
GAZİ ÜNİVERSİTESİ
Gazi
1926

DERİN ÜRETİCİ AĞLAR İLE ÖLÇEKLENEBİLİR İKİLİ GÖRÜNTÜ OLUŞTURMA VE TEK GÖRÜNTÜDEN ÜÇ BOYUTLU NESNE YAPILANDIRMA

Ceren GÜZEL TURHAN

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

OCAK 2020

Ceren GÜZEL TURHAN tarafından hazırlanan "DERİN ÜRETİCİ AĞLAR İLE ÖLÇEKLENEBİLİR İKİLİ GÖRÜNTÜ OLUŞTURMA VE TEK GÖRÜNTÜDEN ÜÇ BOYUTLU NESNE YAPILANDIRMA" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Ana Bilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Hasan Şakir BİLGE

Elektrik Elektronik Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi ……..……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Başkan: Prof. Dr. İlkay ULUSOY

Elektrik Elektronik Mühendisliği Ana Bilim Dalı, Orta Doğu Teknik Üniversitesi ...……

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Prof. Dr. Hasan OĞUL

Bilgisayar Mühendisliği Ana Bilim Dalı, Başkent Üniversitesi ...……

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Prof. Dr. Suat ÖZDEMİR

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi .....……

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Doç. Dr. Hacer KARACAN

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi ...……

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Tez Savunma Tarihi: 28/01/2020

Jüri tarafından kabul edilen bu çalışmanın Doktora Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

……..…….

Prof. Dr. Sena YAŞYERLİ

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

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

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

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

……..

Ceren GÜZEL TURHAN

28/01/2020

DERİN ÜRETİCİ AĞLAR İLE ÖLÇEKLENEBİLİR İKİLİ GÖRÜNTÜ OLUŞTURMA VE TEK GÖRÜNTÜDEN ÜÇ BOYUTLU NESNE YAPILANDIRMA

(Doktora Tezi)

Ceren GÜZEL TURHAN

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Ocak 2020

ÖZET

Derin ağ konusunda son gelişmeler, görüntü oluşturma, tamamlama, sahne değiştirme gibi bilgisayar görü problemleri için Üretici Çekişmeli Ağ (GAN) ve Otokodlayıcıya (AE) dayalı modellerin ortaya çıkmasına neden olmuştur. Bu modeller incelendiğinde daha kısa süren eğitim süreleri ve maliyetleri nedeniyle genellikle düşük boyutlu görüntüler oluşturabildiği değerlendirilmiştir. Bu nedenle, ölçeklenebilir bir üretici ağ modeli oluşturmak öncelikli olarak hedeflenmiştir. Diğer bir taraftan, üretici modellerin görüntü oluşturma performanslarından etkilenilerek bu modelleri üç boyutlu alana aktarmaya odaklanılmıştır. Gerçek problemler için daha kritik olan görüntülerden nesne oluşturma ve yeniden yapılandırma problemi ele alınmıştır. Gerçek nesnelerin üç boyutlu yer gerçekliği verilerinin elde edilmesinin güçlüğü nedeniyle sentetik veriler üzerinde eğitilen modelleri gerçek veriler üzerinde de kullanabilmek üzere RGB görüntüler yerine silüet tabanlı çalışmalar yürütülmüştür. Nesnelerin birden fazla açıdan çekilmiş görüntülerinin her zaman mevcut olamaması nedeniyle ise tek açıdan görüntülere dayalı kategori-bağımsız modeller benimsenmiştir. Tez kapsamında, ilk olarak, VAE/CPGAN ölçeklenebilir bir üretici ağ modeli oluşturmak üzere önerilmiştir. Önerilen model ile ikili görüntülerde istenen boyutlarda görüntülerin, düşük boyutlu görüntülerden elde edilebildiği görülmüştür. Tez kapsamında devam eden çalışmalarda tek açıdan görüntülerden nesne yapılandırmak üzere önerilen VoxCAE/GAN, VoxAE, VoxCAE, SkipVoxCAE ve FusedVoxCAE modelleri, literatürdeki diğer çalışmalardan farklı olarak, türevlenebilir olarak tanımlanan Bileşim üzerinde Kesişim (IoU) maliyetine dayalı olarak eğitilmiştir. Literatürde daha önce nesne yapılandırma için kullanılan amaç fonksiyonları ile analiz çalışmaları yürütülmüştür. Gerçekleştirilen niteliksel ve niceliksel değerlendirmelere göre, tez kapsamında önerilen IoU maliyetine dayalı eğitilen modellerin daha iyi performans sergilediği görülmüştür. Adım adım iyileştirilen modeller ile önde gelen çalışmalara benzer, bazı kategoriler için ise daha iyi sonuçların elde edilebildiği ortaya koyulmuştur.

Bilim Kodu	:	92431
Anahtar Kelimeler	:	Üç boyutlu nesne yapılandırma, Tek açıdan görüntülerden nesne yapılandırma, Üretici çekişmeli ağlar, Otokodlayıcılar, Bileşim üzerinde Kesişim maliyeti
Sayfa Adedi	:	125
Danışman	:	Doç. Dr. Hasan Şakir BİLGE

SCALABLE BINARY IMAGE GENERATION AND SINGLE IMAGE TO THREE-DIMENSIONAL OBJECT RECONSTRUCTION USING DEEP GENERATIVE MODELS

(Ph. D. Thesis)

Ceren GÜZEL TURHAN

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

January 2020

ABSTRACT

Recent improvements on deep generative models have revealed Generative Adversarial Network (GAN) and Autoencoder (AE) based models for image generation, completion, inpainting, and similar computer vision tasks. These models are capable of generating low-dimensional images due to the computational costs. Therefore, it has been addressed to develop a scalable GAN model. Furthermore, the performance of the current generative models has led to studies on transferring these models to three dimensional domain. Image to object reconstruction problem has been considered to be more critical for real-world problems. Differences among synthetic and real images have caused silhouette based studies rather than RGB. The category-agnostic modeling using single image has been targeted in the rest of studies due to the difficulties on obtaining multiple images of an object. In the thesis study, first of all, VAE/CPGAN has been proposed as a scalable generative model. It has been seen the desired sized images can be generated from low-dimensional images. In the following studies of thesis, proposed VoxCAE/GAN, VoxAE, VoxCAE, SkipVoxCAE and FusedVoxCAE models have been trained depending on given differentiable the Intersection-Over-Union (IoU) objective unlike previous studies in the literature. The contribution of given objective on the model performance has been analyzed comparing existing objectives for three dimensional object reconstruction. According to given qualitative and quantitative results, it has been shown that proposed models based on IoU objective can perform better than compared objectives. By improving the performances of presented models step by step, promising results have been recorded when comparing the outstanding studies.

Science Code	:	92431
Key Words	:	Three-dimensional object reconstruction, Single image to object reconstruction, Generative adversarial networks, Autoencoders, Intersection-Over-Union objective
Page Number	:	125
Supervisor	:	Assoc. Prof. Dr. Hasan Şakir BİLGE

TEŞEKKÜR

Doktora çalışmalarım süresince değerli katkılarıyla beni yönlendiren danışman hocam Doç. Dr. Hasan Şakir BİLGE, tez izleme jüri hocalarım Prof. Dr. Suat ÖZDEMİR ve Prof. Dr. İlkay ULUSOY'a ve kıymetli tecrübelerinden faydalandığım hocam Dr. Öğr. Üyesi Fatih NAR'a teşekkür ederim.

Bu süreçte yanımda olan ve desteklerini hiçbir zaman esirgemeyen kıymetli eşim ve anneme borcumu hiçbir zaman ödeyemem. Doktora tez çalışmasını hayat arkadaşım Meftun'a ve bu süreçte hayatımıza giren kızımız İpek'e ithafen sunuyorum.

Bu tez çalışması Tubitak 2211-E Doğrudan Yurt İçi Doktora programı tarafından desteklenmiştir.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

RESİMLERİN LİSTESİ

Resim **Sayfa**

Resim **Sayfa**

SİMGELER VE KISALTMALAR

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

Simgeler	Açıklamalar
$\boldsymbol{a^i}$	Yapay sinir ağı i.ci düğüm
$\boldsymbol{a_i^{(l)}}$	l katman i düğümü aktivasyonu
$\boldsymbol{b}$	Bias parametresi
$\boldsymbol{b^{(l)}}$	l katmanı bias parametresi
$\boldsymbol{b_i^{(l)}}$	l katmanı i düğümü bias ağırlık parametresi
$\boldsymbol{c}$	Veri etiketi
$\boldsymbol{f(.)}$	Bir fonksiyon, aktivasyon fonksiyonu
$\boldsymbol{f_l}$	Yüksek-seviyeli öznitelikler
$\boldsymbol{f^*}$	Öğrenilen fonksiyon
$\boldsymbol{log}$	Logaritma
$\boldsymbol{p}$	Olasılık yoğunluk fonksiyonu, kodçözücü dağılımı
$\boldsymbol{p_{data}}$	Veri dağılımı
$\boldsymbol{p_G}$	G fonksiyonu ile öğrenilen dağılım
$\boldsymbol{p_z}$	Bilinen gizli kod dağılımı
$\boldsymbol{q}$	Kodlayıcı model dağılımı
$\boldsymbol{q_n}$	n sınıfa ait softmax düzenlileştirme fonksiyonu
$\boldsymbol{r}$	Merkezden uzaklık
$\boldsymbol{r_{vec}}$	Merkezden uzaklık vektörü
$\boldsymbol{s}$	Ölçek
$\boldsymbol{s_l}$	l katmanı düğüm sayısı
$\boldsymbol{n}$	Sınıf sayısı
$\boldsymbol{n_l}$	Katman sayısı
$\boldsymbol{w}$	Fonksiyon parametresi, ağırlıklar
$\boldsymbol{w_i}$	i.ci düğüm ağırlığı, sinapsis
$\boldsymbol{x}$	Giriş verisi, giriş görüntüleri, x koordinatı
$\boldsymbol{x_{coord}}$	x koordinat vektörü

Simgeler	Açıklamalar
$\boldsymbol{x_i}$	i.ci giriş görüntüsü
$\boldsymbol{x^i}$	i.ci giriş görüntüsü
$\boldsymbol{\hat{x}}$	Yeniden yapılandırma görüntüleri
$\boldsymbol{(x^{(i)}, y^{(i)})}$	i.ci eğitim örneği
$\boldsymbol{y}$	Çıkış verisi, y koordinatı
$\boldsymbol{y_{coord}}$	y koordinat vektörü
$\boldsymbol{y_{ijk}}$	Orijinal nesne
$\boldsymbol{\hat{y}}$	Hipotez çıktısı, yeniden yapılandırılan nesneleri
$\boldsymbol{\hat{y}_{ijk}}$	Yeniden yapılandırılan nesne
$\boldsymbol{z}$	Gizli kod vektörü
$\boldsymbol{z_i^{(l)}}$	l katmanı i düğümü ağılıklı toplam vektörü
$\boldsymbol{z_\mu}$	Ortalama gizli kod vektörü
$\boldsymbol{z_p}$	Normal dağılımdan örneklenen gizli kod vektörü
$\boldsymbol{z_\sigma}$	Standart sapma gizli kod vektörü
$\boldsymbol{t}$	Eşik değeri
$\boldsymbol{\alpha}$	Öğrenme oranı parametresi
$\boldsymbol{\beta}$	Eniyileme parametresi
$\boldsymbol{\lambda}$	Ağırlık, maliyet fonksiyonu parametresi
$\boldsymbol{\lambda_{th}}$	Eşik değer parametresi
$\boldsymbol{\mu}$	Ortalama
$\boldsymbol{\sigma}$	Standart sapma
$\boldsymbol{D}$	Kodçözücü/Ayrımcı ağ
$\mathbb{D}_{\boldsymbol{KL}}$	Kullback-Leiber (KL) ıraksama uzaklığı
$\boldsymbol{E}$	Kodlayıcı ağ
$\mathbb{E}$	Beklenen değer
$\boldsymbol{G}$	Üretici ağ
$\boldsymbol{I}$	İndikatör fonksiyonu
$\mathbb{N}$	Normal dağılım
$\boldsymbol{L}$	Ağ katmanı
$\boldsymbol{L_d}$	Ayrımcı ağ maliyet fonksiyonu

Simgeler	Açıklamalar
L_d^{dis}	Öznitelik tabanlı maliyet fonksiyonu
L_{fea}	Öznitelik tabanlı maliyet fonksiyonu
L_g	Üretici ağ maliyet fonksiyonu
L_d^{GAN}	Ayrımcı ağ için çekişmeli maliyet fonksiyonu
L_g^{GAN}	Üretici ağ için çekişmeli maliyet fonksiyonu
L_{AE}	Otokodlayıcı maliyet fonksiyonu
L_{CPGAN}	Kompozisyonel üretici ağ maliyet fonksiyonu
L_{GAN}	Üretici çekişmeli ağ maliyet fonksiyonu
L_{IoU}	IoU maliyet fonksiyonu
L_{pixel}	Piksel tabanlı yeniden yapılanma maliyet fonksiyonu
L_{prior}	Bilinen maliyet fonksiyonu
L_{VAE}	Değişimsel otokodlayıcı maliyet fonksiyonu
θ_d	Kodçözücü, ayrımcı ağ parametreleri
θ_e	Kodlayıcı parametreleri
θ_g	Üretici parametreleri
L_g^{GAN}	Üretici ağ için çekişmeli maliyet fonksiyonu
L_{AE}	Otokodlayıcı maliyet fonksiyonu
L_{GAN}	Üretici çekişmeli ağ maliyet fonksiyonu
L_{IoU}	IoU maliyet fonksiyonu
θ_d	Kodçözücü parametreleri
θ_e	Kodlayıcı parametreleri
θ_g	Üretici parametreleri
W	f fonksiyonu parametresi
$W_{ij}^{(l)}$	l ve $l+1$ katmanı i ve j düğümleri arasındaki ağırlık
$W^{(l)}$	l. katman ağırlık parametreleri

Kısaltmalar	Açıklamalar
conv	Evrişim (convolution) katmanı
deconv	Ters evrişim (deconvolution) katmanı

Kısaltmalar	Açıklamalar
fc	Tam-bağlı (fully-connected) katman
f-GAN	f-ıraksamalı Üretici Çekişmeli Ağ (f-divergence Generative Adversarial Network)
mlpconv	Evrişim ve tam bağlı katman
AAE	Çekişmeli Otokodlayıcı (Adversarial AutoEncoder)
AE	Otokodlayıcı (AutoEncoder)
AE-GAN	Otokodlayıcı Üretici Çekişmeli Ağ
BiGAN	Çift-yönlü Üretici Çekişmeli Ağ (Bidirectional Generative Adversarial Network)
CAD	Bilgisayar Destekli Tasarım (Computer-aided Design)
CAE	Koşullu Otokodlayıcı
CGAN	Koşullu Üretici Çekişmeli Ağ (Conditional Generative Adversarial Network)
CIFAR	İleri Araştırmalar için Kanada Enstitüsü (Canadian Institute for Advanced Research)
CNN	Evrişimsel Sinir Ağı (Convolutional Neural Networks)
CNN-S	Yavaş Evrişimsel Sinir Ağı
CNN-M	Orta Evrişimsel Sinir Ağı
CNN-F	Hızlı Evrişimsel Sinir Ağı
CPPN	Kompozisyonel Örüntü Üretici Ağ (Compositional Pattern Producing Network)
CPPN-VAE/GAN	Kompozisyonel Örüntü Üretici Değişimsel Otokodlayıcı Ağ
CVN	Renkli Voksel Ağı
DAE	Gürültü Otokodlayıcı (Denoising AutoEncoder)
DBM	Derin Boltzmann Makinesi (Deep Boltzmann Machine)
DBN	Derin İnanç Ağı (Deep Belief Network)
DCGAN	Derin Evrişimsel Üretici Çekişmeli Ağ (Deep Convolutional Neural Network)
DeconvNet	Ters Evrişimsel Ağ

Kısaltmalar	Açıklamalar
DFNN	Derin İleri beslemeli Sinir Ağı (Deep Feed forward Neural Network)
DRAW	Derin Tekrarlayan Dikkatli Yazar Ağı
DNN	Derin Sinir Ağı (Deep Neural Network)
DVAE	Gürültü Giderici Değişimsel Otokodlayıcı
ELU	Üstel Doğrusal Birim (Exponential Linear Unit)
Fast R-CNN	Hızlı Bölgesel Evrişimsel Sinir Ağı (Fast Region-based Convolutional Network)
Faster R-CNN	Daha Hızlı Bölgesel Evrişimsel Sinir Ağı (Faster Region-based Convolutional Network
FusedVoxCAE	Füzyon edilmiş Voksel Koşullu Otokodlayıcı (Fused Voxel Conditional AutoEncoder)
FFD	Serbest biçimli deformasyon (Free-Form Deformation)
GAL	Geometrik Maliyet Fonksiyonu
GAN	Üretici Çekişmeli Ağ (Generative Neural Network)
GAN-INT-CLS	Manifold İnterpolasyon eşleştirme farkındalıklı Ayrımcılı Üretici Çekişmeli Ağ
GDAE	Genelleştirilmiş Gürültüsüz Otokodlayıcı
GRAN	Üretici Tekrarlayan Çekişmeli Ağ
GSN	Üretici Stokastik Ağ
GoogLeNet	Google Ağı
HMDB51	İnsan Hareketleri Tanıma Veri Tabanı (Human Motion Database)
ILSVRC	ImageNet Büyük Ölçekli Görsel Tanıma Yarışması (ImageNet Large Scale Visual Recognition Competition)
Improved GAN	Geliştirilmiş Üretici Çekişmeli Ağ
InfoGAN	Bilgi Maksimize Üretici Çekişmeli Ağı
IoU	Bileşim üzerinde Kesişim (Intersection over Union)
IS	İnception Skoru (Inception Score)
KL	Kullback Leibler

Kısaltmalar	Açıklamalar
Leaky ReLU	Sızan Düzeltilmiş Doğrusal Birim
LRCN	Uzun Vadeli Tekrarlayan Evrişimsel Ağlar (Long-term Recurrent Convolutional Networks)
LRN	Lokal Yanıt Normalizasyonu (Local Response Normalization)
LSTM	Uzun-Kısa Vadeli Hafıza (Long Short-Term Memory)
L1	Manhattan uzaklığı
L2	Öklid uzaklığı
MADE	Dağılım Tahmini için Maskelenmiş Otokodlayıcı (Masked Autoencoder for Distribution Estimation)
MCMC	Markov Zinciri Monte Carlo
MCP	McCulloch–Pitts
MNIST	Karışık Ulusal Standartlar ve Teknoloji Enstitüsü (Mixed National Institute of Standards and Technology)
MLP	Çok Katmanlı Perseptron (Multi Layer Perceptron)
MsCOCO	Microsoft İçerikte Ortak Objeler Veri Kümesi
MSE	Ortalama Kareler Hatası (Mean Square Error)
MVD	Çok Açıdan Ayrıştıma modeli
MV-BiGAN	Çok Açıdan Çift-yönlü Üretici Çekişmeli Ağ
NIN	Ağ içinde Ağ (Network in Network)
OCR	Optik Karakter Tanıma
OGN	Octree Üretici Ağı
ODM	Ortografik Derinlik Haritası (Orthographic Depth Map)
PCL	Nokta Bulutu Kütüphanesi (Point Cloud Library)
PixelCNN	Piksel Evrişimsel Sinir Ağı
PixelCNN++	Geliştirilmiş Piksel Evrişimsel Sinir Ağı
PixelRNN	Piksel Tekrarlayan Sinir Ağı
PixelVAE	Piksel Değişimsel Otokodlayıcı
PrGAN	Projeksiyon Üretici Çekişmeli Ağ
PTN	Perspektif Dönüştürücü Ağ

Kısaltmalar	Açıklamalar
RBM	Kısıtlı Boltzmann Makinesi (Restricted Boltzmann Machine)
RCNN	Tekrarlayan Evrişimsel Sinir Ağı (Recurrent Convolutional Neural Network)
ReLU	Düzeltilmiş Doğrusal Birim (Rectified Linear Unit)
ResNet	Artık Sinir Ağı (Residual Network)
RGB	Kırmızı Yeşil Mavi
RNN	Tekrarlayan Sinir Ağı (Recurrent Neural Netwok)
RvNN	Öztekrarlayan Sinir Ağı (Recursive Neural Network)
R-CNN	Bölge Evrişimsel Sinir Ağı (Region-based Convolutional Neural Network)
SAE	Yığılmış Otokodlayıcı (Stacked AutoEncoder)
SCE	Softmax Çapraz Entropi (Softmax Cross Entropy)
SÇ	Süper Çözünürlük
SDAE	Yığılmış Gürültü Otokodlayıcı (Stacked Denoising AutoEncoder)
SGAN	Yığılmış Üretici Çekişmeli Ağı
SkipVoxCAE	Ara-bağlantılı Voksel Koşullu Otokodlayıcı (Skipped Voxel Conditional AutoEncoder)
SS-VAE	Yarı Gözetimli Değişimsel Otokodlayıcı
SVHN	Sokak Görünümünde Ev Numaraları (Street View House Numbers)
UCF101	Merkez Florida Üniversitesi 101 Veri kümesi
VAE	Değişimsel Otokodlayıcı (Variational AutoEncoder)
VAE/CPGAN	Değişimsel Otokodlayıcı ile Kompozisyonel Çekişmeli Üretici Ağ
VAE/GAN	Değişimsel Otokodlayıcı ile Çekişmeli Üretici Ağ
VConvDAE	Volümetrik Evrişimsel Gürültü Giderici Otokodlayıcı
VGG	Görsel Geometri Grubu (Visual Geometry Group) ağı
VGG16	Görsel Geometri Grubu 16 katmanlı ağı
VGG19	Görsel Geometri Grubu 19 katmanlı ağı

Kısaltmalar	Açıklamalar
VLAE	Değişimsel Kayıplı Otokodlayıcı (Variational Lossy AutoEncoder)
VRN	Voxception Artık Ağı (Voxception-ResNet)
VoxCAE/GAN	Voksel tabanlı Koşullu Otokodlayıcı ile Çekişmeli Üretici Ağ
VQA	Görsel Soru Yanıtlama (Visual Question Answering)
WAE	Wasserstein Otokodlayıcı (Wasserstein Autoencoder)
WGAN	Wasserstein Üretici Çekişmeli Ağ (Wasserstein Generative Adversarial Network)
WGAN-GP	Gradyan Cezalı Wasserstein Üretici Çekişmeli Ağ
1B	Bir Boyutlu
2B	İki Boyutlu
2,5B	İki Buçuk Boyutlu
3B	Üç Boyutlu
3DGAN	Üç Boyutlu Üretici Çekişmeli Ağ
3D-INN	Üç Boyutlu Yorumlayıcı Ağ (3D Interpreter Network)
3DIWGAN	Üç Boyutlu Geliştirilmiş Wasserstein Üretici Çekişmeli Ağ
3DR2N2	Üç Boyutlu Tekrarlayan Yeniden Yapılandırma Yapay Sinir Ağı
3D-VAE/GAN	Üç Boyutlu Değişimsel Otokodlayıcı Üretici Çekişmeli Ağ
β-VAE	Beta Değişimsel Otokodlayıcı

1. GİRİŞ

Motivasyon

Makine öğrenmesi; sosyal ağlarda içerik filtreleme, web arama ve daha birçok günümüz modern gereksinimlerini sağlayan teknoloji olarak üzerinde uzun yıllardır çalışılmaktadır. Makine öğrenmesi yaklaşımları; sınıflandırma, kümeleme, regresyon veya boyut indirgeme problemlerini çözmeye odaklanmaktadır. Etiketli veriler mevcut olduğunda bu problemleri çözmek üzere gözetimli öğrenme gerçekleştirilirken, verilerin sınıf bilgisi hakkında bir bilgi söz konusu olmadığında ise gözetimsiz öğrenme yaklaşımları uygulanmaktadır. Günümüz uygulamalarında, daha önceden elde edilen, el ile çıkarılan öznitelikler (hand-craftted features) üzerinde çalışan doğrusal sınıflandırıcılar yaygın olarak kullanılmaktadır. Bu sınıflandırıcılar, girdi uzayı bir hiper düzlem ile alt uzaylara ayrılabildiğinde başarılı olabilmekteyken daha karmaşık yapıdaki verilerde benzer performansı sergileyememektedirler. Derin öğrenme modellerinden önce kullanılan sığ sınıflandırıcılar (shallow classifiers) ise bir öznitelik çıkarıcı modele gereksinim duymaktadır.

