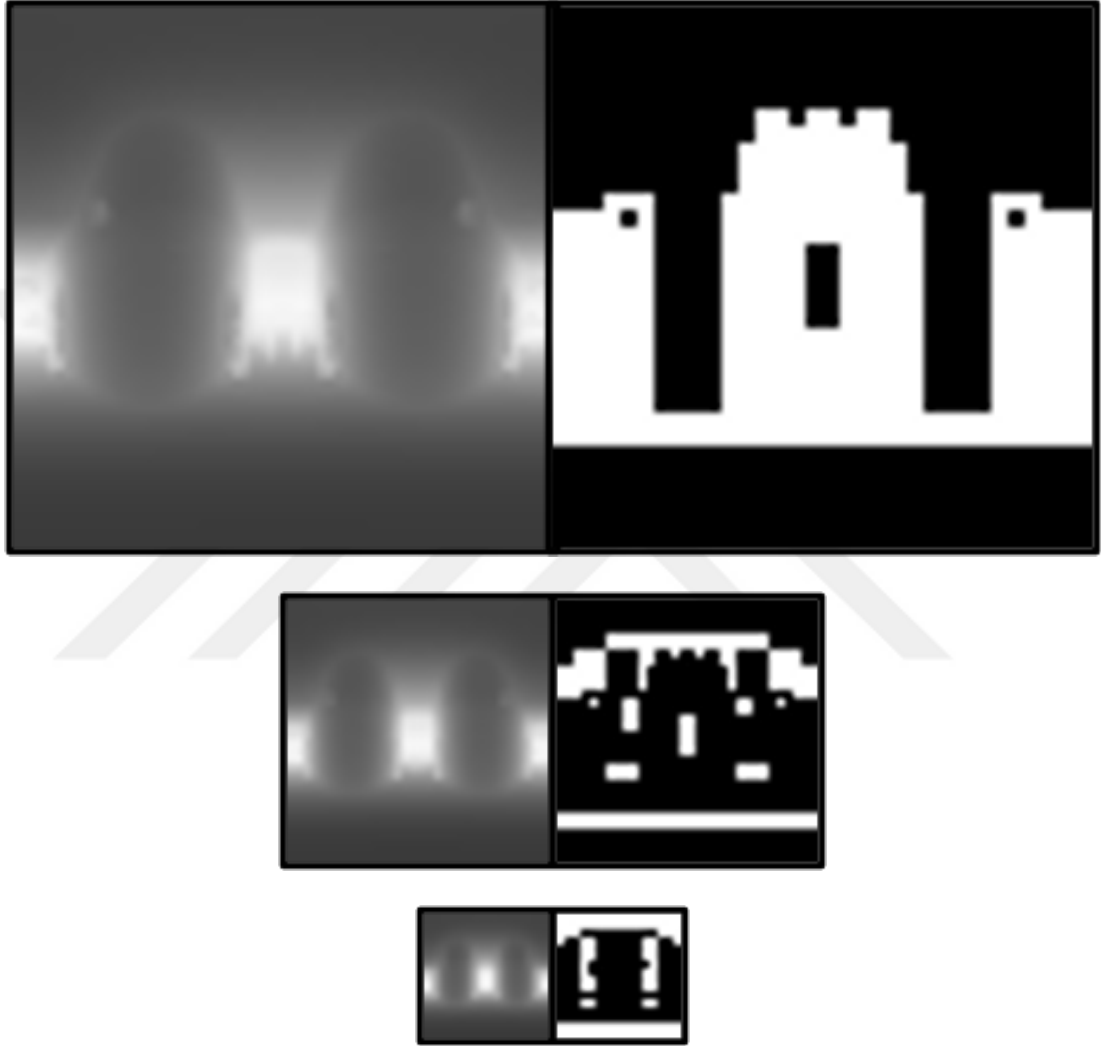
Bu çalışmada, [29] ve [9] çalışmalarında kullanılan adaptif yöntemlere benzer şekilde, küresel derinlik görseli dörtlü ağaç (quad tree) yapısına bölünmüştür. Bu şekilde, görselde önem gerektiren bölümler daha yüksek çözünürlükte temsil edilirken daha az detay gerektiren bölümler daha düşük çözünürlükte temsil edilmiştir. İşlenilecek veri sayısı da bu şekilde düşürülmüştür. Fakat standart U-Net tipi bir evrişimsel sinir ağı, böyle bir girdi verisini kabul etmemektedir çünkü girdi verisinin düzenli ve eşit aralıklı verilerden oluşması, homojen bir ızgara yapıda olması gerekmektedir. Standart U-Net'in girdi sayısı artırılarak her detay seviyesi için ayrı görsel verilmesi, hepsinin hangi bölgelerinin işleneceğinin işaretlenmesi ve yalnızca işaretli bölgeleri işleyecek birer evrişim (convolution) ve azami ortaklama (max pooling) katman implementasyonu ilgili verinin optimal bir şekilde işlenmesine ve dörtlü ağaç yapısından kaynaklanan veri sayısı azalmasından verim anlamında faydalanılmasına olanak tanıyacaktır. Bu farklı detay seviyesindeki katmanların kodlama işlemleri devam ettikçe birbirleriyle birleştirilmeleri gerekmektedir, çünkü bütün detay seviyesinde derin sinir ağı tarafından çıkarılmış özniteliklerin darboğaz işlemine kadar birleştirilmiş ve aynı öznitelik matrisini doldurmuş olması gerekmektedir. Bunların tümü aynı matrise veri yazacağı için, bu katmanların çıktılarını birleştirmek üzere yeni bir birleştirme (concatenation) katmanı yazılması elzemdir. Bu katmanın sıradan bir birleştirme katmanından farkı, iki farklı katmandan gelen bilgiyi arka arkaya ekleyip iki kat boyuta sahip daha büyük bir öznitelik matrisi çıkarmak yerine iki farklı katmandan gelen bilgiyi birleştirip tek ve girdi katmanlarla aynı boyutlara sahip bir öznitelik matrisine yazması olacaktır. Çalışmada eğitim veri kümesi olarak [3] ve [7] çalışmalarından faydalanılmıştır.

##### 4.1 Eğitim Verisi Hazırlama

Adaptif küresel derinlik parametreleri, küresel derinlik parametreleri görselinin detay seviyesine göre dörtlü ağaca bölünmesiyle oluşur. İlk olarak bütün bir görüntünün piksellerinin ortalama ve standart sapma değeri hesaplanır. Hesaplanan standart sapma



Şekil 4.1: Tek Katmanlı Küresel Derinlik Parametrelerinin Dörtlü Ağaca (Quad Tree) Bölünmüş Görüntüsü



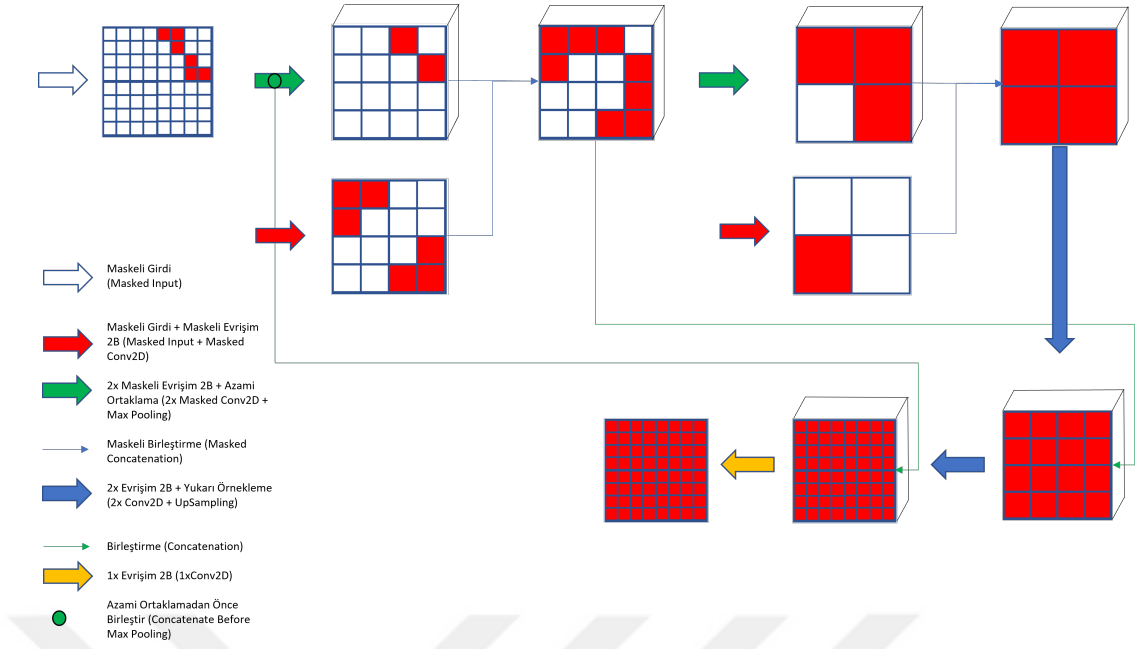
Şekil 4.2: Adaptif Çok Katmanlı Küresel Derinlik Parametreleri

değeri belli bir değerin üstündeyse görüntü tam ortadan dört eş boyutta alt görüntüye bölünür. Eğer standart sapma değeri ilgili değerin altındaysa görüntü bölünmez. Görüntü bölündüyse şayet, her bir alt görüntü için aynı işlem tekrarlanır. Şekil 4.1 bu prosedürün neticesini göstermektedir. Bölünme işlemi, ya önceden belirlenmiş bir derinlik seviyesinde ya da alt görüntü tek bir piksel kalıncaya kadar devam eder. Bu çalışmada görüntü üç kademe derinliğe kadar bölünmüştür. Sırasıyla  $64 \times 64$ ,  $32 \times 32$  ve  $16 \times 16$  çözünürlükte görüntüler elde edilmiştir. Validasyon deneyi için öncelikle otomobil sınıfı kullanılmıştır. Bu görüntülerin bir de aynı çözünürlükte maskeleri vardır, maske, hangi görseldeki hangi bilgilerin o görselin dörtlü ağaç derinliğine ait olduğunu, yani o katmanda işleneceğini anlatır. Küresel derinlik görüntüsü katmanları ve maskeleri Şekil 4.2’de gösterilmektedir. Bu görüntüler maskeleriyle birlikte modele girdi olarak beslenir. Bu veriler, üç katmandan üç farklı girdi olarak modele girilir. Maskelenen bölgelerden elde edilen toplam öznitelik sayısı yaklaşık olarak 3300 civarlarındadır. Bu sayı, 4096 sayısına nazaran oldukça düşüktür. Dörtlü ağacın derinliği artırılarak öznitelik sayısı daha da düşürülebilir.

## 4.2 Sinir Ağı Modeli

[9, 29] çalışmalarında kullanılan yöntem benzer şekilde, fakat yalnızca kodlama katmanlarında olacak şekilde üç kodlama (encoding) ve üç deşifre (decoding) kompleksinden oluşan bir U-Net’e her bir kodlama kompleksinde girdi olarak maskelenmiş görsel girilir. Adaptive OCNN’de [9] deşifre katmanlarında da maskeleyme işlemi uygulanır, nerenin maskelenmesi gerektiğini tahmin etmek için iki tane tam bağlı katmanlı (dense layer) ve bir tane de yığın normalizasyon katmanlı (batch normalization layer) bir sinir ağı kullanılır. Girdi görseli, dörtlü ağacın en alt derinlik seviyesinden kodlama kompleksinin endeksi kadar yukarı çıkılmış seviyesindeki bilgisidir. 0. kodlama kompleksi için dörtlü ağacın en alt derinlik seviyesindeki bilgi girilirken, 1. kodlama kompleksi için bunun bir yukarısı, 2. kodlama kompleksi için ise iki yukarısı verilir. Şekil 4.3’te görüldüğü üzere kodlama komplekslerinden girilen bilgiler birleştirilerek darboğaza (bottleneck) doğru ilerletilir. Bilgi deşifre katmanlarına hem darboğaz üye-





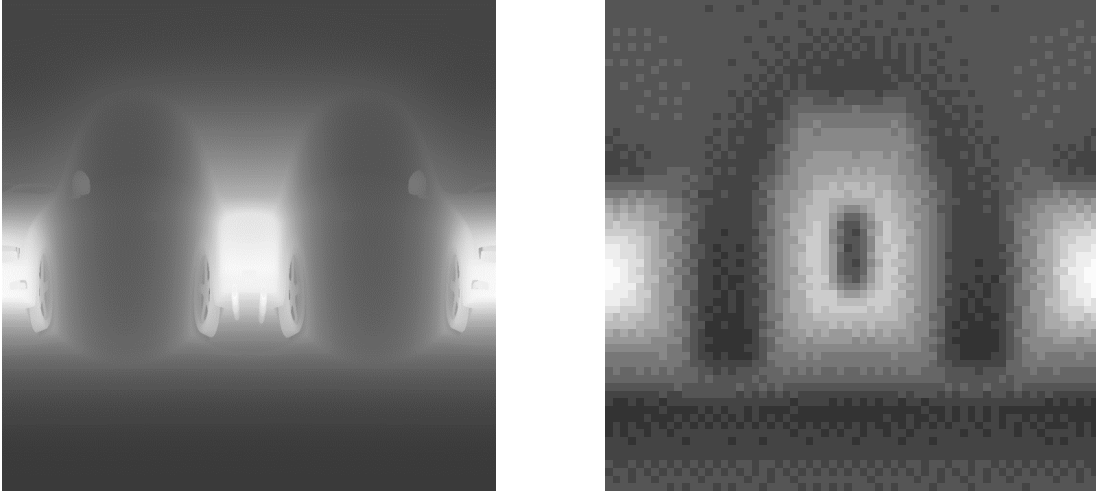
Şekil 4.3: Adaptif Küresel Derinlik Tamamlayıcı U-Net