Geleneksel makine öğrenmesi yöntemlerinin ham veriyi işlemede yetersiz olmaları ve bir uzman tarafından tanımlanmış özniteliklere gereksinim duymaları gibi problemleri söz konusudur [1]. Bu gereksinimler, betimleme öğrenme olarak adlandırılan yaklaşımların yaygınlaşmasına neden olmuştur. Betimleme öğrenme ham verinin işlenerek betimlemelerin otomatik olarak çıkarılmasına imkan vermektedir. Derin öğrenme, bir betimleme öğrenme yaklaşımı olarak çok seviyede betimlemeler içermektedir. Derin öğrenme ile bir seviyedeki betimlemelerden daha soyut seviyedeki yüksek betimlemelere dönüşüm söz konusudur.

Makine öğrenmesi yaklaşımları olarak ele alınan derin öğrenme kavramının temelini oluşturan yapay sinir ağı yaklaşımları daha önce birçok problemin çözümünde başarı ile kullanılmıştır. Yapay sinir ağlarının temeli, 1943 yılında Warren S. McCulloch ve Walter Pitts tarafından sunulan çalışmada [2] nöronların (MCP) aktivasyonu ile mantıksal ifadeler tanımlayabilen yapay nöron modeli ile atılmıştır. 1958 yılında, Frank Rosenblatt [3] ilk modern yapay sinir ağı modeli olarak tek katmanlı perseptron modelini sunmuştur. Bu modelde, MCP nöron modelinden farklı olarak, yapay nöronların gözetimli öğrenme ile verilerden öğrenilebileceği ortaya koyulmuştur. 1959 yılında, David H. Hubel ve Torsten

Wiesel [4], ana görsel korteks üzerinde bulunan basit ve karmaşık hücreler olarak adlandırılan iki çeşit hücreyi keşfetmişlerdir. Bu hücrelerden esinlenerek yapay sinir ağı olarak adlandırılan çok seviyeli yaklaşım ortaya çıkmıştır. Paul Werbos 1974 yılında tamamladığı doktora tez çalışmasında [5], geri-besleme ile yapay sinir ağlarını eğitilebileceğini göstermiştir. 1980 yılında, Dr. Kunihiko Fukushima, Neocognitron [6] olarak adlandırdığı görsel örüntüleri tanıyabilen hiyerarşik çok katmanlı yapay sinir ağını önermiştir. Bu yıllarda el ile çıkarılan özniteliklerin eğitilebilir çok katmanlı ağlarla yer değiştirilmesi amaçlansa bile bu hedef gerçekleştirilememiştir. 1980 ve 1990'lı yıllarda yapay sinir ağları ve geri-besleme makine öğrenmesi konusunda çalışan araştırmacıların ilgisini çekerken henüz bilgisayarla görme alanlarında çalışan araştırmacıların dikkatini çekmemiştir. 1988 ve 1993 yılları arasında Yann LeCun, derin evrişimsel ağlar (Convolutional Neural Networks) (CNN) üzerine çalışmalarını [7, 8] gerçekleştirmiştir. El yazısı ile yazılan posta kodlarını tanıyabilen derin ağ [7] geri-besleme ile eğitilebilmiştir.

Derin evrişimsel ağların eğitiminin oldukça uzun sürmesi sebebiyle, bu çalışmaları takiben, farklı CNN tabanlı uygulamalar uzun yıllar geliştirilememiştir. 2005 yılı itibariyle daha uygun fiyatlı ve yüksek kapasiteli Grafik İşlem Birimlerinin (GPU) üretilmesi sayesinde 10-20 kat kısalabilen eğitim süreleri ile derin öğrenmeye dayalı çalışmaların önü açılmıştır. Artan hesaplama gücü ve veriler ile birlikte Yann LeCun ve Geoffrey Hinton öncülüğünde yapay sinir ağı çalışmalarına devam edilebilmiştir. Ses tanıma problemini çözmek üzere, İleri Araştırmalar için Kanada Enstitüsü (Canadian Institute for Advanced Research) (CIFAR) araştırmacıları tarafından 2009 yılında önerilen derin öğrenmeye dayalı yapay sinir ağı modeli [9] ile doğruluk oranı bakımından en yüksek başarı elde edilmiş, önerilen bu model literatürde oldukça ses getirmiştir. 2012 yılında ImageNet Büyük Ölçekli Görsel Tanıma Yarışmasında (ImageNet Large Scale Visual Recognition Competition) (ILSVRC) bir CNN ağı [10] ile en yüksek performansın elde edilmesi bilgisayarla görü ve makine öğrenmesi camiasının bu konuya yönelmesine neden olmuştur. Devam eden yıllardaki ImageNet yarışmalarında ilk 10'da performans gösteren modellerin, ağırlıklı olarak CNN tabanlı modeller olduğu dikkat çekmiştir [1].

Evrişimsel sinir ağı tabanlı çeşitli ticari uygulamalar da piyasada yer edinmiştir. CNN tabanlı bir banka çeki okuma sistemi AT&T şirketi için 1990'ların başında geliştirilmiştir. 1993 yılında Avrupa ve Amerika'da ATMler için ilk kez ticarileştirilen çek okuma sistemi, 1996 yılında ise çek makinelerine uygulanmıştır [11]. Bu gelişmeler ile birlikte, Microsoft CNN

tabanlı optik karakter tanıma (OCR) ve el yazısı tanıma tabanlı sistemler [12, 13] üzerine odaklanmıştır. Google güvenlik perspektifi ile StreetView [14] görüntülerindeki yüz ve araç plaka tespiti üzerine CNN modelini sunmuştur [15]. NEC, Vidient teknoloji, France Telecom gibi firmalar; CNN tabanlı sistemler ile müşteri takibi, video izleme, yüz tespiti [16] gibi uygulamalar sunmuştur [17]. NVIDIA, Intel, Samsung gibi çok büyük firmalar gerçek zamanlı görüntü işleme uygulamalarını gerçekleştirebilmek üzere Evrişimsel Sinir Ağı (Convolutional Neural Networks) (CNN) çipler üzerine çalışmaktadır. Google, Microsoft, Facebook ve Yahoo ise CNN tabanlı servisler sunmak üzere çalışmalarını sürdürmektedir [1].

CNN ve Tekrarlayan Sinir Ağı (Recurrent Neural Network) (RNN) gibi ayrımcı modellerin (discriminative models) koşullu olasılık tabanında örnekleri ayırt etmek üzere yüksek-seviyeli betimlemeler (high-level representations) öğrenmedeki başarılarının aksine bir dağılımdan örnekleme yapmada etkili olamamaktadır. Son yıllardaki çalışmalarda, ayrımcı modeller olarak ele alınan CNN modellerini, üretici modeller (generative models) takip etmiştir. Üretici modeller, esas olarak örneklerin üretilmesine odaklanırken; ayrımcı modeller, her sınıf için eğitim örneklerini ayırabilmek üzere bir karar doğrusu/düzlemi bulmaya çalışmaktadır.

Resim 1.1. Orijinal görüntüden elde edilen çekişmeli görüntü örneği

Üretici modeller olarak ele alınan öncü modeller verilerin yeniden elde edilebilmesi amacıyla kullanılmış iken çekişmeli eğitim yaklaşımının ortaya çıkması ile üretici modeller yeni bir soluk kazanmıştır. Çekişmeli eğitim (adversarial learning), çekişmeli örnekler olarak adlandırılan sentetik örnekler ile eğitim yapmaya dayanan yaklaşımdır. Pratikte mevcut olmayan çekişmeli örnekler ile eğitilmiş ağların daha gürbüz ve iyi performanslı olacağı öngörülmüştür [18]. Çekişmeli örnekler, orijinal örnekler gürültü eklenerek manipüle edilmiş veriler olarak tanımlanabilir. Bu manipüle işlemi gözle kolaylıkla ayırt edilmese bile makine öğrenmesi yaklaşımlarına göre çekişmeli örnek farklı sınıfa ait bir örnek olarak

değerlendirilebilmektedir. Resim 1.1'de ImageNet [19] veri kümesinde alınan orijinal bir örnek ile o örnekten elde edilen çekişmeli örnek verilmiştir. Basit bir sınıflandırıcı ile şekilde verilen orijinal örneğin "Panda" olarak, çekişmeli örneğin ise "Gibbon" olarak sınıflandırıldığı görülmüştür [20].

Derin üretici ağ modeli olarak, Üretici Çekişmeli Ağ (Generative Neural Networks) (GAN) [21], çekişmeli eğitime dayalı bir yapay sinir ağı modeli olarak derin öğrenme konusuna yeni bir soluk getirerek oldukça ses getirmiştir. Derin öğrenme alanın öncü araştırmacılarından Yann LeCun, GAN ve türevi modelleri makine öğrenmesi konusundaki son 10 yılın en ilginç fikri olarak nitelendirmiştir [22]. GAN, çekişmeli örnekler ile gerçek örnekleri ayırmak üzere bir sınıflandırıcı olan ayrımcı ağ (discriminative network) (D) ile çekişmeli örnekler oluşturabilen üretici ağdan (generative network) (G) oluşmaktadır. Bu ağ modeli ile üretici ağdan elde edilen gerçekçi sentetik veriler ile ağın, sentetik verileri gerçek verilerden ayırt edebilecek düzeyde eğitilerek sentetik verilere karşı gürbüz olması planlanmıştır. Gerçek ve sentetik verileri ayırt edebilmek üzere eğitilen ayrımcı model ile hem üst seviye öznitelikler elde edilebilmekte hem de sentetik yeni örnekler model ile öğrenilen veri dağılımından örneklenebilmektedir.

GAN modelinde, çekişmeli eğitim ile daha güçlü bir model elde edilirken, gerçekçi fakat sentetik örnekler de elde edilerek bu örnekler ile gerçek örnekleri ayırt etmek üzere eğitilen ayrımcı modellerin performansı da arttırılır. GAN modelleri, klasik modeller ile elde edilen düşük seviyeli öznitelikler ile birlikte yüksek seviyeli öznitelikleri elde ederek veri yapısı ve dağılımını modelleyebilmeleri ve öğrenilen veri dağılımından örnekleme yapılarak sentetik fakat gerçekçi örnekler elde edilebilmesine olanak tanımaları nedeniyle son zamanlarda araştırmacıların dikkatini çekmektedir.

GAN tabanlı modellerin çıkarım mekanizmasından yoksun olması nedeniyle limitli üretim kapasitesi bu modellerin Otokodlayıcı (AutoEncoder) (AE) [23] olarak bilinen gözetimsiz öğrenmeye dayalı yeniden yapılandırma (reconstruction) modelinin kodlayıcı kısmı ile bir araya getirilerek uygulanmasına neden olmuştur. AE modeline dahil edilen bir değişim bileşeni ile Değişimsel Otokodlayıcı (Variational AutoEncoder) (VAE) [24] modeli oluşturularak standart otokodlayıcıların çıkarım kabiliyeti geliştirilmiştir. VAE ve GAN modellerinin avantajlarını bir araya getirmek üzere ise hibrit modeller geliştirilmiştir [25-28].

Üretici modeller; sentetik fakat gerçekçi örnekler oluşturmaları ile birlikte verilerin tanımlı niteliklere göre modifiye edilmesi [29-31], cümle ve kelimeleri tanımlayan örneklerin/görüntülerin oluşturulması [32, 33], eksik görüntülerin tamamlanması [34-36], görüntülerdeki sahne bilgisi/arkaplanın değiştirilmesi [37-39] gibi uygulama alanlarına sahiptir.

İki boyutlu görüntüler ile bilgisayarlı görü konusunda oldukça önemli gelişmeler kaydedilmiş iken bu görüntülerden derinlik olarak da ifade edilen geometri bilgisinin sağlanamadığı görülmüştür. Üç boyutlu tarayıcılar ve hesaplama cihazları konusunda son yıllardaki gelişmeler düşük maliyetli Microsoft Kinect ve Asus Xtion gibi ticari üç boyutlu tarayıcılar ile açık kaynak kodlu nokta bulutu kütüphanelerinin (Point Cloud Library) (PCL) [40] üç boyutlu bilgisayarlı görü problemleri için kullanılmasına neden olmuştur [41]. Robotik, biyometri, uzaktan algılama (remote sensing), yapı bilimi, eğlence ve medikal tedavi gibi çeşitli alanlarda üç boyutlu nesneler üzerinde çalışılmaktadır. Nesnelerin, şekil ve yapılarının çıkarılarak yeniden elde edilmesi işlemi olan üç boyutlu (3B) yeniden yapılandırma ise robotik, bilgisayar destekli tasarım (Computer-Aided Design) (CAD), sanal gerçeklik ve arttırılmış gerçeklik konuları için ele alınan önemli bir üç boyutlu bilgisayarlı görü problemi olmuştur [42]. Görüntü tabanlı üç boyutlu yeniden yapılandırmada ise bir ya da daha çok görüntü şeklindeki sahnelerden nesnelerin üç boyutlu olarak elde edilmesi hedeflenmektedir. Robot navigasyonu, nesne tanıma ve sahne anlama, üç boyutlu modelleme ve animasyon, endüstriyel kontrol ve medikal tanılama uygulamaları için görüntülerden nesne oluşturma ve yeniden yapılandırma problemi önem kazanmıştır [43]. Son yıllarda oldukça önem kazanan üretici ağ modelleri ile görüntü oluşturma ve yeniden yapılandırma problemi için önemli sonuçlar elde edilerek çok sayıda iki boyutlu üretici ağ modeli [24, 29-31, 44-49] ortaya çıkmıştır. Bu modellerin 3 boyut uygulamalarının son yıllarda önemli derecede hızlanarak, görüntüden nesne oluşturma ve yeniden yapılandırma alanının hızla gelişmekte olduğu dikkat çekmektedir.

Problem

Üretici ağ modellerindeki çalışmalar ile ortaya çıkan AE, GAN tabanlı modeller ve bu modellerin avantajlarını birlikte kullanabilmek üzere önerilen hibrit modeller olan Otokodlayıcı Üretici Çekilmeli Ağ (AE-GAN) modelleri ile elde edilen görüntülerin daha gerçekçi olması ve çeşitlilik sağlayabilmesi hedeflenmiştir. Bu ağ modelleri incelendiğinde

düşük çözünürlüklü görüntüler ile çalışıldığı dikkat çekmektedir. Bir görüntünün farklı ölçeklerde görüntüsünün yeninde yapılandırma ile elde edilmesinin ya da rastsal bir görüntünün istenen boyutta elde edilmesinin aynı model ile mümkün olmadığı görülmüştür.

Üç boyutlu nesnelerin algılanması; bakış açısı, ışık kaynağından kaynaklanan aydınlanma gibi çok sayıda dış faktörler ile oldukça ilgilidir. Bu faktörler; üç boyutlu nesnelerin anlaşılması, modellenmesi ve sentezlenmesini çok daha karmaşık bir hale getirmektedir. Özellikle son yıllarda, iki boyutlu görüntülerden üç boyutlu nesnelerin algılanabilmesi problemi araştırmacıların ilgisini çeken bir araştırma konusu olmuştur.

Standart GAN ve VAE tabanlı modeller üç boyutlu nesneler için doğrudan uygulanabilir değildir. İki boyutta etkinliği dikkat çeken GAN ve VAE modellerini 3 boyuta aktarabilmek üzere birçok araştırmacı çalışmalarda bulunmuştur [50-54]. Bu konuda yapılan çalışmalarda düşük boyutlu sentetik nesneler modellenmeye odaklanılmıştır. Literatürdeki çalışmaların önemli kısmında [51-57] iki buçuk boyutlu (2,5B) olarak ifade edilen derinlik görüntüleri veya üç boyutlu nesnelerden 3 boyuta dönüşüm üzerine çalışılarak literatürde kaydedilen en iyi performans değerleri elde edilebilmiştir. Üç boyutlu verilerin eğitimi hesaplama maliyeti bakımından ele alındığında, yüksek maliyetli olarak değerlendirilmekle birlikte gerçek hayatta üç boyutlu nesnelerin yer gerçekliği (ground truth) verilerinin mevcut olamaması üç boyutlu nesnelerin gözetimli öğrenme ile eğitilmesini zor kılmaktadır. Literatürde mevcut olan çalışmaların bir kısmında [58-61] ise nesne oluşturma için bir nesnenin çok sayıda görüntüsüne gereksinim duyulduğu dikkat çekmektedir.

Gerçek hayattaki veriler sentetik verilerden aydınlanma, çevresel faktörler ve birden fazla nesne içeren yapıları nedeniyle oldukça farklılık göstermektedir. Yer gerçekliği verilerinin gerçek hayatta mevcut olamaması söz konusudur. Silüet olarak ifade edilen görüntülerden nesnelerin çıkarımın RGB görüntülere göre daha hızlı ve kolay olacağı öngörülmüştür [62]. Bununla birlikte, silüet görüntülerinden nesnelerin geometrik olarak öğrenilmesindeki zorluklar, bu veriler üzerine sınırlı sayıda çalışma olmasına neden olmuştur.

Tezin amacı ve çözüm önerileri

Tez çalışmasında, üretici ağlar ile görüntü ve nesne oluşturma problemi irdelenmiştir. Öncelikli olarak, ölçeklenebilir sentetik görüntü oluşturma problemi ele alınmıştır. Sentetik

görüntü oluşturma problemi için ölçeklenebilir bir model aracılığıyla farklı ölçeklerde sentetik fakat gerçekçi görüntüler elde etmeye çalışılmıştır. Bu amaçla, Kompozisyonel Örüntü Üretici Ağ (Compositional Pattern Producing Network) (CPPN) yaklaşımı ile VAE modeli bir araya getirilerek çekişmeli eğitime dayalı öznitelik tabanlı yeniden yapılandırma amaç fonksiyonu ile yeni bir üretici ağ modeli olan Değişimsel Otokodlayıcı ile Kompozisyonel Çekişmeli Üretici Ağ (VAE/CPGAN) önerilmiştir.

Tezin devam eden bölümlerinde, iki boyutlu tek açıdan görüntülerden üç boyutlu nesnelerin oluşturulmasına odaklanılmıştır. Gerçek problemler için nesnelerin çoğunlukla tek açıdan görüntülerinin mevcut olması ve nesnelere ait görüntülerin elimizde yer gerçekliği olarak ifade edilen üç boyutlu verilerinin mevcut olamaması gibi karşılaşılan problemler nedeniyle, özellikle son yıllarda ele alınan, tek açıdan görüntülerden üç boyutlu nesne oluşturma problemi çözümlenmeye çalışılmıştır. Bu nedenle tez kapsamında iki boyutlu kodlayıcı (encoder) ile üç boyutlu kodçözücü (decoder) ya da üretici (generative) ağlardan oluşan modeller tasarlanarak çok kategorili modelleme amaçlanmıştır. Literatürde ağırlık kazanan kategori bazlı modeller yerine, tek bir model ile farklı kategoriden görüntüleri modelleyebilmek üzere, sınıf etiketleri bir koşul parametresi olarak kullanılarak, öncelikle, çekişmeli eğitime dayalı Voksel tabanlı Koşullu Otokodlayıcı ile Çekişmeli Üretici Ağ (VoxCAE/GAN) modeli önerilmiştir. Önerilen modelde, iki boyutlu kodlayıcı ile tek bir silüet görüntüsünden elde edilen gizli kod (latent code) ve sınıf bilgisi kullanılarak üç boyutlu üretici ağ ve çekişmeli eğitim yaklaşımına dayalı olarak, üç boyutlu nesnelerin elde edilmesi planlanmıştır. Önerdiğimiz VoxCAE/GAN modeli ile silüet görüntüleri kullanılarak öğrenme işlemi kolaylaştırılırken, modelin geri-çatım kabiliyetini artırmak üzere, geri-çatım maliyeti ile çekişmeli eğitime dayalı birden çok amaç fonksiyonunu eniyileme hedeflenmiştir. Bu kapsamda, öncelikli olarak, farklı geri-çatım yaklaşımları üzerine model geliştirilmeye çalışılmıştır. Nesne bölütleme probleminde kullanılan Bileşim üzerinde Kesişim (Intersection-over-Union) (IoU) olarak ifade edilen metriğe dayalı bir maliyet fonksiyonundan esinlenilerek, üç boyutlu nesne geri-çatımı için IoU tabanlı bir optimizasyon yaklaşımı önerilmiştir. Geliştirilen bu model ile üretici modellerde daha önce kullanılan geri-çatım maliyet yaklaşımlarına dayalı eğitime kıyasla hem niteliksel hem de niceliksel olarak daha iyi sonuçların elde edildiği gösterilmiştir. Tez çalışmasında önerilen, IoU tabanlı çekişmeli eğitime dayalı çoklu amaç fonksiyonu ile en iyi performansın elde edildiği literatürdeki diğer modellerde de kullanılan ShapeNetCore [63] sentetik veri seti üzerinde gösterilmiştir.