rinden, hem de atlama katmanları (skip connections) üzerinden aktarılır. Neticede tek bir görsel çıktı elde edilir.

Modelin tamamen C++ ile merkezi işlemcide (CPU'da) çok görevli (multi-threaded) çalışacak şekilde implement edilmiştir. Modelin kendisi, modelde kullanılan tüm katsayılar, dörtlü ağaç veri kümesi yöneticisi de dahil olmak üzere C++ ile belirtilen şekilde en temelden yapılmıştır. Hazır derin öğrenme kütüphaneleri yerine kendi implementasyonumuzu tercih etmemizin sebebi, optimizasyonun evrişim uygulanacak yüzeyi evrişim işleminden önce maskeleyerek yapıyor olmamızdır. Evrişim sonucunu maskelemenin performans getirisi olmayacağından ve mevcut kütüphanelerle bu işlemi yapabilmek için zaten katmanları en temelden tekrar implement etmek gerektiğinden, tümüyle kendi implementasyonumuza bağlı kaldık.

### 4.3 Deney

Öncelikle ağın doğru çalışıp çalışmadığından emin olunması gerektiğinden, overfit testi uygulanmıştır. Bu testte ağın, girdi olarak verilen görselin aynısını çıktı olarak



Şekil 4.4: Olması Gereken Şekil - Overfit Testi Sonucu Kıyaslaması

oluşturup oluşturamayacağı denenmiştir. Bu şekilde eğitilebilir parametrelerin, yoğunluklu olarak evrişim katsayılarının (convolution kernel weights), tam anlamıyla kullanılıp kullanılmadığı gözlemlenmiştir. Şekil 4.4'te olması gereken şekil ile overfit testi sonucu üretilen şeklin kıyaslaması görülmektedir. Gelecek çalışma olarak OCNN'deki gibi [8] deşifre katmanlarına akıllı maske tahmini yapıp deşifre katmanlarını maskeleyecek ayrı bir sinir ağının implementasyonu, modelin veriyi daha iyi öğrenmesine olanak tanıyabileceğinden denemeye değer bir yöntemdir.

## 5. Koşullu Çekişmeli Üretken Sinir Ağları ve Küresel Derinlik Parametreleriyle Öğrenilmiş Sınıf Özgün 3B Şekil Üretimi

Çekişmeli üretken sinir ağları, iki ağın birbirinin çıkarlarına ters masraf fonksiyonlarıyla eğitilmesi sonucunda öğrenilmiş veriye benzer veriler üreten bir sinir ağı türüdür. En büyük avantajlarından birisi, veri üretmek için makul girdi değerlerine ihtiyacı yoktur, öğretildiği takdirde rastgele verilerden mantıklı veriler üretebilir. Bu da çekişmeli üretken ağların az sayıda girdiden sayısız ve daha önce görülmemiş veri üretmesine olanak tanır. Üretilen veriler öğrenilenlere benzerdir ama düzgün bir eğitime tâbi tutulduktan sonra asla aynısı değildir. Bu çalışmada, 2B görsel formunda olan tek katmanlı küresel derinlik parametreleriyle çekişmeli sinir ağlarını kullanarak özgün 3B şekil üretimi yapılacaktır. Bu çalışmada derin öğrenme kütüphanesi olarak TensorFlow [28] kullanılmıştır. Çalışmada veri kümesi olarak [3] ve [7] çalışmalarından faydalanılmıştır. Bu veri kümelerinin otomobil, sandalye, kanepeler ve uçak sınıfları Koşullu Çekişmeli Üretken Sinir Ağları'nı [33] eğitmek için kullanılmıştır.

### 5.1 Çekişmeli Üretken Sinir Ağları ve Koşullu Çekişmeli Üretken Sinir Ağları

Çekişmeli üretken sinir ağları [30] en temel formda iki farklı sinir ağından oluşur. Bunlardan birisinin adı üretken ağ iken diğerinin adı ayırt edici ağdır. Üretken ağın görevi eğitim verilerindeki verilere mümkün mertebe benzer veriler üretmektir. Ayırt edici ağın ise temel görevi üretilen verilerin gerçek olmadığını tespit edebilmek, gerçek olan verilerin ise gerçek olduğunu tespit edebilmektir. Yani bu iki ağın temel görevi üretken ağ için ayırt edici ağ kandırmak, ayırt edici ağ için ise üretken ağa kanmamaktır. Bu iki ağ, beslenecek girdi verisinin şekline ve elde edilecek çıktı verisinin şekline bağlı olarak herhangi bir formda olabilirler. Bizim örneğimizde, 2B görsel verisi işlediğimiz için 2B evrişim tabanlı bir üretken sinir ağı ve yine 2B evrişim tabanlı bir ayırt edici ağ gerekli olacaktır [36]. Diğer bir deyişle, üretken ağ 2B görsel çıktı verirken ayırt edici ağ 2B görsel girdi alacak ve bir sınıflandırma sonucu dönecektir. Bu sınıflandırma sonucu, üretilen verinin gerçek olup olmadığının sınıflandırılmasıdır.

Üretken ağı girdi olarak alacağı bilginin şekli uygulamaya göre değişir. Bu çalışmada, özgün şekiller üretilmesi hedeflendiği için, rastgele sayılardan oluşan bir tek boyutlu vektör girdi olarak alınacaktır. Bu vektörün adı gizli vektördür. Bu adın verilmiş olmasının sebebi, bu vektör baz alınarak gerçeğe yakın veriler üretilecektir fakat vektöre bakınca üretilen olan veriyi kestirmek mümkün değildir. Bu vektörün her bir değer kombinasyonu, öğrenilmiş sınıflara ait özgün bir veri üretme potansiyeline sahiptir. 100 adet 32 bitlik tam sayıdan oluşan bir gizli vektörün alabileceği farklı değer sayısı  $2^{3200}$  adettir. Bu da, düzgün eğitilmiş bir üretken ağı belki de test edilemeyecek kadar çok sayıda veri üretebileceğini gösterir. Yeterince eğitim verisi elde olduğu müddetçe çekişmeli üretken ağlar çok güçlü veri üretim araçlarıdır. Tek boyutlu rastgele sayı dizisi girdi alan ve 2B görsel çıktı veren üretken ağ ile 2B görsel girdi alıp sınıflandırma sonucu çıktı veren ayırt edici ağdan oluşan ağ kompleksinin girdisi rastgele sayı dizisiyken çıktısı sınıflandırma sonucudur.

Çekişmeli üretken ağların bir özel türü de öğreneceği verinin sınıfını girdi olarak alan, dolayısıyla sınıf farkını da öğrenebilen ağ türüdür. Bu ağa koşullu üretken ağ denilir [33]. Bu tür bir çekişmeli üretken ağda varsayılandan farklı olarak hem üretken ağ hem de ayırt edici ağ ilgili verinin sınıf bilgisini girdi olarak alır. Böylelikle üretken ağ veri üretirken istenilen sınıfa ait veri üretme kabiliyetine sahip olurken, ayırt edici ağ da ayırt etme esnasında verinin ilgili sınıfa benzeyip benzemediğini daha geniş aralıklı bir karar hiper-yüzeyiyle yapma şansına sahip olur. Üretilen herhangi bir verinin bütün bir veri kümesindeki verilere benzeyip benzemediğini, sahte olup olmadığını ayırt etmek zorken ilgili verinin arabaya benzeyip benzemediğini söylemek nispeten daha kolaydır. Koşullu üretken ağların akış diyagramı Şekil 5.1’de mevcuttur.

Evrışimsel çekişmeli üretken sinir ağlarının koşullu versiyonu için detaylı inceleme, karşılaştırma ve eğitim materyali Brownlee’nin [37] çalışmasında bulunabilir.

## 5.2 Koşullu Çekişmeli Üretken Ağ Eğitimi

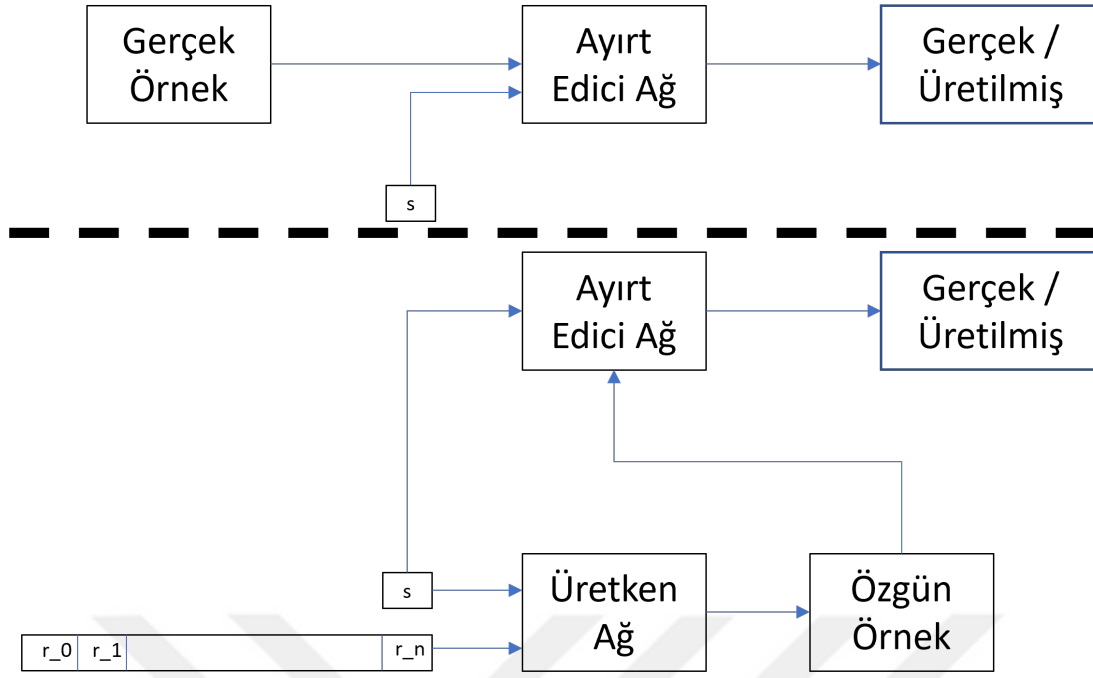
Çekişmeli üretken ağlarda birbiriyle çekişen iki ağın her birinin başarısı diğeri için başarısızlık, başarısızlığı da diğeri için başarıdır. Dolayısıyla, ikisinin masrafı da bir diğerinin başarısı oranındadır.

Tahmin edileceği üzere, Şekil 5.1'de de görüldüğü gibi ayırt edici ağ ilk etapta veri kümesindeki gerçek verilerle eğitilir, bu veriler için olumlu yani bir dönencek şekilde masrafı hesaplanır. İkinci olarak da üretken ağın o zaman dilimindeki halinin ürettiği değerler beslenir. Bunlar da olumsuz yani sıfır olarak sınıflandırılır. Buradaki sınıf, verinin veri kümesinde hangi sınıfa ait olduğu değil, üretim veya gerçek oluşudur. Sonrasında sıra üretken ağı eğitmeye gelir. Üretken ağa bir gizli vektör beslenmek suretiyle tekrar rastgele örnekler üretilir. Bütün bir ağ kompleksi (hem üretken hem de ayırt edici ağı içerecek şekilde), üretken ağın ürettiği değer olumlu sınıflandırılarak geri dağıtılır. Böylelikle ayırt edici hata yapmış olur ve üretken ağ da ayırt edici ağın hatasına göre eğitilir.

Burada dikkat edilmesi gereken birkaç unsur vardır. Ne ayırt edici ne de üretken ağ mükemmel eğitilmelidir. Birisinin diğerine baskın gelmesi, ya ayırt edicinin tüm üretim örneklerini olumsuz olarak sınıflandırmasıyla sonuçlanır ki bu durumda ayırt edicinin masrafı sıfır gelirken üretkenin masrafı sürekli artar, ya da ayırt edicinin tüm üretim örneklerini olumlu olarak sınıflandırmasıyla sonuçlanır ki bu durumda da ayırt edicinin masrafı sürekli artarken üretkenin masrafı sıfır gelir. Üretken eğitilirken yalnızca üretkene değil tüm ağ kompleksine üretilmiş veri olumlu olarak geri dağıtıldığında ayırt edicinin eğitim verisine aşırı bağlanmasının da önüne geçilmiş olur.