IoU amaç fonksiyonuna dayalı VoxCAE/GAN modeli ile elde edilen model performansını iyileştirebilmek üzere, modele çekişmeli eğitim ve koşul parametresinin eğitime katkısını irdeleyebilmek üzere, VoxAE ve VoxCAE modelleri de sunulmuştur.

Önerilen VoxCAE/GAN, VoxCAE ve VoxAE modellerinde kullanılan gizli kod gibi yüksek seviyeli öznitelikler ile birlikte daha düşük seviyeli özniteliklerden yararlanabilmek üzere, artık bağlantılara (residual connection) [64] dayalı bir model üzerine çalışılmıştır. Biyomedikal görüntü bölütleme problemi için önerilen U-Net [65] ve U-Net benzeri modellerin elde ettiği başarılardan etkilenilerek mevcut U-Net modellerinden farklı olarak artık bağlantılara dayalı iki boyutlu otokodlayıcı yerine artık bağlantılar ile iki boyuttan üç boyuta koşullu otokodlayıcı modeline odaklanılmıştır. Böylece, Ara-bağlantılı Voksel Koşullu Otokodlayıcı (Skip-connectional Voxel Conditional AutoEncoder) (SkipVoxCAE) modeli de tez kapsamında sunulmuştur.

Gerçek görüntüler ile üzerinde çalışılan sentetik görüntüler arasında mevcut olan farklılıklar nedeniyle, görüntülerden nesne oluşturma problemini çözmek üzere, doğrudan RGB görüntülerinden nesne oluşturmak yerine, RGB görüntülerden elde edilen silüet görüntülerine dayalı modelleme hedeflenmiştir. Silüet görüntülerinden nesnelerin geometrik yapısının öğrenilmesindeki zorluklar nedeniyle, tez süresince ele alınan diğer modellerde olduğu gibi, silüet tabanlı bir kodlayıcıya ek olarak RGB tabanlı bir kodlayıcıya ve bir voksel kodçözücüye dayalı model üzerinde çalışılmıştır. Böylece, RGB özniteliklerinden de faydalanarak nesne oluşturmak üzere RGB ve silüet kodlayıcıların füzyonuna dayalı bir model olan Füzyon edilmiş Voksel Koşullu Otokodlayıcı (Fused Voxel Conditional AutoEncoder) (FusedVoxCAE) modeli de tez kapsamında önerilmiştir.

Tez çalışmasının literatüre katkısı

Tez çalışmasının literatüre katkıları, çekişmeli eğitime dayalı üretici ağ modelleri ile görüntü oluşturma problemi ve tek açıdan görüntülerden nesne oluşturma problemi açısından iki grupta ele alınmıştır.

Çekişmeli eğitime dayalı üretici ağ modelleri ile görüntü oluşturma problemi üzerine yapılan tez çalışmaları ile literatüre yapılan katkılar:

- Görüntü oluşturma ve yeniden yapılandırma için yeni bir otokodlayıcı ve çekişmeli eğitime dayalı hibrit model: VAE/CPGAN
  - GAN modeline dayalı sunulan model ile sentetik görüntü oluşturma ve görüntülerin yeniden yapılandırılması konularında performansın iyileştirilmesi,
  - Çekişmeli eğitime dayalı sunulan kompozisyonel üretici ağ ile elde edilen ölçeklenebilirlik özelliği sayesinde farklı ölçeklerde süper çözünürlük (SÇ) görüntülerinin bu çözünürlüklerde eğitim yapılmaksızın elde edilebilmesi,
  - Daha önceki çalışmalarda kullanılan piksel tabanlı amaç fonksiyonu yerine öznitelik tabanlı amaç fonksiyonu kullanılarak daha gürbüz bir üretici ağ modelinin elde edilmesi,
  - GAN tabanlı modellerin kapsamlı bir şekilde ilişkisel olarak ele alınması ve kategorik olarak incelenmesi

olarak sıralanabilmektedir.

Tek açıdan görüntülerden nesne oluşturma problemi için katkılar ise:

- İki boyutlu silüet görüntülerinden nesne oluşturmak üzere çekişmeli eğitime ve IoU metriğine dayanan yeni bir hibrit model: VoxCAE/GAN
  - IoU metriğinin türevlenebilir amaç fonksiyonu olarak nesne oluşturma problemine ilk kez uyarlanması ve diğer amaç fonksiyonları ile performans karşılaştırmaları
- İki boyutlu silüet görüntülerinden nesne oluşturmak üzere otokodlayıcı dayalı modeller: VoxAE ve VoxCAE modelleri
  - Çekişmeli eğitim, koşul parametresi olarak kullanılan sınıf etiketi ve yeniden yapılandırma metriklerinin model performansına etkisinin incelenmesi
- İki boyutlu silüet görüntülerinden nesne oluşturmak üzere ara bağlantıları olan, zayıf gözetimli eğitime dayalı otokodlayıcı modeli: SkipVoxCAE
- İki boyutlu RGBA görüntüleri kullanılarak nesne oluşturmak üzere derinlik ve RGB kodlayıcı ve voksel kodçözücüden oluşan, zayıf gözetimli eğitime dayalı otokodlayıcı modeli: FusedVoxCAE
- Üç boyutlu nesne oluşturma ve yeniden yapılandırma için literatürde mevcut olan çalışmaların en kapsamlı şekilde gruplandırılarak ele alınması

olarak özetlenmiştir.

Tez planı

Tez kapsamında ele alınan problem ile bu probleme çözüm önerilerinin sunulduğu, tez çalışmasının amacı ile katkılarına yer verilen "giriş" bölümünü, tez çalışması kapsamında ele alınan üretici ağ modelleri ile ayrımcı ağ modelleri üzerine çalışmaların incelendiği ikinci bölüm olan "ilgili çalışmalar" bölümü takip etmektedir. İlgili çalışmalar bölümünde tez çalışmasında ele alınan konu üzerine literatürde mevcut olan üretici ağa dayalı çalışmalar iki boyutlu ve üç boyutlu üretici ağ modelleri olarak kategorize edilerek bu bölümde detaylı bir şekilde açıklanmıştır. Üçüncü bölümde, tez çalışmasında önerilen modellere temel olan; derin ileri beslemeli ağlar, evrişimsel ağlar ve üretici ağlar kavramsal olarak ele alınarak açıklanmıştır. Dördüncü bölümde, tez kapsamında ele alınan ilk problem olan, görüntü oluşturma problemi için önerilen çekişmeli eğitime dayalı üretici ağ modeli matematiksel olarak ifade edilmiştir. Beşinci bölümde ise, tez çalışması olarak ele alınan diğer problem olan, tek açıdan görüntülerden üç boyutlu nesne oluşturma problemi için önerilen VoxCAE/GAN, VoxAE, SkipVoxCAE ve FusedVoxCAE model açıklamalarına yer verilmiştir. Deney çalışmaları ile ilgili olan altıncı bölümde, her iki problem için kullanılan veri kümeleri, önerilen model mimarileri, eğitimleri ve parametre bilgileri gibi model temel bileşenleri açıklanarak önerilen metrik üzerinde kaydedilen deneysel sonuçlara ve model performans değerlendirmelerine detaylı bir şekilde yer verilmiştir. Son bölümde ise tez çalışmasında ele alınan problem kısaca özetlenerek tez çalışması kapsamında yapılan çalışmalar ve literatüre katkılar tekrar ortaya konulmuş ve elde edilen sonuçlar kısa bir şekilde değerlendirilmiştir.

2. İLGİLİ ÇALIŞMALAR

2.1. Ayrımcı Modeller

Literatürde yer alan derin sinir ağı (Deep Neural Network) (DNN) tabanlı yöntemler görüntü sınıflandırma, sahne sınıflandırma, bölütleme ve diğer problemler için CNN tabanlı yöntemler, dil tanıma problemleri ile video sınıflandırmada, video nesne tanıma ve kümelemede RNN tabanlı yöntemler ön plana çıkmıştır. Nesne tanıma, nesne tespiti, kümeleme ve aktivite sınıflandırma gibi problem için CNN ve RNN ağlarının bir arada kullanıldığı modeller geliştirilmiştir. Ayrımcı modeller olarak kategorize edilen CNN ve RNN ağları üretici ağlar olarak kategorize edilecek çalışmalar literatürde mevcuttur. Bu bölümde DNN çalışmaları bu iki kategori altında ele alınarak temel CNN ve RNN modelleri ile asıl odaklanılan üretici modeller ele alınmıştır.

2.1.1. CNN tabanlı yöntemler

Evrişimsel ağlar üzerine yapılan ilk çalışmalardan biri, LeNet-5 [11] olarak adlandırılan ağdır. Karakter tanımada kullanılmak üzere tasarlanan LeNet-5, 3 evrişim katmanı (conv), 2 örnekleme katmanı, 1 tam-bağlı katman (fully-connected) (fc) ve çıktı katmanı olmak üzere 7 katmandan oluşmaktadır. Karışık Ulusal Standartlar ve Teknoloji Enstitüsü (Mixed National Institute of Standards and Technology) (MNIST) [66] veriseti üzerinde farklı doğrusal sınıflandırma yöntemleri ile yapılan karşılaştırmalara göre hata oranı %0,8'e düşürülmüştür. Bunun yanı sıra, hata oranı ve hafıza gereksinimi gibi karşılaştırmalarda da LeNet-5 ağı başarılı olmuştur.

İlk CNN çalışmalarından biri olan AlexNet [10] 60 milyon parametre ile eğitilmiştir. Çalışmada Lokal Yanıt Normalizasyonu (LRN) ve örtüşen biriktirme (overlapping pooling) ile hata oranı düşürülmüştür. Aşırı öğrenmeye karşı veri arttırma (data augmentation) ve iletim sönümü (dropout) yaklaşımları kullanılmıştır. AlexNet ağı 5 evrişim katmanı ve 3 tam-bağlı katmandan oluşmaktadır. AlexNet'in ilk 2 evrişim katmanını LRN katmanları takip etmiştir. En büyük biriktirme (max pooling) ise LRN katmanlarından ve 5.ci evrişim katmanından sonra uygulanmıştır. AlexNet, ILSVRC 2010'da öznitelik kodlayıcı (feature encoding) tabanlı yöntemlerle karşılaştırılmıştır. Karşılaştırma sonuçlarına göre en yüksek 5 (top-5) tahmin skoruna göre %17.0 hata elde edilmiştir. ILSVRC 2012'de elde edilen

%16.4 hata oranı ImageNet 2011 versiyonu görüntüleri ile eğitilen model finetune edilerek top-5'te %15.3'e düşürülebilmiştir.

Derin ağlardaki performansı artırmak üzere Ağ içinde Ağ (Network in Network) (NIN) modelinde [67] yeni bir çıktı katmanı önerilmiştir. Bu katman ile geleneksel evrişimsel ağların son katmanında bulunan, öznitelikleri vektörleştirilerek softmax olarak adlandırılan katmanı besleyen, yapı yerine kullanılarak aşırı öğrenme probleminin de önleneceği düşünülmüştür. Ağın genelleştirme kabiliyetini artırarak aşırı öğrenmeyi engelleyebilmek üzere global ortalama biriktirme (global average pooling) stratejisi sunulmuştur. Bu strateji ile evrişimsel operasyon, çok katmanlı perseptron katmanı (mlpconv) olarak adlandırılan, son katmandaki her bir öznitelik haritasının ortalaması alınarak elde edilen, vektör ile softmax katmanı beslenmektedir. Ağ içinde Ağ modeli, 3 mlpconv katmanı ve global ortalama biriktirme katmanından oluşmaktadır. Tam-bağlı katman yerine kullanılan global ortalama biriktirme sayesinde sınıf kategorileri ile öznitelik haritaları arasında doğrudan bağlantı kurmak mümkün olmuştur. CIFAR-10 [68], CIFAR-100 [69], Sokak Görünümünde Ev Numaraları (Street View House Numbers) (SVHN) veritabanı [70] ve MNIST üzerinde yapılan karşılaştırmalarda CIFAR-10 ve CIFAR-100 için NIN ile en iyi sonuçlar elde edilmiş iken SVHN ve MNIST üzerinde umut vadeden performanslar elde edilebilmiştir. Tam-bağlı katman ile global ortalama biriktirme katmanını karşılaştırmak üzere de çalışmalar gerçekleştirilmiştir. Bu çalışmalarda, global ortalama biriktirme ile performansın artırıldığı gösterilmiştir.

Google Ağı (GoogLeNet) [71], NIN çalışmasından esinlenilerek inception modülü ile bir ağ yapısına dayanmaktadır. Bu model ile ağ derinliği artırılırken parametre sayısını artırmamak üzere tam-bağlı katman yerine seyrek yapıya odaklanmıştır. Bu seyrek yapıyı sağlamak üzere başlangıç (inception) olarak adlandırdıkları modüllerden yararlanılmıştır. Bir katmanda birden fazla evrişim işlemi ile az parametre kullanılarak iyi bir filtre alanı sağlamak amaçlanmıştır. 1×1, 3×3 ve 5×5 boyutlarında evrişim katmanı ile maksimum ortalamadan oluşan evrişim katmanından yararlanılmıştır. 3×3 ve 5×5 boyutlarındaki evrişim katmanlarından önce kullanılan 1×1 evrişim katmanı ile uzayda değil renk kanalında indirgeme yapılarak parametre sayısı azaltılıp maliyet düşürülmüştür. GoogLeNet ya da inception olarak bilinen ağ parametresi olan katmanlar dikkate alındığında 22 katmandan, tüm katmanlar dikkate alındığında ise 100 katmandan oluşmaktadır. Bu mimari ile AlexNet'ten [10] 12 kat az parametre ile çok daha iyi bir performans elde edilmiştir.

ILSVRC 2014 sınıflandırma yarışmasında top 5'te %6.67'lik hata oranı ile birinci olarak oldukça ses getirmiştir.

Chatfield ve arkadaşları [72], CNN tabanlı yöntemlerin başarılı olmasına rağmen bunun nasıl mümkün olduğunu analiz etmek üzere daha önce öznitelik kodlama yöntemleri üzerine yaptıkları çalışmaya benzer olarak performansın detaylarla olan ilintisini ele almışlardır. 3 farklı senaryodaki görüntü betimlemelerini analiz etmişlerdir. Veri arttırma (data augmentation), ince ayar (finetuning), öznitelik düzenlileştirme (normalizasyonu), renk bilgisi ile CNN son katmandaki çıktı sayısını azaltarak boyut indirgeme üzerine karşılaştırmalar yapmışlardır. 3 farklı önceden eğitilmiş ağ (Yavaş Evrişimsel Sinir Ağı (CNN-S), Orta Evrişimsel Sinir Ağı (CNN-M), Hızlı Evrişimsel Sinir Ağı (CNN-F)) üzerinde performans analizleri yapmışlardır. Bu incelemeler sonunda ,veri arttırma işleminin performansı artırdığını kaydetmişlerdir. İnce ayar işlemi (finetuning) sırasında softmax maliyet fonksiyonu yerine sıralama (ranking) maliyet fonksiyonu ile performans artışı elde etmişlerdir. Filtre boyutları küçüldükçe ve ağ derinleştikçe performansın olumlu etkilendiği görülmüştür. CNN tabanlı yöntemlerin, sığ (shallow) olarak ifade edilen yöntemlerden üstün olduğu fakat CNN'lerde uygulanan detaylarla sığ modellerde de performans artışının mümkün olacağı değerlendirilmiştir. ILSVRC 2012 [19], Pascal VOC 2007 [73], Pascal VOC 2012 [74], Caltech 256 [75] gibi verisetlerinde en iyi sonuçlar elde edilmiştir.

Görsel Geometri Grubu ağı (Visual Geometry Group) (VGG) olarak adlandırılan model [76], 11-19 arası farklı derinliklerde küçük evrişim filtrelerinden oluşan modeller olarak sunulan, literatürde oldukça ses getiren, bir diğer çalışmadır. Bu çalışmada, sığ olarak ifade edilen 11 katmanlı ağdan daha derin modeller elde edilmiştir. Bu modellerden 16 ve 19 katmandan oluşan iki ağın ensemble edilmesi ile ILSVRC 2014 yarışmasının lokalizasyon kategorisinde birincilik, sınıflandırma kategorisinde ise ikincilik elde edilmiştir. Model değerlendirmelerinde farklı ölçeklerde görüntüler ve yoğunluk ve çoklu kırma değerlendirmelerine de yer verilmiştir. Yarışmadaki en iyi performans, Görsel Geometri Grubu 16 katmanlı ağı (VGG16) ve Görsel Geometri Grubu 19 katmanlı ağıdan (VGG19) ensemble edilen model ile yoğunluk ve çoklu kırma değerlendirmesi ile elde edilebilmiştir. Test verisinde top-5 hata oranı %6.8'e düşürülmüştür.

Zeiler ve Fergus [77], derin ağların orta katmanlarında yakalanan orta seviyeli betimlemeleri anlayabilmek üzere ters-evrişim (deconvolution) katmanlarından (deconv) oluşan Ters

Evrişimsel Ağ (DeconvNet) modelini sunmuşlardır. Bu çalışmada mimari olarak AlexNet'e benzer bir model üzerinde çalışmışlardır. Öznitelik aktivasyonlarının girdi piksel uzayına map etmek üzere diğer katmanlardaki aktivasyonlar sıfırlanarak dağıtım (unpooling), düzeltme (rectifying) ve filtreleme işlemleri yapılmıştır. Yapılan çalışma ile elde edilen görselleştirim sayesinde veride transformasyon işleminin etkisi de irdelenmiştir. Modelin ölçekleme ve taşıma işlemlerine karşı stabil kalmış iken rotasyon işlemine kalamadığı görülmüştür. Benzer şekilde görüntü parçaları kapatılarak benzeşme analizi gibi analizler yürütülmüştür. Çalışmada önerilen model ile Caltech-101 [78] ve Caltech-256 verisetlerinde en iyi sonuçlar elde edilmiştir.

Artık Sinir Ağı (Residual Network) (ResNet) modelinde [79], artık öğrenme (residual learning) olarak ifade edilen öğrenme yaklaşımı ile derin ağların eğitimini kolaylaştırmak ve modellerin performanslarını arttırmak hedeflenmiştir Artık öğrenme ile daha önceki kodlanan (encode) katmanlardan daha farklı bir bilgi elde edilmesi sağlanmaktadır. Ayrıca, DNN'lerde sıkça karşılaşılan kaybolan gradyan (vanishing gradient) probleminden kaçınmak mümkün olmaktadır. Artık öğrenme ile ağları optimize etmenin daha kolay olduğu ve oldukça artan derin ağ yapısı ile sınıflandırma performansının iyileştirildiği görülmüştür. ILSVRC 2015 yarışmasında ResNet birincilik elde etmiştir. Lokalizasyon, nesne tespit etme, kümeleme gibi farklı görevler için de benzer sonuçlar elde edilmiştir.

Nesne tespiti üzerine çalışmalar incelendiğinde Bölge Evrişimsel Sinir Ağı (R-CNN) modeli [80] dikkat çekmektedir. R-CNN ile nesneler CNN ile tespit edilerek sınırlayıcı kutular (bounding box) ve etiketler ayrı ayrı çıktı olarak üretilmiştir. Hızlı Bölgesel Evrişimsel Sinir Ağı (Fast R-CNN) [81] ile seçilen arama için öneriler (proposal) oluşturularak sınırlayıcı kutular, etiketler birlikte çıktı olarak elde edilebilmiştir. Devam eden çalışmalarda Daha Hızlı Bölgesel Evrişimsel Sinir Ağı (Faster R-CNN) [82] ile bir bölge öneri ağı ile öneriler oluşturulabilmiştir. Faster R-CNN, bir Fast R-CNN ile elde edilen özniteliklerden bölge önerileri çıkarmaktadır. Faster R-CNN ile daha iyi öneriler daha hızlı bir şekilde elde edilebilmiştir.

Bölütleme problemi için Long ve arkadaşları [83] AlexNet, VGG ve GoogLeNet gibi sınıflandırma için kullanılan modellerden edinilen gösterimleri transfer etmek üzere ince ayar (finetune) ederek bölütleme problemine uyarlamışlardır. Pascal VOC 2012 veriseti üzerinde %20 performans artışı ile elde edilmiştir.

Aktivite tanıma problemi için Simonyan ve Zisserman [84] video çerçeve aktiviteleri tanımak üzere 2 akış şeklinde uzaysal ve zamansal ağlardan oluşan CNN mimarilerini sunmuşlardır. Uzaysal kısım ile nesne ve sahneler hakkında bilgiler elde edilirken zamansal kısım nesnelerin ve kameranın hareketi hakkında bilgi sağlamak üzere kullanılmıştır. Çok-çerçeveli yoğun optik akış (multi-frame dense optical flow) ile eğitilen CNN sayesinde sınırlı sayıdaki eğitim verisine rağmen iyi bir performans elde edebilmişlerdir. Merkez Florida Üniversitesi 101 (UCF101) [85] ve İnsan Hareketleri Tanıma Veri Tabanı (Human Motion Database) (HMDB51) [86] verisetleri üzerinde çalışmada sunulan 2 akış halindeki CNN değerlendirilmiştir. UCF101 verisetinde en iyi sonuçlar elde edilmiş iken HMDB-51 veri setinde de kaydedilen en yüksek sonuçlara yakın performans elde edilebilmiştir.