İki ağın birbirini domine etmemesi için ilgili ağların hiper-parametreleri de büyük önem taşımaktadır. Çok gelişkin bir üretken ağ ayırt ediciyi domine edecekken çok gelişmiş bir ayırt edici ağ da üretkeni domine edecektir.

Bir diğer olası sorun da 2B görsel üreten çekişmeli üretken ağlarda görülen bir sorundur. Bu sorun, evrişim çekirdeklerinin görsel üzerinde atlayarak ilerleme miktarının



Şekil 5.1: Koşullu Çekişmeli Üretken Ağ Örneği:

Üretken ağın girdilerinden  $r$  vektörü rastgele gizli vektörü temsil ederken  $s$  girdisi ilgili verinin hangi sınıfa ait olduğunu veya olması gerektiğini gösterir. Şeklin yukarıda kalan kısmı eğitimin ayırt edici ağı eğitme kısmını gösterirken aşağıda kalan kısmı üretken ağın eğitimini gösterir.

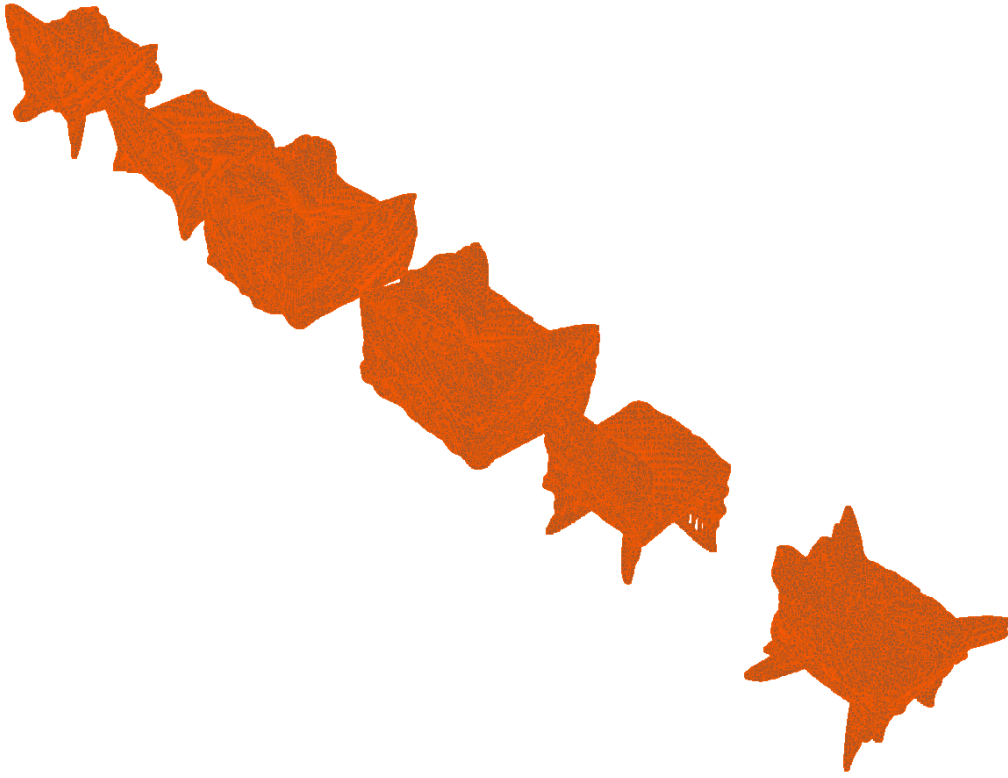
çekirdek boyuna tam bölünmemesinden kaynaklanır [34]. Bu durumda evrişim çekirdeği görsel üzerinde farklı noktalarda gezerken çıktı görseli üzerinde çekirdeğin kenarları önceki işlemlerle kesişir ve damalı bir örüntü ortaya çıkar. RGB görsellerde de sıkıntı olan bu durum, derinlik bilgisi tutan görsellerde daha büyük sorundur çünkü ilgili pikselin tekabüle ettiği doğrultuya orada olmayan bir geometri yerleştirmiş olur. İki adet çözümü vardır [34]. Birincisi çekirdek boyuna bölünebilir atlama miktarları kullanmaktır, uygulaması oldukça kolay bir yöntemdir. İkincisi de gizli vektörden 2B görsel oluştururken ters evrişim operasyonu yerine yukarı örnekleme + evrişim operasyonu kullanmaktır. Bu şekilde resmi büyütme işlemi ters evrişimle değil yukarı örneklemeyle yapılır, ondan sonra evrişim işlemi yukarı örneklenmiş görselin üzerine uygulanır ve tam bölünme sorunu ortadan kalkmış olur.



Şekil 5.2: Çekişmeli Üretken Ağlar ile Üretilen Bir Otomobil Modeli ve Eğitim Veri Kümesinde Kendisine En Çok Benzeyen Otomobil Modelinin Karşılaştırılması (Üretilen model turuncu renkte iken eğitim verisi sarı renktedir). Resimlerde pikseller  $[0, 255]$  aralığında değer almaktadır. İki resim arasındaki ortalama karesel hata 24.634 olarak ölçülmüştür. Görsel olarak da sayısal olarak da iki şeklin farklı olduğu görülmektedir. Üretilmiş şekildeki dikiz aynası, rüzgarlık, tampon çizgileri gibi detaylar çekişmeli üretken ağların ne denli detaylı modelleme yapabildiğini gösterirken, bu detayların şekle en benzeyen eğitim şeklindekilerden farklı oluşu da ağın mod çöküşü yaşamadığını göstermektedir.



Şekil 5.3: Çekişmeli Üretken Ağlar ile Üretilmiş Rastgele Otomobil Şekilleri



Şekil 5.4: Çekişmeli Üretken Ağlar ile Üretilmiş Rastgele Sandalye Şekilleri

### 5.3 Deney

Deneyde 4 sınıfa ait, toplamda 92 farklı rastgele şekil üretilmiştir. Otomobil sınıfı için 23, sandalye sınıfı için 29, uçak sınıfı için 26 ve kanepeler sınıfı için 14 örnek üretil-





Şekil 5.5: Çekişmeli Üretken Ağlar ile Üretilmiş Rastgele Kanepeler

miştir. Üretilen şekillerin eğitim veri kümesindekilerle aynı olup olmadığını kontrol etmek için de üretilen şeklin küresel derinlik görselleriyle kendisine en yakın eğitim verisinin küresel derinlik görselinin ortalama karesel hatası tespit edilmiştir. Elde edilen en yakın resim çifti hem iki boyutta hem de üç boyutta görsel olarak incelenmiştir. Şekil 5.2’de örnek bir kıyaslama görülmektedir. Üretilen şekillerin çeşitliliği Şekil 5.3, 5.4 ve 5.5’te görülmektedir. Bu çalışmanın deney sonuçlarının daha iyi gözlemlenebilmesi için sonuç görselleri önce bilinear yukarı örnekleme (bilinear upsampling)  $64 \times 64$  çözünürlükten  $256 \times 256$  çözünürlüğe yükseltilmiş, sonrasında nokta bulutuna dönüştürülmüştür.