2.1.2. RNN tabanlı modeller

RNN ile gizli katmandaki nöronlardaki bir durum vektörü ile sekansın eski elemanları hakkında bilgi tutulması mümkün olmaktadır. Ortaya çıkış amacı uzun süreli bağıntıları öğrenmek olsa da RNN ile pek mümkün olmadığı görülmüştür. Bu sorunu çözmek üzere Uzun-Kısa Vadeli Hafıza (LSTM) [87] diye adlandırılan ağı bir dış hafızada tutan yapı önerilmiştir. LSTM'de hafıza hücresi olarak ifade edilen özel hücre bir akümülatör gibi davranmaktadır. Sonraki zamanda kendine bağlanılmasıyla kendi değerini kopyalamış ve biriktirmiş olacaktır [1].

CNN'in bir zaman dilimine sınırlı olması, zamanla ilgili öznitelikleri öğrenme konusunda yetersiz olması CNN ve RNN tabanlı modellerin bir arada kullanıldığı çalışmalara neden olmuştur. Bu çalışmalarda her bir modellerin ayrı öğrenilerek uçtan uca eğitilebilir bir sistem sağlayamaması söz konusudur. Bu soruna karşılık Uzun Vadeli Tekrarlayan Evrişimsel Ağlar (LRCN) [88] diye adlandırılan görüntü tanıma ve açıklama sunmak üzere yeni bir yaklaşım ortaya koyulmuştur. Bu çalışmada Evrişimsel katmanlarla uzun aralıklı zamansal yineleme ile çift derinlikte bir ağ yapısına odaklanılmıştır. Bu yapı sayesinde uçtan uca eğitim söz konusu olmuştur. Video aktivite tanıma problemi için CNN, AlexNet [10] ile DeconvNet [77] birleşimi bir model olarak ILSVRC 2012 verisetinde eğitilmiştir. Temel modeli olarak her bir çerçeve CNN ile bağımsız olarak sınıflandırılarak final etiketi her bir zaman diliminde yapılan tahminlerin ortalaması alınarak elde edilmiştir. UCF101 veriseti üzerinde yapılan değerlendirmelerde 8 çerçeve aralıkla alınan 16 çerçeve kullanılmıştır.

LRCN-fc6 olarak adlandırılan LSTM'in CNN fc6 katmanı yerine konulduğu model ile en yüksek sonuç olan 76.95 başarım elde edilmiştir. Görüntü açıklama için başlıktan görüntü ve görüntüden başlık elde etmek üzere 2 farklı karşılaştırma yapılmıştır. Yapılan karşılaştırmalarda LRCN en iyi performansı elde etmiştir. Son LRCN versiyonu olan video açıklama için hazırlanan model ile en iyi sonuçlar sağlanmıştır.

Xu ve arkadaşları [89] görüntü başlıklandırma problemi için bir ilgi tabanlı mimari sunmuşlardır. Bu çalışmada insanların görüntüyü anlamlandırma mekanizmasından etkilenilmiştir. Çalışmada sunulan model evrişimsel ağlar ile özyinelemeli ağları bir araya getirerek ilgi mekanizmasını modellemeye odaklanmıştır. Modelde girdi olarak alınan görüntüden evrişimsel ağlar ile elde edilen öznitelikler ilgi mekanizmasına göre parça parça alınarak RNN ile içerik bilgisi elde edilmektedir. Flickr8k, Flickr30k ve Microsoft İçerikte Ortak Objeler Veri Kümesi (MsCOCO) [90] gibi verisetleri üzerinde yapılan karşılaştırmalarda en iyi sonuçlar elde edilmiştir.

RNN tabanlı yapılan diğer bir ilginç çalışma ise açık uçlu sorular ve görüntülerden oluşan veriseti sunulan Görsel Soru Yanıtlama (Visual Question Answering) (VQA) isimli model [91] olmuştur. Bu çalışmada Turing testine benzer şekilde sorulan açık uçlu sorular görüntü yorumlanarak cevaplanmaya çalışılmaktadır. Alınan görüntü bir Evrişimsel ağ ile eğitilerek öznitelikler elde edilmektedir. Sorulan açık uçlu sorular bir kelime vektörüne dönüştürülerek LSTM ağı ile bir çıktı vektörü elde edilmektedir. Bu vektör ile CNN çıktısının girdi olarak alındığı bir ağ ile sorulan açık uçlu soruya bir yanıt üretilmektedir. Oluşturulan veriseti üzerinde önerilen yaklaşım ile açık uçlu sorularda %54.06, çoktan seçmeli sorularda ise %59.85 performans elde edilmiştir.

Nesne tanıma için Tekrarlayan Evrişimsel Sinir Ağı (RCNN) [92] olarak adlandırılan her bir evrişimsel katmanlar arasında Tekrarlayan bağlantılar kurmaya dayanan model sunulmuştur. Bu çalışmada neocorteks olarak adlandırılan bölgede ileri beslemeli snapsislerden sayıca fazla olan yenilemeli snapsislerden esinlenilmiştir. Bu yaklaşım ile her bir RCNN nöron aktiviteleri komşu nöron aktiviteleri ile ilişkilendirilerek içerik bilgisi elde edilebilmektedir. Çalışmada sunulan model açıldığında RNN'lerde olduğu gibi sabit parametreli derin bir ağ elde edilmektedir. Bu model ile tekrarlayan yapının nesne tanımadaki etkisi ortaya konulmuştur.

2.2. İki Boyutlu Üretici Modeller

Bu bölümde üretici modeller dört kategoriye ayrılarak; gözetimsiz öğrenmeye dayalı temel modeller, AE tabanlı modeller, otoregressif modeller, GAN tabanlı modeller ve AE/GAN hibrit modelleri başlıkları altında ele alınmıştır. Şekil 2.2'de iki boyutlu üretici modeller, modeller arası ilişkiler ile görselleştirilmiştir.

2.2.1. Gözetimsiz öğrenmeye dayalı temel modeller

Gözetimsiz öğrenmeye dayanan temel modeller, el yazısı sayılarının sınıflandırılması ve doku sentezi gibi konuları üzerine daha önce çalışılmış ancak elde edilen sonuçlar nedeniyle yeterli olamamıştır. Kısıtlı Boltzmann Makinesi (RBM) [93], artan hesaplama imkanları nedeniyle bir Boltzmann Makinesi (BM) modeli olarak ortaya çıkarak çeşitli problemlere uygulanmıştır. RBM ve Derin Boltzmann Makinesi (DBM) [9] gizli vektör olarak ifade edilen gösterimden kod çözücüleri ve Gibbs örneklemesi ile giriş görüntülerini yeniden oluşturabilmiştir. Markov zinciri Monte Carlo (MCMC) [94], RBM ağından örnek oluşturmak üzere kullanılan bir metot olmuştur. Derin İnanç Ağları (DBN) olarak adlandırılan [95] RBM ağlarının yığılması ile yüksek seviyeli özniteliklerim elde edilebildiği çok katmanlı mimari elde edilmiştir.

2.2.2. AE tabanlı modeller

Otokodlayıcılar [23], kodlayıcı ve kod çözücü yapısı ile verileri sıkıştırarak sıkıştırılmış versiyonlarından geri elde edilebilmesi için eğitilen gözetimsiz öğrenmeye dayalı bir ağ modelidir.

Yığılmış Otokodlayıcı (SAE), yığılmış yapıdaki otokodlayıcının açgözlü katman katman eğitimine dayanmaktadır. Gürültü Otokodlayıcı (DAE) modelinde [96], girdi olarak gürültülü veri alarak daha güçlü gösterim elde edilmesine odaklanılmıştır. DAE, SAE yaklaşımlarının bir araya getirildiği Yığılmış Gürültüsüz Otokodlayıcı (SDAE) [97] modelinde de yığılmış model gürültü girdilerini alarak gürültüsüz verilerin elde edilmesini mümkün kılmıştır. Otokodlayıcı modellerinde karşılaşılan kaybolan gradyan ve aşırı öğrenme problemini çözmek üzere bir varyasyon bileşeni kullanılarak Kullback-Leibler (KL) ıraksama metriği tabanında model eğitilmesine dayanan VAE [24] modeli ortaya çıkmıştır.

Çekişmeli eğitim yaklaşımındaki gelişmeler ile birlikte bu yaklaşım KL ıraksama maliyet fonksiyonu yerine kullanılarak Çekişmeli Otokodlayıcı (AAE) [45] modeli ortaya çıkarılmıştır.

VAE modelini yarı-gözetimli eğitime dayalı model olarak Yarı Gözetimli Değişimsel Otokodlayıcı (SS-VAE) [98] modeli geliştirilmiştir.

Üretici Stokastik Ağ (GSN) [99], Genelleştirilmiş Gürültüsüz Otokodlayıcı (GDAE) [100] modelinden geliştirilerek MCMC metodunun bir adımı olarak parametrelerin öğrenilerek görüntüler oluşturulmasına dayanmaktadır.

Literatürde oldukça ses getiren çalışmalardan biri olan Derin Tekrarlayan Dikkatli Yazıcı (DRAW) [48] modelinde, dikkat mekanizması yaklaşımı ile Tekrarlayan otokodlayıcı yapısı bir araya getirilmiştir. Bu yaklaşım ile model giriş görüntülerinin hangi bölümlerinin çıktı ile ilişkilendirildiğini çözümleyebilmektedir.

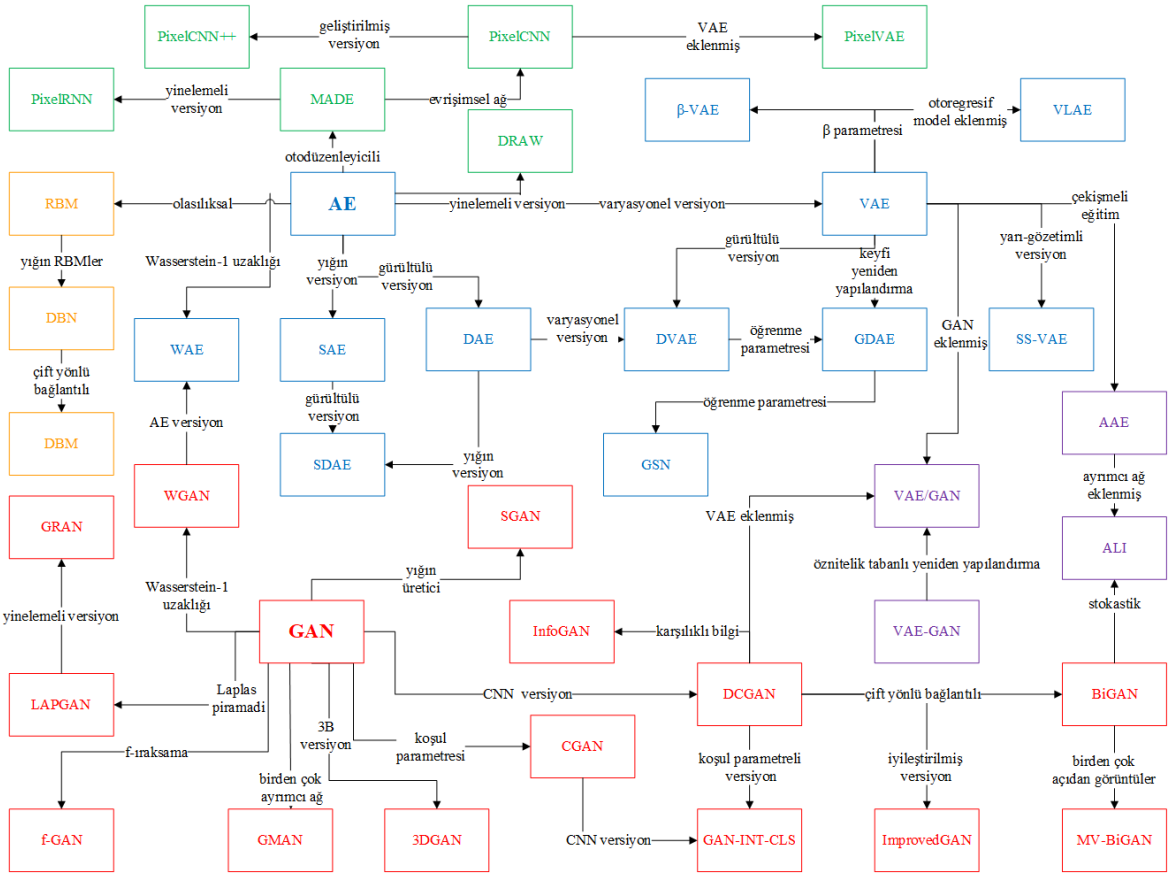
Gürültü Giderici Değişimsel Otokodlayıcı (DVAE) [101] VAE ve DAE yaklaşımlarını bir araya getirerek sadece giriş seviyesinde değil VAE'nin stokastik gizli katman seviyesinde de modele gürültü ekleyerek daha gürbüz bir model elde edilmeye çalışılmıştır.

Beta Değişimsel Otokodlayıcı (β-VAE) [102], VAE modeline eklenen bir β hiper-parametresi ile yorumlanabilir (disentangled) özniteliklerin öğrenilebilmesini hedefleyerek gizli kod kapasitesi ile yeniden yapılandırma başarımı için bağımsızlık kısıtlamaları kapasitesini dengelemek üzere kullanılmıştır.

Wasserstein Otokodlayıcı (WAE) [49] modelinde, Wasserstein uzaklığına ve önerilen düzenleyici fonksiyona göre yeniden yapılandırma objektif fonksiyonu elde edilerek model maliyet fonksiyonu optimize edilmeye çalışılmıştır.

2.2.3. Otoregresif modeller

Bu kategoride ele alınan modellerde tüm görüntünün tek seferde modellenmesi yerine piksel piksel olacak şekilde zaman serisine göre adım adım elde edilmesi hedeflenmiştir.



Şekil 2.2. İki boyutlu üretici modeller

Dağılım Tahmini için Maskelenmiş Otokodlayıcı (MADE) modeli [103], otokodlayıcı parametrelerinin maskelenerek her bir girdinin önceki girdilere göre sıra dikkate alınarak koşullu olasılık tabanında yeniden yapılandırılmasına dayanmaktadır. Piksel Tekrarlayan Sinir Ağı (PixelRNN) modelinde [104], daha önceden oluşturulan piksel değerlerine göre sonraki piksel değerinin koşullu olasılık tabanlı elde edilmesine odaklanarak modelleme problemini bir zaman serisi problemine dönüştürmüştür. Bir seri modeli olan RNN ile piksel değerleri ve dağılımları arasındaki uzun süreli ve doğrusal olmayan bağıntılar çözümlenmeye çalışılmıştır. PixelRNN modelinin eğitiminin uzun sürmesi, RNN yerine evrişim katmanılar kullanarak veri işleme alanın (receptive field) tek piksel yerine piksel öbeği şeklinde arttırılması ile sunulan Piksel Evrişimsel Sinir Ağı (PixelCNN) modelinin [105] iyileştirilmesi gerçekleştirilmiştir. Geliştirilmiş Piksel Evrişimsel Sinir Ağı (PixelCNN++) [106], PixelCNN modelinin kısa-bağlantı, alt örnekleme, iletim sönümü yaklaşımı gibi modifikasyonlar ile iyileştirilmesine dayanmaktadır. Piksel Değişimsel Otokodlayıcı (PixelVAE) [107], VAE ve PixelCNN yaklaşımlarının avantajlarını bir araya getirmek üzere önerilmiştir. Değişimsel Kayıplı Otokodlayıcı (VLAE) [108], RNN, MADE ve RixelRNN/CNN gibi otoregresif

modellerden yararlanarak bilinen olasılık ve kodçözücü dağılımı öğrenilerek VAE üretici performansı iyileştirilmiştir.

2.2.4. GAN tabanlı modeller

Veri dağılımının öğrenilerek modellenmesi ve öğrenilen dağılımdan veri örneklenmesine dayanan bir diğer üretici model yaklaşımıdır. Otoregresif modeller ile bilinen bir olasılık dağılımından açık, belirgin bir dağılım elde edilebilir iken GAN modelleri ile net, açık ve belirgin olmayan model dağılımı elde edilebilmektedir.

Koşullu Üretici Çekişmeli Ağ (CGAN) [29], yüz niteliklerine gibi koşullar tabanında sentetik yüz görüntüleri oluşturmak üzere önerilmiştir. Modelin koşul tabanlı yapısı sebebiyle model ile önceden belirtilen niteliğe uygun görüntüler oluşturulabilmiştir. Laplas Piramit Üretici Çekişmeli Ağ (LAPGAN) modelinde [44], birden çok üretici ve ayrımcı ağ ile düşük çözünürlüklü, kaba bir görüntü ile başlayarak daha gerçekçi, yüksek çözünürlüklü ve iyi görüntüler Laplas piramidi şeklindeki bir mimari ile elde edilmiştir. Derin Evrişimsel Üretici Çekişmeli Ağ (DCGAN) [30], GAN modelinde kullanılan çok katmanlı perseptron ağları yerine üretici ve ayrımcı ağ modellerinin evrişimsel ağlar tanımlanması ile elde edilmiş bir GAN modelidir.

GRAN modelinde [109], DRAW ve LAPGAN modellerinde olduğu gibi çoklu zaman diliminde görüntü oluşturmaya odaklanılmıştır. Boş bir düzlem ve bir gürültü vektörü ile başlanılarak her bir zaman diliminde önceki zaman diliminin çıktısı ve gürültü vektörü sonraki zaman dilimine aktarılmaktadır. Tüm zaman dilimlerinde elde edilen çıktıların bir araya getirilmesi ile görüntüler oluşturulmuştur.

Geliştirilmiş GAN (Improved GAN) [46], GAN modelinin yığın normalleştirme (batch normalization), etiket yumuşatma ve geri yayılım gibi farklı yaklaşımlar ile performansını geliştirmek üzere sunulan bir model olmuştur. Çift yönlü GAN (BiGAN) [110], girdi görüntülerden gizli kod elde ederek ters fonksiyon ile gizli koddan tekrar görüntü oluşturmaya odaklanılmıştır. Modelin çok açılı versiyonu olan Çok Açıdan Çift-yönlü Üretici Çekişmeli Ağ (MV-BiGAN) [111] yoğunluk ve ekstra bakış açılarına sahip görüntüler oluşturulabilmiştir. GAN modelinde tanımlanan en küçük-en büyük (mini-max) problemini iki oyuncu yerine çok oyunculu olarak tanımlayan Üretici Çoklu-Çekişmeli Ağ (GMAN)

[112] modelinde, birden fazla tanımlanan ayrımcı ağ ile model eğitilerek, güncellenen objektif fonksiyonuna dayalı olarak görüntüler elde edilebilmiştir. Yığılmış Üretici Çekişmeli Ağı (SGAN) [113], yığılmış kodlayıcı ile kodçözücülerden oluşan ağ modelidir. Üretici ağ olarak da ele alınan kodçözücü ile gizli kod ile birlikte gürültü vektörleri girdi olarak alınarak yeni görüntüler elde edilmesi hedeflenmiştir. Ayrımcı ağ ile üretilen görüntüler, çekişmeli maliyet fonksiyonuna göre değerlendirilerek koşullu maliyet ve entropi tabanında ağ eğitilmiştir.

F-ıraksamalı Üretici Çekişmeli Ağı (f-GAN) [114], f-ıraksama tabanında yoğunluk oranı tahmini ile üretici model iyileştirilmiştir. Çalışmada GAN modeli, f-GAN modelinin özel bir durumu olarak tanımlanmıştır. Wasserstein Üretici Çekişmeli Ağ (WGAN) [47] modelinde, GAN modelinin eğitilmesi sırasında karşılaşılan tutarsızlık ve diğer sorunları indirgemek üzere wasserstein-1 uzaklık metriği tabanında eğitim yaklaşımı önerilmiştir. Bilgi Maksimize Üretici Çekişmeli Ağ (InfoGAN) [31], üretici kısımda karşılıklı bilgiye dayalı olarak bir c parametresi ile görüntü oluşturmaya dayanmaktadır. Manifold İnterpolasyon eşleştirme farkındalıklı Ayrımcılı Üretici Çekişmeli Ağ (GAN-INT-CLS) modeli [32], diğer çalışmalardan biraz daha farklı olarak, verilen bir cümleye karşılık gelen görüntüyü oluşturabilmek üzere önerilmiştir. Modelde, tekrarlayan kodlayıcı ile elde edilen gizli kod ve girdi olarak alınan metinlerden elde edilen koşul öznitelikleri ile görüntüler oluşturulabilmiştir.

2.2.5. AE-GAN tabanlı hibrit modeller

GAN ile VAE modellerini avantajlarını bir araya getirmek üzere bir grup çalışma yürütülmüştür. VAE ve GAN ile ilgili ilginç çalışmalardan biri olan VAE/GAN modelinde [25] bu iki yaklaşımı olarak adlandırılan bir modelde birleştirmektedir. GAN modelindeki eleman eleman yeniden yapılandırma hedefi yerine VAE yeniden yapılandırma ile model eğitilmiştir. VAE/GAN maliyet fonksiyonu, VAE bilinen olasılığa bağlı maliyet, öznitelik benzerliğine dayalı log-olabilirlik (log-likelihood) çekişmeli eğitime dayalı olarak tanımlanmıştır. Başka bir VAE-GAN modelinde [26], VAE yeniden yapılandırma hatası yerine önceden eğitilmiş bir yardımcı ağa dayalı eğitim gerçekleştirilmiştir. Benzer şekilde, yeniden yapılandırma görüntülerindeki bulanıklık sorununu gidermek üzere [28] numaralı çalışmada DeePSiM olarak adlandırılan maliyet fonksiyonuna dayalı hibrit model önerilmiştir. ALI modeli [27], bir çekişmeli otokodlayıcı modeline dayanmaktadır.

2.3. Üç Boyutlu Üretici Modeller

Günümüzdeki son gelişmelere kadar, geleneksel şablon tabanlı modeller [115-117] ile bu problem çözülmeye çalışılmaktaydı. Bu modellerin yeni nesne sentezlemedeki yetersizliği ile çok sayıda parametre olarak ifade edilebilecek nesne parçalarına gereksinim duymaları gibi zorlukları mevcuttur. Derin öğrenme konusundaki gelişmeler bilgisayarlı görü alanındaki birçok problemi etkilediği gibi önceleri üç boyutlu nesne tanıma problemi için uygulanmıştır. Özellikle CNN tabanlı çok sayıda modelde, üç boyutlu nesne sınıflandırmanın başarıyla yapılabildiği görülmüştür [118-122].

Literatür incelendiğinde üç boyutlu görü problemleri için verilerin
- nokta bulutu [123-129],
- octree [122, 130, 131],
- iskelet verisi [132],
- yüzey örgüsü [133-142],
- kübik yapı [143],
- voksel [50-61, 144-150]

olarak işlenebildiği görülmüştür. İncelenen çalışmalara göre, bu veri tipleri arasında, vokseller, yüzey örgüleri ve nokta bulutu şeklindeki verilerin daha çok tercih edildiği görülmüştür.

Yüzey örgüsü modelleri, deformasyon modelleri ve voksel tabanlı modellerde olduğu gibi parametrik modeller olarak iki grupta ele alınabilir. Deformasyon modelleri bir şablon nesnenin bir deformasyon alanı ile orijinal nesneye dönüştürülmesine dayalı modellerdir [133-135, 137]. Han ve arkadaşları tarafından yapılan inceleme makalesine [43] göre yüzey örgüsü modelleri tepe deformasyonu (vertex deformation), adım adım dönüştürülebilir modeller (morphable models) ve serbest biçimli deformasyon (Free-Form Deformation) (FFD) olarak ele alınabilir.

Özellikle vokseller basitlikleri nedeniyle çok sayıda çalışmada kullanılmıştır. Tez çalışmasında sunulan modellerin voksel tabanlı olması nedeniyle bu bölümde voksel tabanlı modelleme üzerine yapılan çalışmaların incelenmesine ağırlık verilmiştir.

Çizelge 2.3. Tez çalışmasında kategorik olarak ele alınan nokta bulutu, yüzey örgüsü, octree, iskelet verisi, kübik yapı modelleri

Model	Girdi	Model bileşenleri			Projeksiyon/ Model özeti	Maliyet fonksiyonu
		Kodlayıcı ağ	Kodçözücü/ Üretici ağ	Ayrımcı ağ		
PointGen [123]	2B RGB	2B kodlayıcı	3B kodçözücü	–	2B→z→3B	Yeniden yapılandırma hatası
PointGAL [124]	• Nokta bulutu • 2B RGB	2B kodlayıcı	1B Kodçözücü	• MLP 2B CNN (VGG16)	• 2B projeksiyon • 2B→z→nokta bulutu→3B nokta bulutu + 2B→0/1	• Çok-açıdan Geometrik Maliyet Fonksiyonu (GAL) • Şartlı çekilmeli eğitim maliyeti • Champer uzaklığı
Mv3D [129]	2B RGB	2B kodlayıcı	2B kodçözücü	–	2B→z→2B görüntüler→3B	• L1 uzaklığı • L2 uzaklığı
OGN [130]	• Octree • 2B RGB	2B kodlayıcı	3B kodçözücü	–	2B→z→3B	Karşı entropi
3D-INN [132]	2B RGB	2B kodlayıcı	3B yorumlayıcı (3B kodçözücü)	–	• 2B projeksiyon • 2B→ 2B ananokta→3B iskelet → 2B ananokta	Yeniden yapılandırma hatası
Pixel2Mesh [133]	• 2B RGB • 3B mesh	• 2B kodlayıcı (VGG16) • 3B kodlayıcı (ResNet benzeri)	3B kodçözücü (ResNet benzeri)	–	• 2B→ görüntü öznitelikleri • 3B→3B	• Chamfer maliyet • Normal maliyet düzenlileştirme
SurfNet [139]	2B RGB	2B kodlayıcı	3B kodçözücü	–	2B→z→3B	Şekil farkındalık maliyeti
ShapeMVD [140]	2B iskelet görüntüsü	2B kodlayıcı	3B kodçözücü (U-Net benzeri)	3B CNN	2D→görüş haritaları→3B nokta bulutu → mesh	• Normal maliyet • Çekişmeli eğitim maliyeti • Piksel tabanlı derinlik maliyeti • Maskeler arası benzerlik
U-MNIST3D [141]	• 3B voksel • 2B RGB	LSTM	LSTM	–	• 2B gözetimli öğrenme • 2B+3B→z→3D→2D	• KL ıraksama • Piksel tabanlı yeniden yapılandırma
PixVoxView [142]	• 2B RGB • 2B silüet	2B kodlayıcı (çok girişli)	• 3B kodçözücü • 2,5B kodçözücü	-	• 2B RGB→2B derinlik/silüet>3B/ 2,5B	• Lojistik maliyet • MSE
Im2Struct [143]	2B RGB	• 2B kodlayıcı • VGG16 öznitelilk çıkarımı • 2B maske öznitelikleri	• Yapısal kodçözücü • Komşuluk kodçözücü • Simetri kodçözücü • Kutu kodçözücü	–	• 2B→2B maske • 2B→Kübik yapı	• Yeniden yapılandırma • Karşı entropi

İncelenen voksel tabanlı çalışmalar ise üç boyutlu gözetimli öğrenmeye dayalı modeller ve görüş alanına dayalı çalışmalar olarak iki kategoride incelenmiştir.

2.3.1. Voksel tabanlı üç boyutlu gözetimli öğrenmeye dayalı modeller

Literatürdeki ilk çalışmalar, GAN olarak bilinen modelin iki boyutlu görüntü oluşturmadaki başarısından etkilenilerek modelin 3 boyut için uyarlandığı çalışmalar olmuştur. Bu çalışmaların ilki olan üç boyutlu Üretici Çekişmeli Ağ (3DGAN) [50], GAN modelinin üç boyutlu versiyonu olarak önerilmiştir. Geleneksel GAN modeline benzer şekilde rastsal bir değeri girdi olarak alınarak üç boyutlu üretici ile üç boyutlu ayrımcı ağdan oluşan model ile üç boyutlu nesneler elde edilmiştir.

Volümetrik Evrişimsel Gürültü-giderici Otokodlayıcı (VConvDAE) [51] modeli, DAE olarak adlandırılmış gürültü giderici otokodlayıcının üç boyuta taşınması ile volümetrik verilere uygulandığı modeldir. TL-network [52], üç boyutlu nesneler ile bu nesnelerin iki boyutlu görüntülerinden elde edilen düşük boyutlu özniteliklerden yeniden üç boyutlu nesne oluşturmaya dayanmaktadır. Eğitim süresince üç boyutlu ve iki boyutlu kodlayıcılardan elde edilen öznitelikler bir benzerlik ölçütüne göre yakın değerde olacak şekilde model parametreleri güncellenerek model eğitilmiş; test sırasında ise, üç boyutlu kodlayıcı olmaksızın, eğitim aşamasında optimize edilen üç boyutlu üretici ağ ile iki boyutlu görüntüden çıkarılan özniteliklerden oluşturulmuştur. VConvDAE, Üç Boyutlu Değişimsel Otokodlayıcı Üretici Çekişmeli Ağ (3D-VAE/GAN) (3DGAN modelinin üç boyutlu geri-çatım için kullanılan versiyonu) ve TL-Network modelleri ilk evrişimsel volümetrik otokodlayıcı modelleri olmuştur.

Benzer şekilde Voxception Artık Ağı (VRN) [53], voksel geri-çatımı için önerilen volümetrik bir değişimsel otokodlayıcı modeli olmuştur. SNAP [54] modeli, interaktif yapısı sebebiyle mevcut üç boyutlu gözetimli öğrenmeye dayalı modellerden farklı bir model olarak dikkat çekmiştir. Modelde, kullanıcıdan alınan SNAP olarak adlandırılan istem ile bir projeksiyon operatörü aracılığıyla voksellerden elde edilen gizli kod kullanılarak farklı üç boyutlu şekil manifoldlarından örnekleme yapılabilmiştir.

Çok-açıdan Ayrıştıma (MVD) modeli [144], diğer modellerden farklı olarak yüksek çözünürlüklü üç boyutlu nesneler oluşturmada kullanılmıştır. Modelde, düşük boyutlu ve üst-

örnekleme ile elde edilen nesnelerden çıkarılan ortografik derinlik haritalarından (ODM) yüksek çözünürlüklü nesneler elde edilmiştir.

2.3.2. Voksel tabanlı görüş alanına dayalı modeller

Üç boyutlu geri-çatım problemi için yapılan ilk çalışmalarda üç boyutlu gözetimli öğrenme tercih edilse bile gerçek hayatta üç boyutlu nesnelerin yer gerçekliği verilerinin her zaman mevcut olamayacağı için bu modeller pratikte uygulanabilir olamamaktadır. Bu nedenle, görüş alanına dayalı çalışmalarda üç boyutlu nesneler yerine nesnelerin görüntülerinden üç boyutlu nesneler elde edilmeye çalışılmaktadır. Bu çalışmalarda üç boyutlu nesneler, tek bir görüş alanından alınan görüntülerden elde edilebileceği gibi aynı nesnenin birden fazla görüş alanından alınan birden fazla görüntüsünden de oluşturulabilmektedir.

Voksel tabanlı birden fazla görüş alanına (multi-view) dayalı modeller

Voksel tabanlı modellerde mevcut olan ölçeklenebilirlik problemi nedeniyle modellerin yüksek boyutlu çok sayıda şekil gösterimine gereksinim duyması görüş alanına dayalı modellerin ortaya çıkmasına neden olmuştur [131]. Tek görüş alanından alınan görüntüler yerine aynı nesnenin çok sayıda görüş alanından görüntülerinden nesne oluşturmak daha kolay olacağı için iki boyuttan üç boyuta geri-çatım yapmak üzere nesnenin birden çok görüşten toplanan görüntülerinin modellenmesi üzerine çalışmalara ağırlık verildiği görülmüştür. Bu alandaki ilk çalışmalardan biri olan üç boyutlu Tekrarlayan Yeniden yapılandırma Yapay Sinir Ağı (3DR2N2) [58], iki boyutlu bir kodlayıcı, üç boyutlu evrişimsel LSTM ve üç boyutlu kod çözücü ile bir ya da birden çok görüş alanından oluşan görüntülerden üç boyutlu nesne geri-çatımını gerçekleştirmektedir. Soltani ve arkadaşları [59], tek ya da az sayıdaki bakış açısından elde edilen derinlik görüntülerinden üç boyutlu nesneleri modellemeye odaklanmıştır. Derinlik ve silüet görüntülerini elde etmek için ise modelin ilk adımlarında AllVPNet, DropoutNet ve SingleVPNet ağları kullanılmıştır. Projeksiyon Üretici Çekişmeli Ağ (PrGAN) [60] üç boyutlu üretici ağ, görüş açısı üretici, izdüşüm modülü ve iki boyutlu ayrımcı ağdan oluşan yapısı ile üç boyutlu nesne, görüş açısı tahmini ve yeni görüş alanından iki boyutlu görüntüleri oluşturabilmektedir. McRecon [61] modeli, ray-trace biriktirme ve ön plan maskelerinin eğitim sırasında kullanılması ile üç boyuttan ön plan maske dönüşümünü gerçekleştirebilmektedir. SilNet [62] modelinde, bir veya birden fazla açıdan toplanmış görüntülerden üç boyutlu kod çözücü ile üç boyutlu nesneler, iki boyutlu kod

çözücü ile yeni açılardan görüntüler oluşturulabilmiştir. Bir projeksiyon katmanı vasıtasıyla oluşturulan üç boyutlu nesneler iki boyutlu siluetlere dönüştürülerek eğitim üç boyutlu gözetimli öğrenme yapılmaksızın tamamlanmıştır.

Voksel tabanlı tek görüş alanına (single-view) dayalı modeller

Gerçek hayatta karşılaşılan problemlerde bir nesnenin farklı açılardan görüntüleri elimizde her zaman mevcut olamamaktadır. Bu nedenle, son zamanlarda, tek açıdan alınan görüntülerden üç boyutlu nesne üretme ve geri-çatımına odaklanılmaktadır.

3D-VAE/GAN, 3DGAN modelinin bir varyasyonu olarak tek bakış açısından elde edilen görüntülerden üç boyutlu geri-çatım oluşturmak üzere kullanılmıştır. Alınan görüntülerden bir kod çözücü ile olasılıksal gizli vektör uzayı öğrenilerek bu uzaydan üç boyutlu nesneler örneklenmiştir.

Bu çalışmayı takip eden diğer bir çalışmada [145] ise bir yığın hiyerarşik ağ ile iki boyutlu görüntülerden üç boyutlu geri-çatım yapılmıştır. Bir otokodlayıcı olan Perspektif Dönüştürücü Ağ (PTN) modelinde [146], üç boyutlu yer-gerçekliği verilerine gerek duyulmaksızın tek görüş açısından alınan iki boyutlu görüntülerden elde edilen üç boyutlu nesnelerin iki boyuta perspektif dönüşüm katmanı ile dönüştürülmesi sonucunda oluşturulan iki boyutlu iz-düşüm görüntüleri ile eğitim gerçekleştirilerek üç boyutlu nesneler üretilmiştir. 3DensiNet [147] modeli, girdi olarak alınan iki boyutlu görüntülerden çıkarılan yoğunluk ısı haritası tabanlı bir öğrenmeye dayalı olarak iki boyuttan üç boyuta dönüşüm işlemini gerçekleştirmektedir. Görüntülerin poz bilgilerini kullanarak tek açıdan görüntüden üç boyutlu nesne oluşturmaya dayanan 3d-recon [148] modelinde birden çok açıdan görüntü mevcut olduğu durumlarda tanımlanan poz bağımsız maliyet fonksiyonuna ve projeksiyon ile elde edilen iki boyutlu görüntülere göre model eğitilerek nesneler elde edilmiştir. Son çalışmalardan biri olan Renkli Voksel Ağı (CVN) [149] modelinde, literatürdeki diğer çalışmalardan farklı olarak, renkli üç boyutlu obje geri-çatımı için uçtan-uca bir ağ yapısı kullanılarak iki boyuttan üç boyuta akış nesnesi ile renk nesnesi füzyonu gerçekleştirilmiştir. Pix3D [150] modelinde yeni bir otokodlayıcı ağı ve görüntü-obje çifti veriseti olarak sunulmuştur. Pix3D modeli, iki boyutlu görüntülerden iki buçuk boyutlu taslakların çıkarılabilmesi için iki buçuk boyutlu taslak tahmin ağı, elde edilen taslakların gizli kodlara dönüştürülebilmesi için iki buçuk boyutlu taslak kodlayıcı ağı ve gizli kodlardan üç boyutlu

nesnelerin elde edilebilmesi için ise üç boyutlu şekil kodçözücü ağından oluşmaktadır.

Çizelge 2.4. Voksel tabanlı incelenen modellerin detaylı karşılaştırmaları

Model	Girdi	Model bileşenleri			Projeksiyon/ Model özeti	Maliyet fonksiyonu
		Kodlayıcı ağ	Kodçözücü/ üretici ağ	Ayrımcı Ağ		
3DGAN [50]	• 2B RGB • Rastsal gizli kod	2B kodlayıcı	3B kodçözücü	3B CNN	2B→z→3B	• KL ıraksama • L2 üretici maliyeti • Çekişmeli maliyet fonksiyonu
VConvDAE [51]	3B voksel	3B kodlayıcı	3B kodçözücü	-	3B→z→3B	• MSE • Karşı entropi
TL-Network [52]	• 3B voksel • 2B RGB	• 3B kodlayıcı 2B kodlayıcı	3B kodçözücü	-	• Eğitim: 3B→1B→3B 2B→1B • Test: 2B→1B 1B→3B	• Sigmoid karşı entropi • L2 uzaklığı
VRN [53]	3B voksel	3B kodlayıcı	3B kodçözücü	-	3B→z→3B	• Ağırlıklandırılmış BCE • KL ıraksama • Yeniden yapılandırma maliyeti
SNAP [54]	3B voksel	3B kodlayıcı	3B kodçözücü	3B CNN	3B→z→3B	• Öznitelik benzerlik maliyeti • Çekişmeli eğitim maliyeti
3DIWGAN [55]	2,5B	2B kodlayıcı	3B kodçözücü	3B CNN	2,5B →z→3B→0/1	• WGAN maliyeti • L2 yeniden yapılandırma • KL ıraksama • L2 üretici maliyeti
3DRecGAN [56]	2,5B	2B kodlayıcı (U-Net)	3B kodçözücü (U-Net)	3B CNN	2,5B →z→3B	• BCE • WGAN-GP
3DRecGAN++ [57]	2,5B	2B kodlayıcı (U-Net)	3B kodçözücü (U-Net)	3B CNN CGAN	2,5B →z→3B→ Süper çözünürlüklü 3B	• BCE • WGAN-GP
3DR2N2 [58]	2B RGB	2B kodlayıcı	3B kodçözücü	-	• 3B evrişimsel LSTM • 2B→z→3B	• 3B voksel tabanlı softmax
SingleVPNet [59]	2B derinlik haritaları/ silüet	2B kodlayıcı	2B kodçözücü	-	2B→z,c→2B→ (füzyon ile) 3B nokta bulutu	• KL ıraksama • Piksel tabanlı yeniden yapılandırma • Sınıflandırma
PrGAN [60]	• 2B silüet • Rastsal gizli kod	-	3B kodçözücü/ üretici	2B CNN	• Projeksiyon modülü z→3B→2B→0/1	Çekişmeli maliyet fonksiyonu
McRecon [61]	2B RGB	2B kodlayıcı	3B kodçözücü	3B CNN	Ray-trace biriktirme fonksiyonu ile 2B maske gözetimli öğrenme	• Çekişmeli eğitim maliyeti • Projeksiyon maliyeti • Yeniden yapılandırma

Çizelge 2.4. (Devam) Voksel tabanlı incelenen modellerin detaylı karşılaştırmaları

Model	Girdi	Model bileşenleri			Projeksiyon/ Model özeti	Maliyet fonksiyonu
		Kodlayıcı ağ	Kodçözücü/ üretici ağ	Ayrımcı ağ		
SilNet [62]	2B RGB	2B kodlayıcı	• 2B kodçözücü • 3B kodçözücü	-	2B→z→3B→2B	Sınıflandırma doğruluk oranı
MVD [144]	• 2B RGB • 3B Voksel	• 2B kodlayıcı • 3B kodlayıcı	3B kodçözücü	-	• 2B→z→3B • 3B→ODM→SÇ ODM • 3B + SÇ ODM→ SÇ 3B	ODM silüet benzerliği
SSNet [145]	2B silüet	3B kodlayıcı	3B kodçözücü	-	2B→3B→z→3B→z> 3B	Yeniden yapılandırma hatası
PTN [146]	2B RGB	2B kodlayıcı	3B kodçözücü	-	• Perspektif dönüşüm ile 2B gözetimli öğrenme • 2B→z→3B→2B	L2 uzaklığı
3DensiNet [147]	• 2B sıcaklık haritası • 2B RGB	2B kodlayıcı	• 2B kodçözücü • 3B kodçözücü	3B CNN	2B→2B sıcaklık haritası→3B→0/1	• MSE • Karşı entropi • Çekişmeli eğitim maliyeti
3d-recon [148]	• 2B silüet • Rastsal gizli kod • Poz bilgisi	2B kodlayıcı	3B kodçözücü	2B CNN	• 2B→z, p→ 3B→2B • 2B→0/1	• Yeniden yapılandırma hatası • Poz bağımsız maliyet fonksiyonları
CVN [149]	2B RGB	2B kodlayıcı	• 3B kodçözücü (geometri) • 3B kodçözücü (renk)	-	• 2B→z→3B • 3B→2B perspektif projeksiyon ile renk öğrenme	• L2 uzaklığı o renk regresyon o renk eşleme o görüntü akış

2.3.3. Voksel tabanlı 2,5B derinlik görüntülerine dayalı modeller

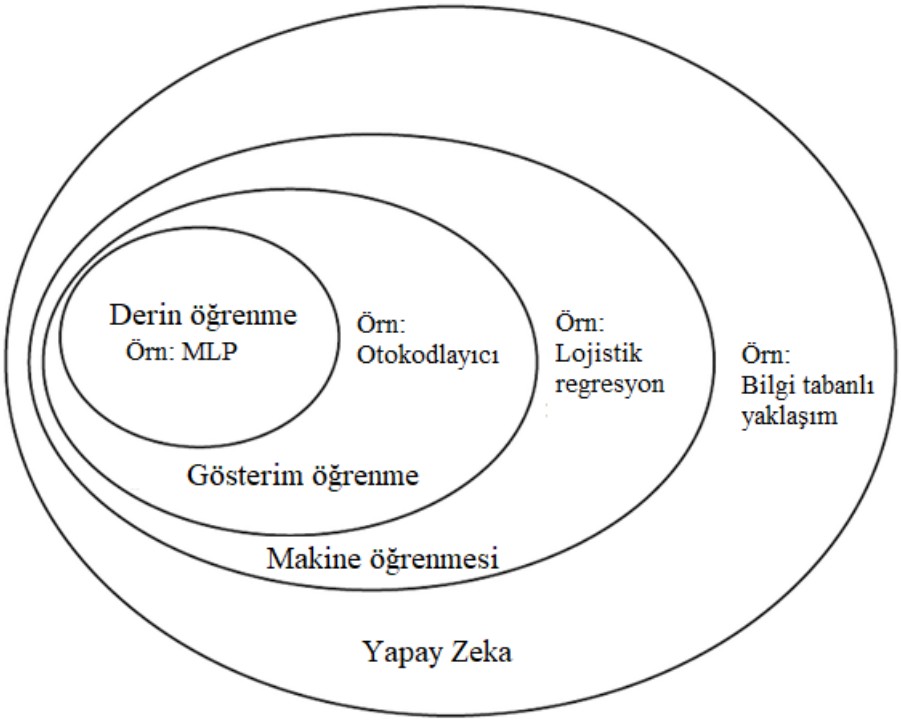
Üç boyutlu Geliştirilmiş Wasserstein Üretici Çekişmeli Ağ (3DIWGAN) [55] modelinde nesneler 2,5B derinlik görüntülerinden Wasserstein metriği [47] tabanında çekişmeli eğitim ile elde edilmiştir. Benzer şekilde 2,5B görüntülerden üç boyutlu nesne oluşturmak üzere 3DRecGAN [56] modelinde CGAN ile otokodlayıcı bir araya getirilerek U-Net benzeri bir model ile tek bakış açısından elde edilen derinlik görüntülerinden üç boyutlu nesneler oluşturulabilmiştir. 3DRecGAN modelinin geliştirilmiş versiyonu olan 3DRecGAN++ [57] modelinde ise önceki modele ek bir üst-örnekleme modülü ile derinlik görüntüleri, literatürde mevcut modeller ile elde edilen nesnelere (32x32x32) göre çok daha yüksek çözünürlüklü,

(256x256x256) volümetrik verilere dönüştürülebilmiştir.