Şekil 5.6’da üretilen model koltuk olduğu halde çekişmeli üretken ağın sandalye ayağı koyma çabası görülmektedir. ShapeNet [3] eğitim veri kümesinde sandalyeler ve koltuklar aynı sınıf olarak değerlendirildiği için, ikisinin özneliklerinin tek bir üretim sonucunda birlikte görülmesi de bundandır. Bu da, veri kümelerinin amaca yönelik sınıflandırılmasının önemini göstermektedir. Ayrıca, 5.5’de en soldaki kanepenin üst bölümünden yükselen çıkıntı, üretimin rastgeleliğinin bazı örneklerde gürültülü üretime sebep olabildiğini göstermektedir. Şekil genel itibarıyla sınıfa uygun olup ayırt edici ağı kandırabilse de, bölgesel gürültüler çekişmeli üretken ağların çıktılarında görülebilmektedir.

Çizelge 5.1: AZAMI BENZERLİK SONUÇLARI

	Otomobil	Sandalye	Uçak	Kanepe
Ort. Karesel Hata	22.34	48.86	35.81	28.86

## 5.4 Sonuçlar

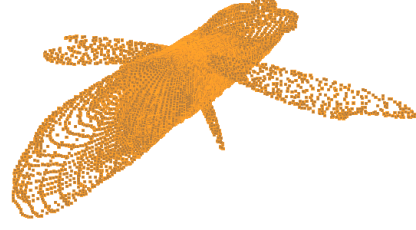
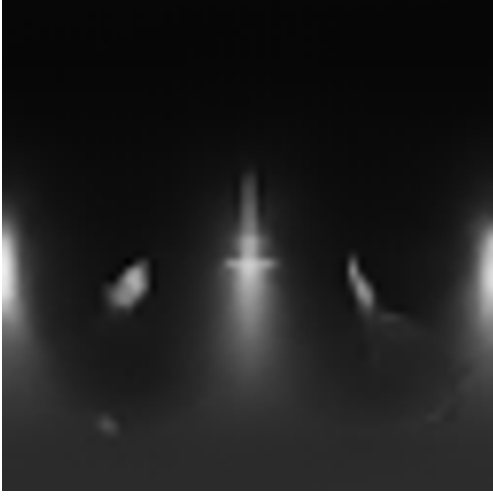
Deney sonucunda görülmüştür ki koşullu çekişmeli üretken ağlar tek katmanlı küresel derinlik parametreleri ile eğitildiği takdirde üretken ağın sonuçları gerçek verilere oldukça yakın sınıflandırılmış veriler üretebilmektedir. Üretilen verilerin sınıflandırılmış veri olmasından ötürü veri kümesinin spesifik örneklere göre daha da daraltılması gerektiği görülmüştür. Bu olmadığı takdirde, üretilen veri üzerinde lokal kontrole sahip olmayan Çekişmeli Üretken Ağlar sınıflandırılmamış semantik alt sınıfların karması çıktı verebileceği için, çıktı şeklinde tutarsızlıklar olması olasıdır, Şekil 5.6 buna güzel bir örnektir. Sonuçlarda üretilen şekillerin eğitim veri kümesinde kendilerine en çok benzeyen şekiller ile farkları, küresel derinlik görselleri üzerinden ortalama karesel hata hesaplanarak bulunmuştur. Bu farkların değerleri, Çizelge 5.1’de görülmektedir. Buna göre, otomobil sınıfında üretilen örneklerin eğitim verileri arasında kendilerine en çok benzeyenlerle farkının ortalama karesel hata cinsinden değeri (piksel başına değer aralığı [0, 255] iken) 22.34, sandalye sınıfının farkı 48.86, uçak sınıfının farkı 35.81 ve son olarak kanepe sınıfının farkı 28.86 olarak bulunmuştur. Bu da, üretilen verilerin kendilerine en çok benzeyen örneklerden bile oldukça uzak olduğunu, gerçekçi oldukları kadar eğitim verisinden uzak ve özgün olduklarını gösterir.

Bu kıyaslamaların yanısıra, üretilen sonuçların ilgili sınıflara benzerliği görsel kıyaslamamanın yanısıra 3B veya 2B evrişimsel sınıflandırıcı sinir ağları ile de yapılabilir. Öte yandan, üretilen şekillerin gerçek hayatta kullanıma uygun olup olmadığı, üretilen mobilya sonuçlarının ergonomik ve kullanışlı olup olmadığı, üretilen taşıt sonuçlarının gerçek hayatta kullanılabilir olup olmadığı da sayısal kıyaslamamanın yanısıra uygulanabilirlik açısından önem arz eden kıyaslamalardır. Gelecek çalışmalarda koşullu derin evrişimsel çekişmeli üretken sinir ağlarının bu sonuçları bu gibi açılardan ele alınıp farklı bağlamlarda da tartışılabilir.



Şekil 5.6: Çekişmeli Üretken Ağlar ile Üretilmiş Hatalı Koltuk Şekli

Üretilen modellerden otomobil, kanep ve sandalye sınıfı için gerçekçi örnekler üretilebilmektedir. Uçak sınıfındaki örneklerin aileron, elevatör, rudder, kanat ve dikey sabitleyici gibi yüzeylerinin oldukça ince olmasından dolayı küresel derinlik parametrelerinin daha yüksek çözünürlükte kullanılması, gelecek çalışmalarda alınacak sonuçların isabet oranının artırılması adına önemlidir. Şekil 5.7 üretilen uçak denemelerinden çözünürlüğün yetersiz olması kaynaklı kanatları ve dikey sabitleyicisi tam üretilememiş bir tanesinin küresel derinlik görseli ve üç boyutlu yeniden üretimini (reconstruction) göstermektedir. Düşük çözünürlüğe rağmen, görselde de görüleceği üzere gövde üretimi başarılıdır. Bunun sebebi, kalın bir yüzey olan gövdenin çözünürlüğün az olması sebebiyle sıkıntı yaşamamasıdır.