3DShapeNets [151] modelinde, bir evrişimsel DBN [152] ile 2,5B derinlik görüntüleri üç boyutlu volümetrik şekillere dönüştürülerek nesne tanıma, nesne geri-çatımı ve şekil tamamlama problemleri çözülmeye çalışılmıştır

3. TEMEL MODELLER

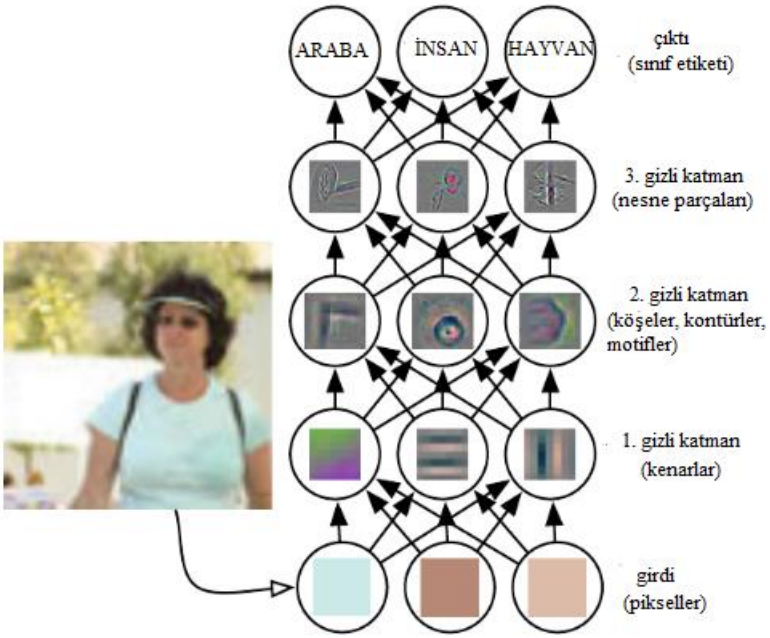
Makine öğrenmesi teknolojileri, web aramadan içerik filtreleme ve tavsiye sistemlerine kadar birçok uygulamasıyla modern dünyada oldukça yaygın hale gelmiştir. Geleneksel yöntemler, verileri ham olarak alıp işleme konusunda yetersiz kalmıştır. Bu yaklaşımlarda alan uzmanı kişilerce çıkarılan özniteliklere gereksinim duyulması nedeniyle öznitelik mühendisliği kavramı söz konusu olmuştur. Gösterim öğrenme olarak literatürde geçen kavram ise bir seri yöntem kullanılarak ham veriden gösterim olarak nitelendirilen özniteliklerin çıkarılmasına dayanmaktadır. Bu kavram ile derin öğrenme ve makine öğrenme kavramlarının ilişkisi Resim 3.1'de verilmiştir.



Resim 3.1. Derin öğrenme, makine öğrenmesi ve gösterim öğrenme ilişki ven şeması [153]

Derin öğrenme yaklaşımlarında, doğrusal olmayan fonksiyonların bir kombinasyonu ile gösterimler ham veriden doğrudan elde edilebilmektedir. Her bir fonksiyon ile elde edilen gösterimler, farklı seviyede dönüşüm işlemleri ile elde edilen yeni özniteliklere karşılık gelmektedir. Ağ modellerinin katmanlı yapısı sebebiyle düşük seviyeli özniteliklerden yüksek seviyeli özniteliklere dönüşüm gerçekleştirilebilmektedir. Klasik sığ modeller ile sadece düşük seviyeli öznitelikler elde edilebilirken derin öğrenme yaklaşımları ile sınıflandırma gibi problemler için daha ayırt edici olan yüksek seviyeli öznitelikler ile ara-

seviye öznitelikler de elde edilebilmektedir. Örnek olarak, Resim 3.2'de verildiği gibi, piksellerden oluşan bir görüntü düşünüldüğünde, ilk katmanda öğrenilen gösterimler kenarlar, ikinci katmanda görüntünün içerdiği motifler, daha üst katmanlarda ise nesne parçaları olacaktır.

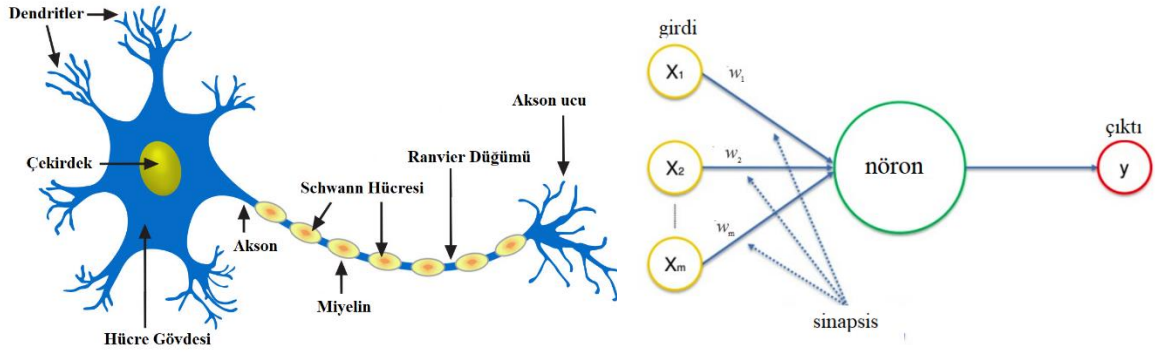


Resim 3.2. Derin ağ modeli ile her katmanda öğrenilen gösterimler [154]

3.1. Derin İleri beslemeli Sinir Ağı

Çok Katmanlı Perseptron (MLP) ya da ileri beslemeli yapay sinir ağı olarak da bilinen Derin İleri-beslemeli Sinir Ağı (DFNN), x giriş verilerinden y çıkış verisini elde etmek üzere parametresi W olan f fonksiyonunu öğrenerek bilinmeyen bir f^* fonksiyonuna yakınsamayı hedeflemektedir. Ayrık değerlerden oluşan y için ele alınan problem bir sınıflandırma problemi iken bu değerler sürekli olduğunda problem regresyon problemi olarak adlandırılmaktadır. Bu ağ modeli, giriş verilerinden çıkış verisine doğru bir graf şeklinde olduğundan çıkış ile giriş arasında bir geri-besleme bağlantısına sahip olmadığı için ileri-beslemeli ağ olarak ifade edilmiştir. DFNN'in ağ olarak isimlendirilmesi, birden fazla katmandan (L) oluşan yapısından kaynaklanmaktadır. Zaman serisi şeklindeki veriler için tasarlanan RNN ağında geri-beslemeli bağlantılar ile seriler arasında da bağlantılar kurulmuştur.

Resim 3.3'te verilen sinir ağları olarak bilinen yapılar, biyolojik nöron yapısına benzer yapıda olmaları nedeniyle bu şekilde adlandırılmıştır. Ağı oluşturan her bir düğüm nöronlardan, her bir bağlantı ise nöronların sinapsislerinden esinlenilmiştir.



(a) Biyolojik nöron yapısı [155] (b) Nöron yapısı [156]

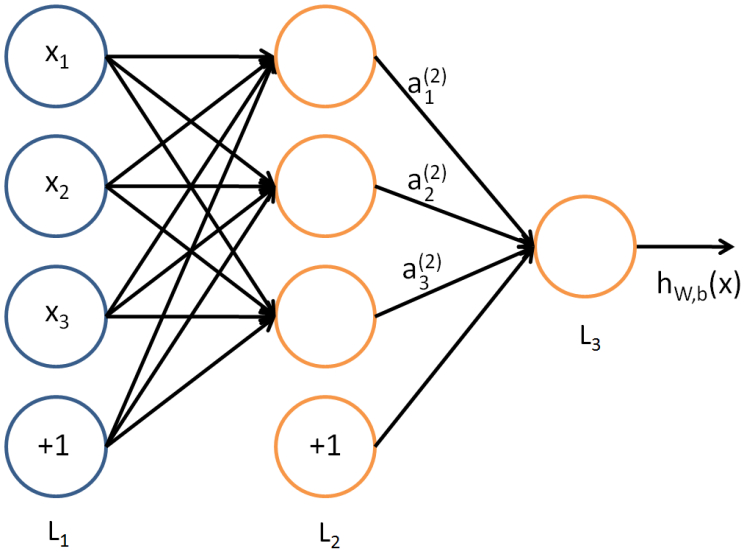
Resim 3.3. Biyolojik nöron ve yapay sinir ağı nöron ilişkisi

Resim 3.3 (b)'de verilen bir düğüm örneği için, nöron olarak ifade edilen, x girdi değerlerini alarak y çıkışını üretebilen bir hesaplama birimidir. $\hat{y}$ çıkışı, m girdi sayısı olmak üzere $f(W^T x) = f(\sum_{i=1}^{m} w_i\, x_i + b)$ fonksiyonu ile girdilerin ağırlıklandırılmış toplamına bias değeri eklenerek ifade edilir. $f: \mathbb{R} \to \mathbb{R}$ fonksiyonu, aktivasyon fonksiyonu olarak adlandırılmıştır.

Yapay sinir ağı olarak ifade edilen ağ modeli, her bir katmanı birden fazla nörondan oluşan çok katmanlı ağ modelidir. Her bir katman farklı bir fonksiyon olarak düşünüldüğünde, f fonksiyonu Eş. 3.1'de gösterildiği gibi farklı fonksiyonların bir kombinasyonu olarak değerlendirilebilmektedir.

$$f(x; W, b) = f_3(f_2(f_1(x; W^{(1)}, b^{(1)}); W^{(2)}, b^{(2)}); W^{(3)}, b^{(3)}) \tag{3.1}$$

Eşitlikte verildiği gibi, 3 katmandan oluşan basit bir ağ için f_1 giriş katmanı, f_2 gizli katmanı, f_3 ise son katman olan çıktı katmanını ifade etmektedir. Katman sayısı, n_l, ağın derinliği olarak da ifade edilmektedir. Katman sayısı ne kadar artarsa elde edilen ağ o kadar derin ve karmaşık olmaktadır. Bu nedenle çok katmandan oluşan ağ modelleri derin ağ olarak adlandırılmaktadır.



Resim 3.4. Örnek yapay sinir ağı [157]

Yapay sinir ağlarını matematiksel olarak açıklayabilmek üzere Resim 3.4'te örneği verilen model için $n_l = 3$ ve her bir katmandaki düğüm sayısı $s_l = 3$ olmak üzere, $W^{(l)}$ olarak verilen l. katmana ait parametreleri düğümlerle ilişkilendirebilmek için l. katmandaki i. düğüm ile $(l+1)$. katmandaki j. düğümü arasındaki ağırlık $W_{ij}{}^{(l)}$ şeklinde adlandırılmıştır. Ele alınan örnek için $W^{(1)} \in \mathbb{R}^{3\times3}$, $W^{(2)} \in \mathbb{R}^{3\times1}$. Bu örnek için l. katmanın i.ci düğümünü ifade eden $a_i{}^{(l)}$ aktivasyon fonksiyonu $i = 1$ ve $l = 2$ için Eş. 3.2'de gösterildiği şekilde hesaplanmaktadır. Resim 3.4'te verilen ağ için $\hat{y}$ çıkışı ise Eş. 3.2-3.5'te gösterildiği gibi adım adım hesaplanabilmektedir.

$$a_1{}^{(2)} = f(W_{11}{}^{(1)}x_1 + W_{12}{}^{(1)}x_2 + W_{13}{}^{(1)}x_3 + b_1{}^{(1)}) \tag{3.2}$$

$$a_2{}^{(2)} = f(W_{21}{}^{(1)}x_1 + W_{22}{}^{(1)}x_2 + W_{23}{}^{(1)}x_3 + b_2{}^{(1)}) \tag{3.3}$$

$$a_3{}^{(2)} = f(W_{31}{}^{(1)}x_1 + W_{32}{}^{(1)}x_2 + W_{33}{}^{(1)}x_3 + b_3{}^{(1)}) \tag{3.4}$$

$$\hat{y} = a_1{}^{(3)} = f(W_{11}{}^{(2)}a_1{}^2 + W_{12}{}^{(1)}a_2{}^2 + W_{13}{}^{(1)}a_3{}^2 + b_1{}^{(2)}) \tag{3.5}$$

Daha basit şekilde ifade edebilmek üzere $z_i{}^l$, i. düğümdeki girdinin ağırlıklandırılarak bias eklenmesi şeklinde ifade edildiğinde Eş. 3.6 ile elde edilir.

$$z_i^{(2)} = \sum_{j=1}^{n} W_{ij}^{(1)} x_j + b_i^{(1)} \tag{3.6}$$

Böylece her bir katmanın çıktısı olan $a_i^{l} = f(z_i^{(l)})$ olarak basit şekilde ifade edilmektedir. Daha kompakt şekilde ifade edilmek istenirse, $\hat{y}$ çıkışı Eş. 3.7-3.10'da gösterildiği gibi elde edilir.

$$z^{(2)} = W^{(1)}x + b^{(1)} \tag{3.7}$$

$$a^{(2)} = f(z^{(2)}) \tag{3.8}$$

$$z^{(3)} = W^{(2)}a^{(2)} + b^{(2)} \tag{3.9}$$

$$\hat{y} = a^{(3)} = f(z^{(3)}) \tag{3.10}$$

Daha genel olarak Eş. 3.11 ve 3.12'de gösterildiği şekilde, $a^{(1)} = x$ olmak üzere $(l+1)$. katman çıktısı Eş. 3.11 ve Eş. 12 ile hesaplanmaktadır.

$$z^{(l+1)} = W^{(l)}a^{(l)} + b^{(1)} \tag{3.11}$$

$$a^{(l+1)} = f(z^{(l+1)}) \tag{3.12}$$

Yapay sinir ağlarında geri-besleme yapabilmek üzere maliyet fonksiyonu, $\{(x_1, y_1), \ldots ((x_m, y_m)\}$ eğitim kümesi için kolit (L2) metriğine göre Eş. 3.13'de verilen fonksiyon ile elde edilir.

$$J(W, b, x, y) = \frac{1}{2} \|y - \hat{y}\|^2 \tag{3.13}$$

Maliyet fonksiyonu, düzenlileştirme cezası (regularization penalty) eklendiğinde, λ ağırlık düşürme parametresi ile Eş. 3.14'de verildiği gibi hesaplanmaktadır.

$$J(W, b) = \left[\frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\|\hat{y}_i - y_i\|^2\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}(W_{ji}^{\ l})^2 \tag{3.14}$$

Eş. 3.14'te verilen maliyet fonksiyonu ise regresyon problemleri için tanımlanan genel bir gösterimdir.

$$J(W, b, x, y) = \frac{1}{2} \|1 - y\hat{y}\|^2 \tag{3.15}$$

Ele alınan problem ikili bir sınıflandırma problemi ise Eş. 3.15 ile tanımlanabilmektedir. Birden çok (n) sınıflı problemler için ise "Softmax" olarak bilinen bir normalizasyon fonksiyonu kullanılmaktadır.

$$q_c(x) = \frac{e^{f_c(x)}}{\sum_{c=1}^{n} e^{f_c(x)}} \tag{3.16}$$

Eş. 3.16'da verilen softmax düzenlileştirme (normalizasyonu) ile elde edilen $q_n(x) \epsilon (0,1]$.

Softmax düzenlileştirmeye dayalı maliyet fonksiyonu çapraz entropi (cross entropy) olarak adlandırılmıştır. Çapraz entropi maliyet fonksiyonu Eş. 3.17'de verilmiştir.

$$J(W, b, x, y) = -\sum_{c=1}^{n} y_c \log q_c(x) \tag{3.17}$$

3.1.1. Geri-besleme

Derin ileri beslemeli ağlarda hedeflenen, maliyet fonksiyonunun W ve b parametre değerlerine bağlı olarak minimize edilmesidir. Bu parametrelerin ilk değer ataması yapıldıktan sonra ağın eğitilmesi için gradyan inişi (gradient descent) eniyileme gibi optimizasyon algoritmaları kullanılmaktadır. Gradyan inişi gibi eniyileme yaklaşımları oldukça başarılı olmalarına rağmen, $J(W, b)$ konveks olmayan bir fonksiyon olduğu için optimizasyon algoritmaları, yerel minimumu global minimum zannedebilmektedir. Gradyan inişi eniyilemeye göre her bir iterasyonda W ve b parametreleri Eş. 3.18 ve 3.19'te verildiği gibi α öğrenme oranına göre güncellenmektedir.

$${W_{ij}}^l := {W_{ij}}^l - \alpha \frac{\partial}{\partial {W_{ij}}^l} J(W, b) \tag{3.18}$$

$$b_i{}^l := b_i{}^l - \alpha \frac{\partial}{\partial b_i{}^l} J(W, b) \tag{3.19}$$

Geri-besleme algoritması için Eş. 3.18 ve 3.19'da hesaplanması gereken kısmi türevler x ve y değerlerine bağlı olarak Eş. 3.20 ve 3.21'de verilmiştir.

$$\alpha \frac{\partial}{\partial W_{ij}{}^l} J(W, b) = \left[\frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial W_{ij}{}^l} J(W, b; x_i, y_i) \right] + \lambda W_{ij}{}^l \tag{3.20}$$

$$\alpha \frac{\partial}{\partial b_i{}^l} J(W, b) = \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial b_i{}^l} J(W, b; x_i, y_i) \tag{3.21}$$

Geri-besleme algoritmasına ait sözde kod Resim 3.5'te verilmiştir.

1: **procedure** GERI-BESLEME (x, y)
2: Eğitim kümesi $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$
3: $L \leftarrow$ katman sayısı
4: $W^{(1)}, \ldots, W^{(L)}$ ilk değer atamaları yapılır
5: $b^{(1)}, \ldots, b^{(L)}$ ilk değer atamaları yapılır
6: **for** $i = 1{:}\, m$ **do**
7: $a^{(1)} \leftarrow x^{(i)}$
8: İleri-besleme ile $l = 2, 3, \ldots, L$ için $a^{(l)}$ hesaplanır
9: $y^{(i)}$ etiketi ile $\delta^{(L)} = a^{(L)} - y^{(i)}$ hesaplanır
10: $\delta^{(L-1)}, \delta^{(L-2)}, \ldots, \delta^{(2)}$ hesaplanır
11: $\frac{\partial}{\partial W^{(l)}} J(W, b; x, y) = a^{(l)} \delta^{(l+1)}$, her katman için hesaplanır
12: $\frac{\partial}{\partial b^{(l)}} J(W, b; x, y) = \delta^{(l+1)}$, her katman için hesaplanır
13: **end for**
14: **end procedure**

Resim 3.5. Geri-besleme algoritması sözde kodu

3.1.2. Kapasite, aşırı-öğrenme, yetersiz-öğrenme ve düzenlileştirme

Makine öğrenmesi yaklaşımlarındaki temel problem sadece eğitim kümesi üzerinde başarılı bir model değil yeni örnekler üzerinde de başarı elde edebilmektir.

Bir makine öğrenmesi yaklaşımının performansının yüksek olabilmesi için eğitim hatasının küçük olması ve eğitim ile test hatası arasındaki farkın az olması beklenmektedir. İlk durum

yani modelin eğitim verisi üzerinde yeterince düşük hataya sahip olması yetersiz-öğrenme (underfitting), eğitim ile test hatasının arasındaki farkın fazla olması durumu ise literatürde aşırı-öğrenme (overfitting) olarak adlandırılmıştır. Modellerin aşırı-öğrenmeye ya da yetersiz-öğrenmeye eğilim göstereceği, kapasite olarak adlandırılan kavramla ilişkilidir. Model kapasitesi, modeli oluşturan fonksiyon çeşitliliğine bağlı olmaktadır. Düşük kapasiteli modeller, eğitim kümesinin modellenmesi konusunda yetersiz olabilmektedir. Yüksek kapasiteli modeller ise eğitim kümesine dayalı parametreleri aşırı-öğrenmeleri nedeniyle test verilerinde yeterince performans sergileyememektedirler [154].

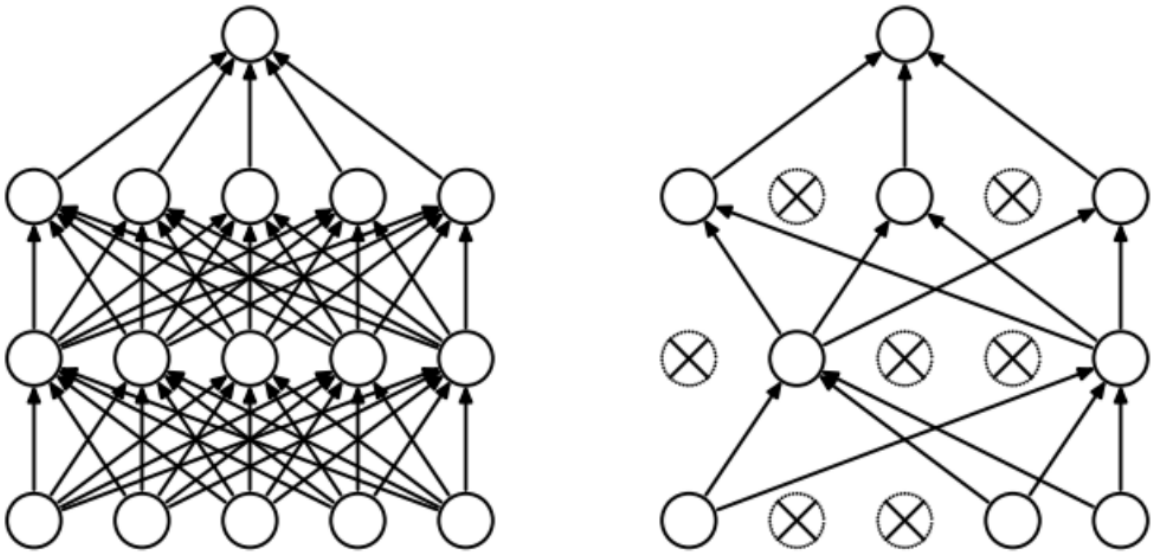
Görülmeyen örnekler üzerinde başarı elde edebilmeye düzenlileştirme (generalization) denilmektedir. Düzenlileştirme hatası (generalization error) ise yeni veriler üzerinde beklenen hatayı ifade etmektedir. Düzenlileştirme hatasını tespit edebilmek üzere, test verisi olarak adlandırılan, farklı verilerden oluşan bir veri kümesi üzerinde yapılan ölçümler kullanılmaktadır [154]. Düzenlileştirme, Eş. 3.20'de verildiği gibi düzenlileştirme cezası (manhattan (L1), Öklid (L2) düzenlileştirme gibi), gürültü ekleme, veri arttırma (data augmentation) ve iletim sönümü (dropout) [153] gibi teknikler ile gerçekleştirilmektedir.

Makine öğrenmesi yaklaşımlarında modellerin genelleştirme kabiliyetini arttırabilmek üzere daha çok veri ile eğitim gerçekleştirilmelidir. Bu nedenle, elimizde daha çok eğitim verisi olmadığı durumlarda, eğitim verisinin veri büyütme yöntemleri ile arttırılması tercih edilmiştir. Nesne tanıma gibi problemlerde oldukça başarılı olan bu yaklaşımla, iki boyutlu dönüşüm işlemleri ile yüksek boyutlu görüntülerden yeni görüntüler elde edilmektedir.

Giriş verisine gürültü eklemek [158] bir çeşit veri büyütme olarak değerlendirilmiştir. Daha önceki araştırmalarda, yapay sinir ağlarının gürültüye karşı dayanıklı oldukları gözlemlenmiştir [159]. Bu nedenle, yapay sinir ağlarının daha gürbüz hale getirilebilmeleri için DAE modelinde olduğu gibi girdi katmanında verilere rastsal gürültü eklenerek modelin eğitilmesi söz konusu olmuştur.

İletim sönümü, modellerin genelleştirme kapasitesini arttırmak üzere önerilen, modellerin hesaplama maliyetini düşüren bir yaklaşımdır. Aşırı öğrenme problemini çözebilmek üzere önerilen iletim sönümü, farklı yapay sinir ağı mimarilerini üstel olarak bir araya getirmeyi mümkün kılmıştır. İletim sönümü kavramı, yapay sinir ağını oluşturan düğümlerden bazıları ve bu düğümler ile daha önceki ve sonraki katmanlar arasındaki bağlantıların çıkarılmasına

dayanmaktadır. İletim sönümünde çıkarılacak düğümlerin hangileri olduğu rastsal olarak seçilmektedir. Daha önceki çalışmalarda da tercih edilen 0,5 gibi sabit bir olasılık değerine göre düğümlerin çıkarılması ile elde kalan düğümler eğitim süresince kullanılmaktadır. Resim 3.6 (b)'da iletim sönümü ile elde edilen yapay sinir ağı örneği verilmiştir.



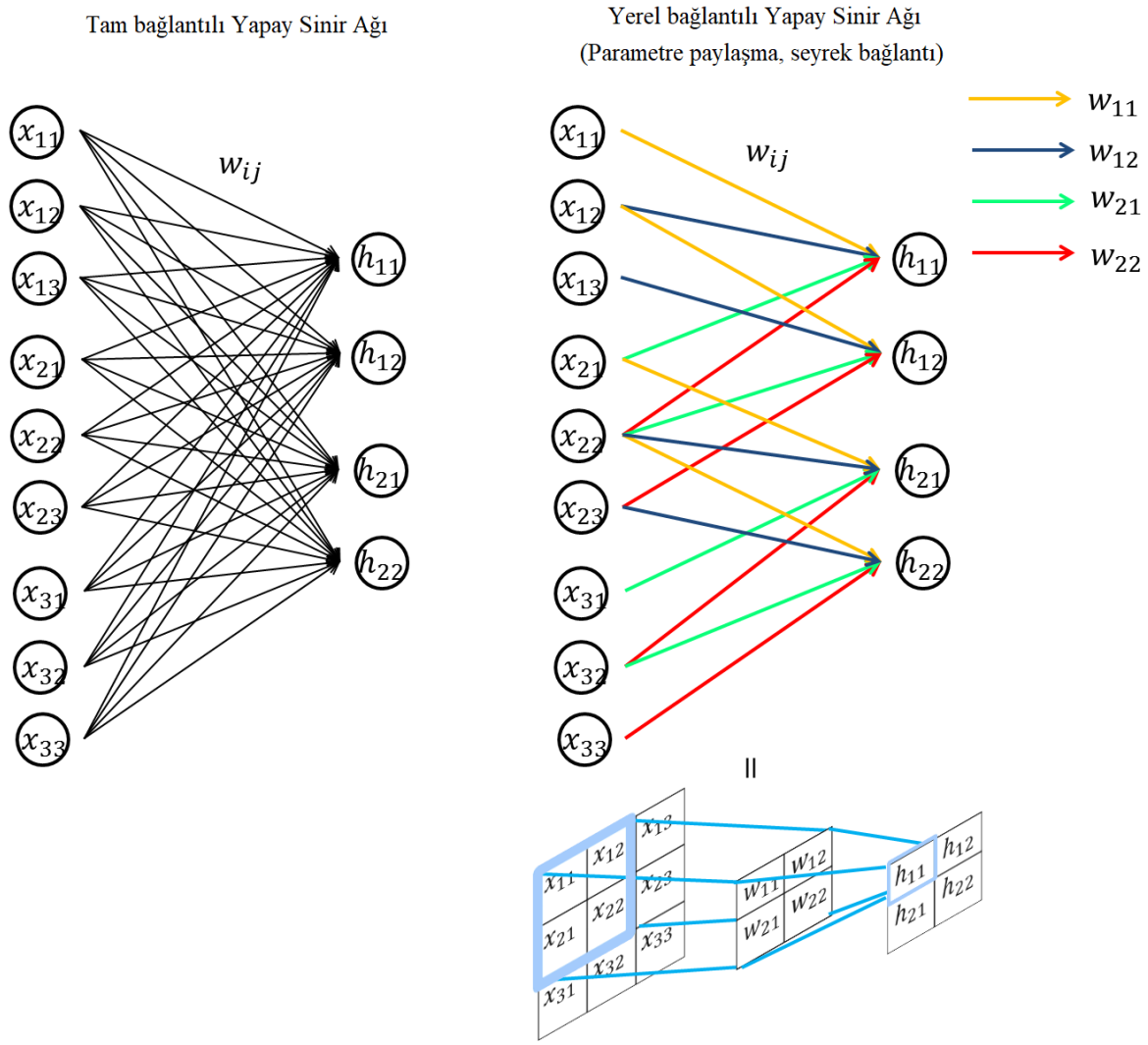
(a) Standart Yapay Sinir Ağı (b) İletim sönümü ile Yapay Sinir Ağı

Resim 3.6. İletim sönümü örneği [153]

3.2. Evrişimsel Sinir Ağı

Evrişimsel Sinir Ağları, ızgara benzeri topolojiye sahip verileri işlemek üzere kullanılan yapay sinir ağlarının özel bir durumudur. Evrişimsel adı, ağ modelini oluşturan evrişim işleminden gelmektedir.

Evrişimsel Sinir Ağlarında kullanılan veriler, zaman serisi şeklindeki bir boyutlu (1B) ızgara şeklinde düşünülebilecek düzenli zaman aralıklarında alınan veriler, görüntü olarak ifade edilen iki boyutlu piksel ızgarası olarak ifade edilen veriler olmaktadır. Evrişimsel ağ, yapay sinir ağındaki genel matris çarpımı yerine evrişim işlemine dayanan ağ modeli olarak ifade edilebilmektedir. Böylece CNN ağları NN ağlarına göre seyrek bağlantı, parametre paylaşımı ve eşdeğer gösterim bakımından üstün olmaktadır.



Resim 3.7. Yapay sinir ağlarından CNN ağlarına geçiş [160]

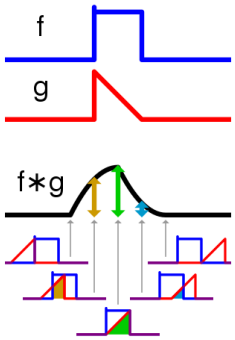
Yapay sinir ağlarından CNN ağlarına seyrek bağlantı ve parametre paylaşımı ile kurulan bağlantı Resim 3.7'de gösterilmektedir. Resimde aynı renkte verilen bağlantılarda aynı parametrelerin kullanıldığı gösterilmektedir. Bu nedenle, CNN modellerinde NN modellerine göre daha az bağlantı ile daha az parametre kullanıldığı görülmektedir.

3.2.1. Evrişim operasyonu

Evrişim işlemi bir çeşit doğrusal operasyondur. Bir fonksiyonun diğerine benzerliğini ölçmek üzere iki fonksiyon bileşenlerinin toplamı ya da integrali olarak tanımlanmıştır. Evrişim işlemi Resim 3.8'de gösterildiği gibi, bir kare dalga sinyalinin x ekseni boyunca kaydırılarak diğer sinyal üzerindeki kare dalgaların bulmasıdır.

Evrişim işleminden önce bir fonksiyon değişkenlerinin t den τ ya dönüştürülerek kaydırma (shift) işlemi gerçekleştirilmektedir. Matematiksel olarak $*$ ile ifade evrişim işlemi, f ve g fonksiyonları t'de iki fonksiyon olmak üzere f ve g'nin sonsuz aralıktaki integrali olarak Eş. 3.22'de verildiği şekilde elde edilmektedir.

.



Resim 3.8. Sinyal üzerinde evrişim işlemi [161]

$$f * g = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{3.22}$$

Eş. 3.22'de verilen fonksiyon için aralık $[0, t]$ sonlu aralığına dönüştürülürse, evrişim işlemi Eş. 3.23'de verildiği şekilde ifade edilebilmektedir.

$$s(t) = [f * g](t) = \int_{0}^{t} f(\tau)g(t-\tau)d\tau \tag{3.23}$$

Ayrık değerlerden oluşan f ve g fonksiyonları ise Eş. 3.24 ile elde edilebilmektedir.

$$s = f * g = \sum_{\tau=-\infty}^{\infty} f(\tau)g(t-\tau) \tag{3.24}$$

Eşitlikte f girdi, g ise kernel yani evrişim filtresi, çıktı s ise öznitelik haritası olarak ifade edilmektedir.

Evrişim işlemi iki boyutlu veriler ele alındığında, $m \times n$ filtre boyutu, I girişi iki boyutlu görüntü, K ise iki boyutlu filtre olmak üzere Eş. 3.25 ve 3.26'da verildiği şekilde

gerçekleştirilmektedir.

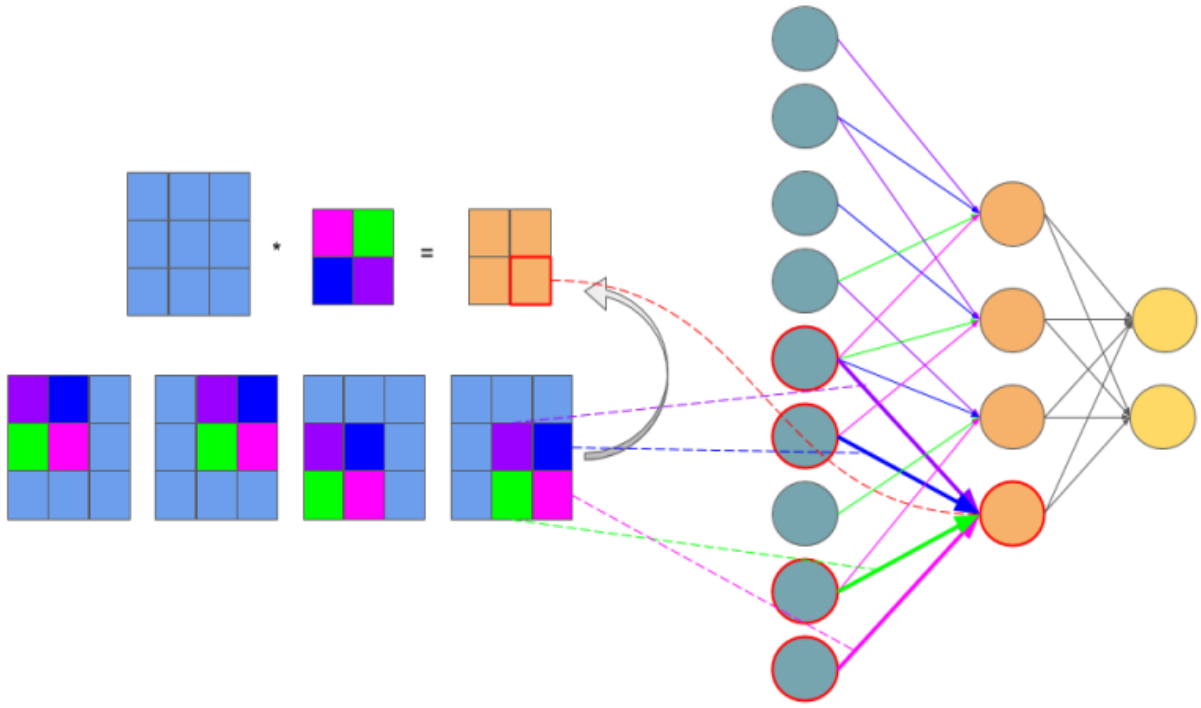
$$s(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{3.25}$$

$$s(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-\ n)K(m,n) \tag{3.26}$$

Yapay sinir ağı uygulamalarında evrişim işlemi, aslında çapraz korelasyon (cross-correlation) olarak ifade edilen, evrişim filtresinin döndürülmeden kullanılmasına denk gelen işlem olarak kullanılmaktadır. Bu şekilde tanımlanan evrişim işlemi Eş. 3.27'de verilmiştir.

$$s(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i+m, j+\ n)K(m,n) \tag{3.27}$$

Resim 3.9'da evrişim operasyonunun iki boyutlu bir görüntü üzerinde uygulanması ve yapay sinir ağı olarak ele alındığında evrişim filtresine karşılık gelen bağlantılar evrişim işlemi sonucunda elde edilen görüntülere karşılk gelen düğümler görselleştirilmiştir.



Resim 3.9. CNN ağlarında evrişim operasyonu ve çok katmanlı yapay sinir ağı ilişkisi [162]

3.2.2. Aktivasyon fonksiyonu

Yapay sinir ağlarında; aktivasyon fonksiyonu olarak sigmoid, hiperbolik tanjant, üstel doğrusal birim (exponential düzeltilmiş doğrusal birim) (ELU) [163], düzeltilmiş doğrusal birim (rectified linear unit) (ReLU), sızan düzeltilmiş doğrusal birim (Leaky ReLU) [164] ve büyük-çıktı (maxout) gibi fonksiyonlar kullanılmaktadır. Eş. 3.28-3.33'de sigmoid, hiperbolik tanjant, ELU, ReLU, Leaky ReLU ve büyük-çıktı aktivasyon fonksiyonları sırasıyla verilmiştir.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.28}$$

$$f(x) = \tanh(x) \tag{3.29}$$

$$f(x) = \begin{cases} x & ,x \geq 0 \\ a(e^{x} - 1) & ,x < 0 \end{cases} \tag{3.30}$$

$$f(x) = [x]^{+} = \max(0, x) \tag{3.31}$$

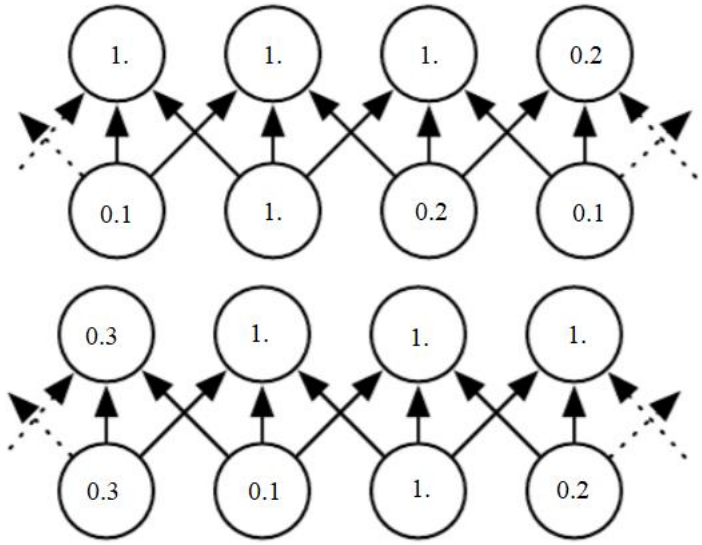
$$f(x) = 1(x < 0)(\alpha x) + 1(x \geq 0)(x) \tag{3.32}$$

$$f(x) = \max({w_1}^{T} x + b_1, {w_2}^{T} x + b_2) \tag{3.33}$$

Sigmoid fonksiyonu; sınırlanmış, türevlenebilir ve her noktada türevi negatif olmayan yapısı nedeniyle yapay sinir ağı çalışmalarında ve uygulamalarında tercih edilmiştir. Yapay sinir ağlarında, düşük giriş değerlerinden dolayı hesaplanan küçük türev değerleri nedeniyle kaybolan gradyan problemi söz konusu olmaktadır. ReLU sınırlanmış, 0 noktası dışında her noktada türevlenebilir ve türev değerinin 0 ya da 1 olması garantilendiği için kaybolan gradyan sorununa neden olmaması nedeniyle en popüler aktivasyon fonksiyonlarından biri olmuştur. NN ve CNN ağlarındaki öğrenmeyi hızlandırmak üzere önerilen ELU, kaybolan gradyan sorunuyla karşılaşılmadan yüksek sınıflandırma başarısı elde edilmesini sağlamıştır. Literatürde hangi aktivasyon fonksiyonunun daha üstün olduğuna yönelik bir fikir birliğine henüz varılamamıştır. Bilgisayarlı görü ve makine öğrenmesi uygulamalarında probleme ve giriş verisine özgü olarak farklı aktivasyon fonksiyonları tercih edilmektedir.

3.2.3. Biriktirme (pooling) fonksiyonu

Evrişimsel ağlarda bir katman genellikle üç aşamadan oluşmaktadır. Evrişim olarak bilinen ilk adımı doğrusal olmayan bir aktivasyon fonksiyonu takip etmektedir. Bu adım "tespit adımı" (detector stage) olarak da adlandırılmaktadır. Son adımda yer alan biriktirme fonksiyonu ile bir lokasyondaki piksel değeri komşu piksel değerlerinin bir ortalama ya da maksimum değeri gibi istatistiği ile değiştirilmektedir. Bu işlem, giriş görüntüsündeki küçük varyasyonlara karşı elde edilen gösterimlerin stabil olmasını garantilemek üzere tercih edilmektedir.



Resim 3.10. Biriktirme fonksiyonu ile stabillik ilişkisi [154]

Resim 3.10'da biriktirme fonksiyonu kullanılarak stabillik kavramının nasıl sağlandığı örneklenmiştir. Resimde orijinal veri ile bir piksel kaydırılmış veri biriktirme adımına girdi olarak verilmiştir. Resimde de görüldüğü gibi tüm piksel değerleri değişmiş olmasına rağmen çıktının sadece yarısının değiştiği gözlemlenmektedir.

Biriktirme operatörünün evrişimsel ağlarda yaygın olarak kullanılmasının bir nedeni ağ boyutunun indirgenebilmesidir. Böylece, eğitim süresi kısalarak daha az maliyetle eğitim gerçekleştirilebilmektedir.

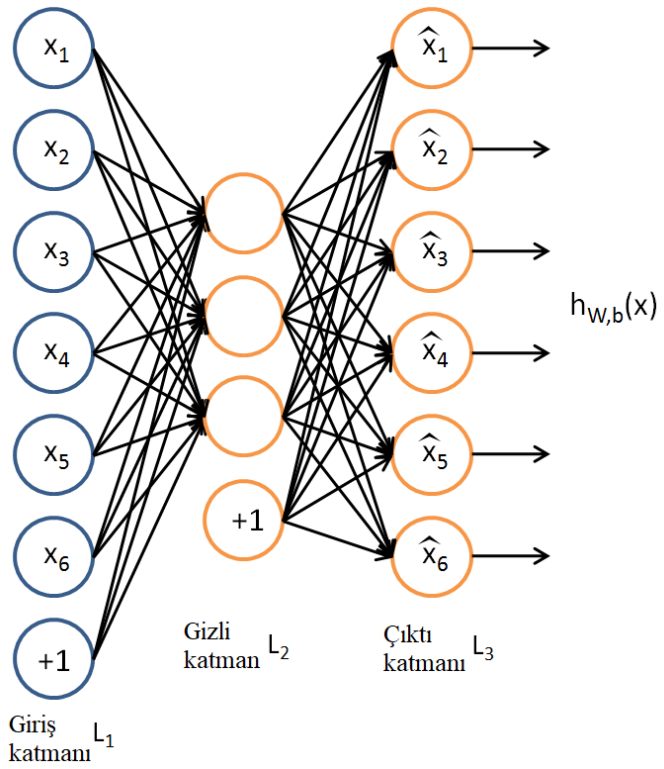
Biriktirme operatörünün diğer bir özelliği ise çok-ölçeklilik sağlayabilmesidir. Evrişim ve biriktirme işlemleri ardarda uygulandığında çok-ölçekli bir ağ elde edilmektedir.

3.3. Üretici Modeller

Üretici modellerde öğrenme, ayrımcı modellerden farklı olarak, x, y girdi çiftlerinin $p(x, y)$ bileşik olasılığına bağlıdır. Üretici modeller ile ayrımcı modellerde olduğu gibi bir sınıf etiketi üreterek verileri ayırt edebilmek üzere eğitilmemektedir. Bu modellerde, veri dağılımını öğrenmek üzere yapılan eğitim aracılığıyla yeni örneklerin bu dağılımdan elde edilmesi hedeflenmiştir.

3.3.1. AE

Otokodlayıcılar, girdiyi çıktıya kopyalamak üzere eğitilen yapay sinir ağı modeli olarak ortaya çıkmıştır. Yapısal olarak en temel hali bir giriş katmanı, bir gizli katman ve çıkış katmanından oluşmaktadır. Gizli kod olarak adlandırılan gizli katman, görüntünün/verinin bir gösterimi olarak ele alınabilmektedir. Tarihçesi eski zamanlara dayanan otokodlayıcı modelleri, boyut indirgeme ve öznitelik öğrenmek üzere kullanılmıştır. Örnek olarak Resim 3.11'de, 3 katmandan, her katmanı ise en az bir düğümden oluşan basit bir otokodlayıcı modeli verilmiştir.