Şekil 5.7: Kanat ve Dikey Sabitleyici Üretimi Bozuk ve Eksik Olan Uçak Şekli (3B görüntü Blender [35] programı ile elde edilmiştir)

## 6. Sonuçlar ve Öneriler

Çalışmamızda üç ana başlıkta üç alt çalışma ele alınmıştır. Bunlardan birincisi U-Net benzeri sinir ağı ve Çok Katmanlı Küresel Derinlik Parametreleri kullanarak 3B şekil tamamlama, ikincisi Adaptif Küresel Derinlik Parametreleri kullanarak 3B şekil tamamlama ve üçüncüsü de Koşullu Çekişmeli Üretken Ağlar ve Küresel Derinlik Parametreleri kullanarak özgün 3B şekil üretmedir.

Birinci çalışmada dört farklı sınıf için U-Net benzeri sinir ağı eğitilmiştir. Üç farklı sınıf için (otomobil, sandalye ve kanepeler) makul değerler elde edilmiş ve bu değerler taban doğru değerlerle kıyaslanmıştır. Dördüncü sınıf yani uçak için ise taban doğru değerlerin ilgili çözünürlükte ( $256 \times 256$ ) yeterli düzeyde temsil edilememesinden ötürü kıyaslama mümkün olmamıştır. İlgili kıyaslama için test verisinin çözünürlüğünü  $512 \times 512$  veya  $1024 \times 1024$  gibi değerlere çekmek, çalışmanın ölçeklenebilir (scalable) olması sebebiyle gelecek çalışmalarda kıyaslamayı mümkün kılacaktır. Bunun yerine önem tabanlı örnekleme yaklaşımları ile daha yüksek işlem hızıyla benzer sonuçlar elde edilebilir. Bu çalışmanın sonucunda, üç sınıf için state of the art bir çalışma olan OCNN with Output Guided Skip Connections ile yakın, hatta tüm testlerde ondan daha iyi sonuçlar elde edilmiştir. Bu deneyler, farklı öğrenme oranları ve farklı

eksik nokta yüzdeleri ile denenmiştir. Bunun yanısıra, Kısmi Evrişimli Sinir Ağları ve önem örneklemeli masraf fonksiyonu gibi başka yöntemler de denenmiş fakat en başarılı sonucu U-Net benzeri sinir ağı ve Çok Katmanlı Küresel Derinlik Parametrelerinin, ortalama karesel hata masraf fonksiyonuyla uygulandığı durumda elde edildiği gözlemlenmiştir.

İkinci çalışmada ise birinci çalışmada başarılı olan yöntemi daha az evrişim operasyonu ile yapabilmek için bir optimizasyon denenmiştir. Adaptif Küresel Derinlik Parametreleri yani dörtlü ağaç yapısına bölünmüş Küresel Derinlik Parametreleri ile U-Net benzeri fakat dörtlü ağaç yapısında çoklu girdili ve maskeli evrişim/ortaklama işlemleri uygulayabilen bir sinir ağı eğitilmiştir. Öncelikle modeli doğrulamak (validation) için tek sınıf ile overfit testi yapılmış, modelin girdi ile çıktı görselinin aynı olmasını sağlayıp sağlayamayacağı gözlemlenmiştir. Overfit testi sonucunda elde edilen sonuç neticesinde modelin bazı eklemelere ihtiyaç duyduğu gözlemlenmiştir. OCNN [8] çalışmasındaki gibi deşifre katmanlarında maske tahmini yapan yardımcı sinir ağının kullanılması, elde edilecek sonucun başarı oranını artırabilir. Gelecek çalışmalarda yardımcı bir sinir ağı ile deşifre işleminin denenmesi kayda değer bir yaklaşımdır.

Üçüncü ve son çalışmada ise Koşullu Çekişmeli Üretken Sinir Ağları ve Tek Katmanlı Küresel Derinlik Parametreleri kullanılarak 3B özgün şekil üretimi denenmiş, deneylerin sonucunda görsel olarak estetik ve gerçekçi sınıfsal veriler elde edilirken bu verilerin eğitim kümesindeki verilerden de oldukça farklı olduğu gözlemlenmiş, üretilen şekillerin hem özgün hem gerçekçi olduğu görülmüştür. Üç sınıf için (otomobil, sandalye ve kanepeler) sonuçlar oldukça tatmin ediciyken, uçak sınıfı için test amaçlı kullanılan çözünürlüğün artırılması gerektiği kanısına varılmıştır. Bunun sebebi, ince yüzeylerin  $64 \times 64$  çözünürlükte yeterince hassas temsil edilememesidir. Bununla birlikte, uçak sınıfında da uçağın gövdesinin başarıyla oluşturulduğu ve görsel anlamda gerçekçi, benzerlik anlamında ise özgün olduğu görülmüştür. Gelecek çalışmalarda daha yüksek detay seviyesinde üretim denenebilir. Bunun yanısıra, tek katman yerine Çok Katmanlı Küresel Derinlik Parametreleri'nin sinir ağına tek girdi olarak beslen-

mesi, şekillerde eksik kalan yüzeylerin tamamlanmasını sağlayabilir. Tek katmanın eksik kaldığı noktaları çok katmanlı uygulama, birinci çalışmamızda değinildiği gibi kapatabilir. Bunun için birinci çalışmadan farklı olarak, parametreler üç farklı görsel olarak değil çok katmanlı bir tek görsel olarak ağa sağlanmalıdır. Boyutluluk laneti (curse of dimensionality) olarak bilinen fenomenen kurtulmak için katman bazlı veya yüzeyin detayı tabanlı masraf fonksiyonları uygulanabileceği gibi, daha gelişkin Çekışmeli Üretken Ağlar mimarileri de uygulanabilir. Sonuçların değerlendirilmesi için eğitim veri kümesinde, üretilen örneğe en yakın şeklin ilgili örnekle farkı ölçülmüştür. Bu şekilde, üretilen örneğin eğitim veri kümesindekilerle aynı olmadığı, özgün olduğu gösterilmiştir. Bunun yanısıra, gelecek çalışmalarda, üretilen bu şekillerin gerçek hayata uygulanabilir olup olmadığı, kullanılabilirlik ve ergonomik olup olmadıkları da ele alınabilir, sayısal kıyaslamaların yanısıra kullanılabilirlik açısından da üretilen şekiller amaçlarına göre değerlendirilebilir.

## Kaynakça

- [1] T. Hu, Z. Han and M. Zwicker. 3D Shape Completion with Multi-view Consistent Inference. *AAAI*, New York, pp. 10997–11004, 2020
- [2] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, G. Ranzuglia MeshLab: an Open-Source Mesh Processing Tool *Sixth Eurographics Italian Chapter Conference*, page 129-136, 2008
- [3] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, “ShapeNet: An Information-Rich 3D Model Repository,” Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.
- [4] B. Shi, X. Bai, S. Bai and Z. Zhou. DeepPano: Deep Panoramic Representation for 3-D Shape Recognition *IEEE Signal Processing Letters* 22(12):2339-2343, 2015
- [5] A. Sinha, A. Unmesh, Q. Huang and K. Ramani. SurfNet: Generating 3D shape surfaces using deep residual networks. *CVPR*, Honolulu, pp. 791–800, 2017.
- [6] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation“, in *International Conference on Medical image computing and computer-assisted intervention*, Munich, pp. 234–241, 2015
- [7] S. Fidler, S. Dickinson and R. Urtasun. 3D Object Detection and Viewpoint Estimation with a Deformable 3D Cuboid Model. *NIPS*, Nevada, pp. 620–628, 2012.
- [8] P.-S. Wang, C.-Y. Sun, Y. Liu, and X. Tong. O-CNN: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (SIGGRAPH)* 36(4), 2017
- [9] P.-S. Wang, C.-Y. Sun, Y. Liu, and X. Tong. Adaptive OCNN: A patch-based deep representation of 3D shapes. *ACM Transactions on Graphics*, 37(6), 2018.
- [10] X. Gu, S.J. Gortler, H. Hoppe. Geometry Images. Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, *SIGGRAPH 2002*, pp. 355-361. ACM, New York, 2002.

- [11] W. Wang, Q. Huang, S. You, C. Yang, and U. Neumann. Shape Inpainting Using 3D Generative Adversarial Network and Recurrent Convolutional Networks. Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, Venice, pp. 2298-2306, 2017.
- [12] Z. Cao, Q. Huang, and R. Karthik. 3D Object Classification Via Spherical Projections. in *International Conference on 3D Vision (3DV)*, Qingdao, pp. 566-574, Oct. 2017.
- [13] J. Wu, C. Zhang, T. Xue, B. Freeman, J. Tenenbaum. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. *Advances in Neural Information Processing Systems (NIPS)*, Barcelona, pp. 82–90, 2016
- [14] P.-S. Wang, C.-Y. Sun, Y. Liu, and X. Tong. Deep Octree-based CNNs with Output-Guided Skip Connections for 3D Shape and Scene Completion. *CVPRW*, Seattle, 2020.
- [15] B. Yang, H. Wen, S. Wang, R. Clark, A. Markham, and N. Trigoni, 3D Object Reconstruction From a Single Depth View with Adversarial Learning, in *ICCV WORKSHOP*, Venice, pp. 679–688, 2017.
- [16] X. Zhang, Z. Zhang, C. Zhang, J. B. Tenenbaum, W. T. Freeman and J. Wu, "Learning to Reconstruct Shapes from Unseen Classes" in *NeurIPS*, Montréal, pp. 2263–2274, 2018
- [17] J. Wang and Z. Fang, "GSIR: Generalizable 3D Shape Interpretation and Reconstruction" in *ECCV*, Glasgow, pp. 498–514, 2020
- [18] M. Kolla and A. Kumar, "Manideep2510/eye-in-the-sky." <https://github.com/manideep2510/eye-in-the-sky/>. Online; accessed 22 Feb 2021.
- [19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, Lille, pp. 448–456, 2015.
- [20] P.-S. Wang, C.-Y. Sun, Y. Liu, and X. Tong, "Ao-cnn." <https://wang-ps.github.io/AO-CNN.html>. Online; accessed 03 March 2021.
- [21] A. Ağca and F. B. Atalay, "3D Shape Completion Using Multilayer Spherical Depth Parameters," in *29th Signal Processing and Communications Applications Conference (SIU)*, 2021, pp. 1-4, doi: 10.1109/SIU53274.2021.9477782.
- [22] G. Liu, F.A. Reda, K.J. Shih, T.C. Wang, A. Tao, and B. Catanzaro (2018) Image inpainting for Irregular Holes Using Partial Convolutions. In *ECCV*, 2018.
- [23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.



- [24] Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015)
- [25] Clark, A. (n.d.). python-pillow/Pillow: The FRIENDLY PIL FORK (Python Imaging Library). GitHub. <https://github.com/python-pillow/Pillow>.
- [26] Wang, P. S., Li, J., Pisani, D., Yuan, Y. & Microsoft. (n.d.). microsoft/O-CNN: O-cnn: OCTREE-BASED convolutional neural networks for 3d shape analysis. GitHub. <https://github.com/microsoft/O-CNN>.
- [27] NVIDIA. (2021, May 10). CUDA toolkit. NVIDIA Developer. <https://developer.nvidia.com/cuda-toolkit>.
- [28] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv: Distributed, Parallel, and Cluster Computing, 2015.
- [29] Jayaraman P.K., Mei J., Cai J., Zheng J. (2018) Quadtree Convolutional Neural Networks. In: Ferrari V., Hebert M., Sminchisescu C., Weiss Y. (eds) Computer Vision - ECCV 2018. *ECCV 2018. Lecture Notes in Computer Science*, vol 11210. Springer, Cham. [https://doi.org/10.1007/978-3-030-01231-1\\_34](https://doi.org/10.1007/978-3-030-01231-1_34)
- [30] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. In: *Advances in Neural Information Processing Systems (NeurIPS)*. (2014)
- [31] LeCun Y., Haffner P., Bottou L., Bengio Y. (1999) Object Recognition with Gradient-Based Learning. In: Shape, Contour and Grouping in Computer Vision. *Lecture Notes in Computer Science*, vol 1681. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/3-540-46805-6\\_19](https://doi.org/10.1007/3-540-46805-6_19)
- [32] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. *Neural computation*. 9. 1735-80. 10.1162/neco.1997.9.8.1735.
- [33] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014.
- [34] Odena, et al., "Deconvolution and Checkerboard Artifacts", *Distill*, 2016. <http://doi.org/10.23915/distill.00003>.
- [35] Community, B. O. (2021). Blender - a 3D modelling and rendering package. Stichting Blender Foundation, Amsterdam. Retrieved from <http://www.blender.org>
- [36] A. Radford, L. Metz, and S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434, 2015.

[37] Brownlee, J. (2020, September 1). How to develop a GAN for Generating mnist handwritten digits. Machine Learning Mastery. <https://machinelearningmastery.com/how-to-develop-a-conditional-generative-adversarial-network-from-scratch/>.