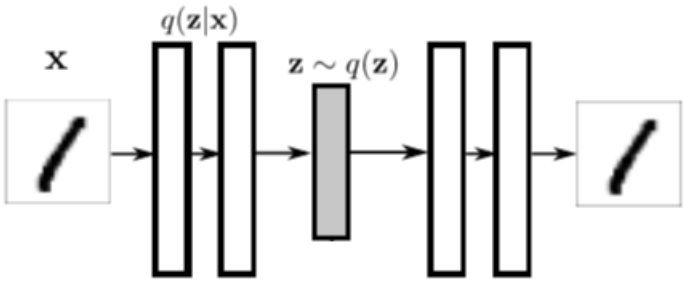
Resim 3.11. Basit otokodlayıcı modeli [157]

Yapay sinir ağlarında olduğu gibi Eş. 3.34'da verilen $a^{(2)}$ gizli katman ile $\hat{x}$, Eş. 3.35'de gösterildiği şekilde elde edilmektedir.

$$a^{(2)} = f(W^{(1)}x + b^{(1)}) \tag{3.34}$$

$$\hat{x} = f(W^{(2)}a^{(2)} + b^{(2)}) \tag{3.35}$$

Son yıllarda çekişmeli eğitim yaklaşımı ile tekrar gündeme gelen üretici modeller ile otokodlayıcı modelleri ilişkilendirilerek, otokodlayıcılar tekrar önem kazanan bir konu olmuştur. Otokodlayıcı modelleri, ileri-beslemeli ağların özel bir durumu olarak düşünülebilmektedir. İleri-beslemeli ağlarda da olduğu gibi, hesaplanan gradyan değerlerine göre ağırlıklar geri-besleme ile güncellenerek ağ eğitilmektedir.



Resim 3.12. Otokodlayıcı görüntü yeniden yapılandırma modeli [45]

AE, kodlayıcı (E) ve kodçözücü (D) olmak üzere iki ağ bileşeninden oluşan gözetimsiz öğrenmeye dayalı bir ağ modelidir. Kodlayıcı x görüntüsünü girdi olarak alarak sıkıştırıp gizli kod (latent code) olarak da ifade edilen z vektörüne dönüştürmeye çalışmaktadır. Eş. 3.36'da verilen kodlanmış z vektörü, gerçek veri dağılımından gelen x görüntülerine bağlı olarak $q(z|x)$ ile elde edilmektedir.

$$z = E(x, \theta_e) = q(z|x) \tag{3.36}$$

Kodçözücü ağ ise sıkıştırılmış görüntüyü alarak yeniden görüntü elde etmeye çalışmaktadır. Eş. 3.37'de gösterildiği gibi $p(x|z)$ ile $\hat{x}$ görüntüsünü elde etmek üzere kodçözücü ağ kullanılmaktadır.

$$\hat{x} = D(z, \theta_d) = p(x|z) \tag{3.37}$$

Otokodlayıcı modeli Eş. 3.38'te verilen kayıp fonksiyonunu minimize etmek üzere eğitilerek model dağılımı ile gerçek veri dağılımı yakınsanmaya çalışılmaktadır.

$$L_{AE} = \frac{1}{N}\sum_{i}^{N} \|x_i - \hat{x}_i\|_2^2 \tag{3.38}$$

3.3.2. VAE

VAE modeli, AE modelinde görülebilen kaybolan gradyan ve aşırı öğrenme problemleri ile karşılaşmamak üzere bir varyasyon terimi eklenerek elde edilen otokodlayıcı modelidir. Kodlayıcı (*E*) ve kodçözücü (*G*) olmak üzere 2 ağ modelinden oluşmaktadır. Modelde, x verileri girdi olarak alınarak sıkıştırılmış z_μ ortalama ve z_σ standart sapma vektörlerine dönüştürülmektedir. Bu vektörlerin bir varyasyon terimi ile bir araya getirilmesi ile z gizli kodu elde edilmektedir.

$$z_\mu, z_\sigma = E(x;\ \theta_e) \tag{3.39}$$

$$\mathrm{z} = z_\mu + \epsilon z_\sigma = q(z|x) \tag{3.40}$$

Eş. 3.40'da verildiği gibi, z gizli kod vektörü, $q(z|x)$ olarak da ifade edilen gerçek verilerinin dağılımından elde edilmektedir. Kodçözücü ağ, gizli kodu tekrar giriş verisine dönüştürebilmek üzere kullanılan bir ağdır. Eş. 3.41'de verilen yeniden yapılandırma verisi olarak ifade edilen $\hat{x}$, $p(x|z)$ model dağılımdan elde edilmektedir.

$$\hat{x} = G(z;\ \theta_d) = p(x|z) \tag{3.41}$$

Elde edilen yeniden yapılandırma verilerinin gerçek verilere olan benzerliğini ölçebilmek üzere Eş. 3.42'de verilen L_{pixel} piksel tabanlı yeniden yapılanma maliyet fonksiyonu kullanılmaktadır.

$$L_{pixel} = -\mathbb{E}_{q(z|x)}\left[\log p(x|z)\right] \tag{3.42}$$

Model dağılımı ile veri dağılımını örtüştürebilmek üzere, L_{prior} olarak Eş. 3.43 ve Eş. 3.44'te ifade edilen Kullback Leiber (KL) ıraksama maliyeti kullanılmaktadır.

$$L_{prior} = \mathbb{D}_{KL}(\, q(z|x)||\, p(z)) \tag{3.43}$$

$$\mathbb{D}_{KL}\big(\, q(z|x)||p(z)\big) = \int_{-\infty}^{\infty} q(z|x)\ \log\big(q(z)\big)\big|p(z)\big)dz \tag{3.44}$$

VAE modeli L_{pixel} ve L_{prior} maliyet fonksiyonlarının toplamı şeklinde Eş. 3.45'te ifade edilen objektif fonksiyonuna göre eğitilmektedir.

$$L_{VAE} = L_{pixel} + L_{prior} \tag{3.45}$$

3.3.3. GAN

GAN modellerinin ortaya çıkmasından önce üretici modeller, hesaplama maliyetleri ve karmaşık yapıları nedeniyle tercih edilmemiştir. GAN modeli ile birlikte üretici modeller, veri oluşturma ve veri dağılımını öğrenme gibi kabiliyetler kazanmıştır.

Üretici modeller ile

- Sentetik fakat gerçekçi veri oluşturma: görüntü sentezleme [44, 30, 47]
- Tanımlanan kelime ve cümle tabanında şartlı içerik oluşturma [32, 33]
- Çekişmeli eğitim ile modelin sınıflandırma kapasitesini değerlendirerek geliştirmek [165]
- Verilerde mevcut olan eksikliklerin tahmin edilerek tamamlanması: görüntü tamamlama, boyama [34-36]
- Orijinal görüntüleri daha önce tanımlanmış özniteliklere göre gizli kod interpolasyonu ile poz, görüntü parçaları ve nesneleri modifiye etmek [30, 31]
- Orijinal görüntülerdeki sahne yapısının değiştirilerek görüntülerin bir sahneden başka sahneye transfer edilmesi: görüntüden-görüntüye geçiş [37, 38]
- Birden fazla problemler için tek bir modelin eğitilebilmesi [166, 167]

gibi uygulamalar yapılmıştır.

Bir GAN modeli iki ağdan oluşur: üretici ağ (G) ve ayrımcı ağ (D). GAN modelinde G ve D ağları çok katmanlı perseptron ağlarıdır. Bu model, p_z önceden bilinen bir olasılıktan örneklenen z gürültü girdisini alarak $G(z;\ \theta_g)$ fonksiyonu ile bir p_g dağılımını öğrenmeye çalışmaktadır. D ağı ise bir x görüntüsünü girdi olarak alan $D(x;\ \theta_d)$ fonksiyonudur. Ayrımcı ağ x girdisinin p_{data} dağılımından olup olmadığını ayırt edebilmek üzere eğitilir. Eğer x, p_{data} dağılımından bir gerçek görüntü ise D ağının 1'e yakın bir çıktı üretmesi beklenmektedir. Aksi takdirde, yani x, p_g dağılımından ise çıktının 0'a yakın olması beklenir. D ağının hedefi, x gerçek verileri için $D(x)$'i maksimize etmek iken p_z'den örneklenen görüntüler için ise $D(x)(=\ D(G(z;\ \theta_g)\ )$ fonksiyonunu minimize etmektir. Bu zıt amaçlı durum literatürde mini-max problemi olarak adlandırılmıştır. G ağının amacı ise $D(G(z;\ \theta_g)\ )$ fonksiyonunu maksimum yapacak şekilde ayrımcı ağı kandırabilmektir. Bu durum $1 - D(G(z;\ \theta_g)\ )$'yi minimize etmeye eşdeğerdir. Bu denklik ile ayrımcı ağ maliyet fonksiyonu Eş. 3.46'da verildiği şekilde ifade edilebilmektedir.

$$\mathrm{L}_d^{GAN} = \max_{\theta_g,\theta_d}(\mathbb{E}_{x\sim p_{data}}\ [\log\ (\ D(x;\theta_d)] + \mathbb{E}_{z\sim p_z}\big[\log\ (1 - D(G\big(z;\ \theta_g\big);\ \theta_d)\big]) \quad (3.46)$$

Üretici ağ için ise hedeflenen, $D\big(G\big(z;\ \theta_g\big);\ \theta_d\big)$) değerini maksimize etmek yani Eş. 3.47'de de gösterildiği gibi $1 - D\big(G\big(z;\ \theta_g\big);\ \theta_d\big)$) değerini minimize etmektir.

$$\mathrm{L}_g^{GAN} = \min_{\theta_g,\theta_d}(\ \mathbb{E}_{z\sim p_z}\big[\log\ (1 - D(G\big(z;\ \theta_g\big);\ \theta_d)\big]) \quad (3.47)$$

Böylece, GAN maliyet fonksiyonu Eş. 3.48'de gösterildiği şekilde bir araya getirilmiştir.

$$L_{GAN} = \min_{\theta_g,\theta_d}\max(\ \mathbb{E}_{z\sim p_z(z)}\big[\log\ (1 - D\big(G\big(z;\ \theta_g\big);\ \theta_d\big)\big] + \mathbb{E}_{x\sim p_{data}}\ [\log\ (\ D(x;\theta_d)]) \quad (3.48)$$

GAN ile hedeflenen optimum durum p_g = p_{data} durumudur.

4. GÖRÜNTÜ OLUŞTURMA VE GÖRÜNTÜNÜN YENİDEN YAPILANDIRILMASI

Tez çalışması kapsamında öncelikli olarak görüntü oluşturma ve yeniden yapılandırma problemi ele alınmıştır. Bu amaçla, görüntü oluşturma ve yeniden yapılandırma problemi için, tez çalışmasının ilk bölümünde, VAE ve GAN modellerinin avantajlarını bir araya getirmek üzere bir Değişimsel Otokodlayıcı ile Çekişmeli Üretici Ağ (VAE/GAN) hibrit modeli sunulmuştur. Ayrıca, CPPN olarak bilinen modelden esinlenilerek piksel yoğunluğuna göre görüntü oluşturma yaklaşımı ile görüntülerin yüksek çözünürlüklü olarak yeniden yapılandırılabilmesi hedeflenmiştir. Model performansını iyileştirmek üzere çekişmeli eğitim yaklaşımı modele dahil edilmiştir. Bu şekilde tasarlanan model VAE/CPGAN modeli olarak adlandırılmıştır.

Gözetimsiz bir ağ modeli olan VAE, klasik otokodlayıcı modelinde olduğu gibi kodlayıcı (*E*) ve kodçözücü (*G*) ağlarından oluşan bir otokodlayıcı ağ modelidir. Klasik otokodlayıcı modelinde karşılaşılabilen kaybolan gradyan ve aşırı-öğrenme problemlerini çözmek üzere otokodlayıcıya bir varyasyon terimi eklenmesi ile elde edilmiştir. Klasik AE modelinden farklı olarak, kodlayıcı ağı x görüntülerini girdi olarak alarak Eş. 4.1'de gösterildiği gibi sıkıştırılmış z_μ ortalama ve z_σ standart sapma vektörlerine daha sonra ise bu vektörlerin doğrusal kombinasyonu ile z gizli kod vektörüne dönüştürülmektedir.

$$z_\mu, z_\sigma = E(x;\ \theta_e) \tag{4.1}$$

$$\mathrm{z} = z_\mu + \epsilon z_\sigma = q(z|x) \tag{4.2}$$

Eş. 4.2 'de verildiği gibi, z gizli kod vektörü $q(z|x)$ ile ifade edilebilen verilerinin gerçek dağılımından elde edilir. Kodçözücü, klasik otokodlayıcıda olduğu gibi, gizli kodu otokodlayıcı modeline girdi olarak alınan veriye tekrar dönüştürebilmek üzere kullanılmaktadır. Eş. 4.3'te olduğu gibi $\hat{x}$ ile gösterilen yeniden yapılandırma görüntüleri $p(x|z)$ ile ifade edilen dağılımdan elde edilir.

$$\hat{x} = G(z;\ \theta_d) = p(x|z) \tag{4.3}$$

Piksel tabanlı yeniden yapılandırma maliyet fonksiyonu Eş. 4.4'te verilen L_{pixel} fonksiyonu ile hesaplanmaktadır.

$$L_{pixel}= -\mathbb{E}_{q(z|x)}\,[\log p(x|z)] \tag{4.4}$$

Ayrıca, L_{prior} olarak Eş. 4.5'te ifade edilen Kullback Leiber (KL) ıraksama maliyeti Eş. 4.6'da gösterildiği gibi model dağılımı ile veri dağılımını örtüştürmek üzere kullanılmaktadır.

$$L_{prior} = \mathbb{D}_{KL}(\, q(z|x)||\, p(z)) \tag{4.5}$$

$$\mathbb{D}_{KL}\big(\, q(z|x)||p(z)\big) = \int_{-\infty}^{\infty} q(z|x)\, \log\big(q(z)\big)\big|p(z)\big)dz \tag{4.6}$$

Eş. 4.7'de verilen VAE objektif fonksiyonu L_{pixel} ve L_{prior} maliyet fonksiyonlarının toplamı şeklinde ifade edilebilmektedir.

$$L_{VAE} = L_{pixel} + L_{prior} \tag{4.7}$$

CPPN modeli, girdi olarak alınan verileri hedeflenen çıkış değerlerine eşlemek için kullanılan fonksiyonlar bileşimi şeklindeki bir ağ modelidir. Geleneksel yapay sinir ağlarından farklı olarak, sigmoid ve Gauss gibi aktivasyon fonksiyonlarından farklı fonksiyonların bir araya gelmesiyle oluşmaktadır. Daha önceki çalışmalarda, (x, y) koordinat sisteminden iki boyutlu ikili görüntülerin NEAT gibi bir evrişim algoritması ve evrişimsel olarak öğrenilen $f(w, x, y, r)$ fonksiyonu ile üretebildiği görülmüştür [12, 36].

Eş. 4.8'de öğrenilebilecek bir kompozisyonel ağ örneği temsili olarak sunulmuştur.

$$f(w, x, y, d)=(f_1(x, y)), f_5\big(f_2(x, y), f_4\big(f_2(x, y), f_3(x, y)\big)\big) \tag{4.8}$$

VAE modelinin çıkarım mekanizmasından yararlanabilmek üzere VAE modelinin kodlayıcı kısmı modelin ilk kısmını oluşturacak şekilde model tasarlanmıştır. VAE/CPGAN modeli kodlayıcı ağı (E) x görüntüsünü alarak VAE kodlayıcı modelinde olduğu gibi olarak

z_μ ortalama ve z_σ standart sapma gizli kod vektörleri aracılığıyla z gizli kod vektörüne dönüştürmektedir.

VAE/CPGAN modelinin ikinci kısmını CPPN benzeri kodçözücü ağ (G) oluşturmaktadır. Üretici ağ olarak da nitelendirilebilen kodçözücü ağ, z veya $z_p \in \mathcal{N}\ (0,1)$ vektörü ile koordinat sistemindeki x ekseni için x_{coord}, y ekseni için y_{coord} ve merkezden uzaklığı ifade eden yarıçap değeri olarak r'yi girdi olarak alarak z veya z_p vektörü ile füzyon işlemi için x_{vec}, y_{vec} ve r_{vec} vektör haline getirir, s ölçek faktörü için $m \times m$ boyutlarındaki x_p veya $\hat{x}$ görüntülerine dönüştürmek üzere kullanılır.

Modelin son kısmını ise ayrımcı ağ modeli oluşturmaktadır. Ayrımcı ağ (D), x gerçek görüntülerini sentetik x_p ve $\hat{x}$ görüntülerinden ayırt etmeye çalışmaktadır. Ağ ikili bir sınıflandırıcı olarak sentetik görüntüleri 0 gerçek görüntüleri ise 1 olarak sınıflandırmaya çalışmaktadır. Ayrımcı ağ, yüksek seviyeli özniteliklerin çıkarılması için de kullanılmaktadır. Ara katman l'den elde edilen öznitelikler $f_l(x)$ ile ifade edilmektedir. Bu öznitelikler tabanında karşılaştırma yapmak üzere tanımlanan eleman tabanlı maliyet fonksiyonu L_{fea} Eş. 4.9'da verilmiştir.

$$L_{fea} = -\mathbb{E}_{q(Z|X)}\ [\log p(\ f_l(x)|z)] \tag{4.9}$$

VAE modelinde olduğu gibi L_{prior} Kullback Leiber (KL) ıraksama maliyet fonksiyonu (Eş. 4.7'de daha önce verildiği gibi) objektif fonksiyonu olarak kullanılmıştır. Çekişmeli eğitim yaklaşımı için L_{CPGAN} maliyet fonksiyonu Eş 4.10'da gösterildiği şekilde x_p ile ayrımcı model eğitildiği için x_p dikkate alınacak şekilde yeniden tanımlanmıştır.

$$\begin{aligned} &L_{CPGAN} = \\ &\min_{\theta_g,\theta_d} \max \big(\mathbb{E}_{z_p\sim\mathcal{N}\ (0,1)}\big[\log\ (1 - D\big(G\big(z_p, x_{vec}, y_{vec}, r_{vec};\ \theta_g\big);\ \theta_d\big)\big] - \\ &\mathbb{E}_{z\sim p_z}\big[\log\ (1 - D\big(G\big(z, x_{vec}, y_{vec}, r_{vec};\ \theta_g\big);\ \theta_d\big)\big] + \\ &\mathbb{E}_{x\sim p_{data}}[\log\ (D(x;\theta_d)]) \end{aligned} \tag{4.10}$$

Önerilen VAE/CPGAN modelinin VAE/GAN modelinden uyarlanan eğitim algoritmasına

Resim 4.1'de yer verilmektedir.

1: $\theta_E, \theta_G, \theta_D$ ←parametrelerin ilk değer atamaları
2: k ←yığın boyutu
3: m ←örnek sayısı
4: **while** E, G, D yakınsanmamış **do**
5: $x \leftarrow x^{(i)}, i = 1, \ldots, m$ yığını
6: $z_\mu, z_\sigma \leftarrow E(x;\ \theta_E)$
7: $z \leftarrow z_\mu + \varepsilon z_\sigma, \varepsilon \in \mathcal{N}(0, I)$
8: $L_{prior} \leftarrow \mathbb{D}_{KL}(\ q(z|x)||\ p(z))$
9: $x_{vec}, y_{vec}, r_{vec} \leftarrow m \times m$ ızgara
10: $\hat{x} = G(z, x_{vec}, y_{vec}, r_{vec}; \theta_G)$
11: $z_p \leftarrow \mathcal{N}(0, I)$
12: $\hat{x}_p = G(z, x_{vec}, y_{vec}, r_{vec}; \theta_G)$
13: $L_{fea} \leftarrow -\mathbb{E}_{q(Z|X)}\ [\log p(\ f_l(x)|z)]$
14: $L_{CPGAN} \leftarrow \mathbb{E}_{z_p \sim \mathcal{N}\ (0,1)}\big[\log\ (1 - D\big(G\big(z_p, x_{vec}, y_{vec}, r_{vec};\ \theta_g\big);\ \theta_d\big)\big] +$
$\mathbb{E}_{z \sim p_z}\big[\log\ (1 - D\big(G\big(z, x_{vec}, y_{vec}, r_{vec};\ \theta_g\big);\ \theta_d\big)\big] +$
$\mathbb{E}_{x \sim p_{data}}[\log\ (D(x; \theta_d)])$
15: $\theta_E \leftarrow \theta_E - \nabla_{\theta_E}(L_{prior} + L_{fea})$
16: $\theta_G \leftarrow \theta_G - \nabla_{\theta_G}(L_{fea} - L_{CPGAN})$
17: $\theta_D \leftarrow \theta_D - \nabla_{\theta_D} L_{CPGAN}$
18: **end while**

Resim 4.1. VAE/CPGAN model eğitim algoritması

5. TEK GÖRÜNTÜDEN ÜÇ BOYUTLU NESNE YAPILANDIRMA

Tez çalışmalarının ikinci kısmını oluşturan bu bölümde, daha önceki bölümde görüntü oluşturma ve yapılandırma problemi için kullanılan üretici modeller 3 boyuta aktarılmaya çalışılmıştır. Bu bölümde hedeflenen, önceki bölümde olduğu gibi görüntülerin yeniden yapılandırılması yerine görüntülerin ait oldukları nesnelerin görüntülerden elde edilebilmeleridir. Nesnelerin tüm bakış açılarından elde edilen görüntülerinin her zaman elde edilememesi ve genellikle tek açıdan görüntülerinin mevcut olması nedeniyle, bu çalışmada da, tek açıdan görüntülerden nesne oluşturma problemine odaklanılmıştır. Ayrıca, tek bir nesne kategorisine yönelik modeller yerine tez çalışmasında çok-kategorili modellemeye odaklanılmıştır. Bu amaçla, sınıf bilgisinin kullanılmasının yararlı olacağı değerlendirilerek, sınıf bilgisi bir koşul parametresi olmak üzere koşullu otokodlayıcı tabanlı modeller 3 boyuta aktarılmaya çalışılmıştır. İkinci bölümde incelenen çalışmalarda görüldüğü üzere, son yıllarda, çekişmeli eğitime dayalı çalışmaların tercih edilmesi nedeniyle öncelikli olarak çekişmeli üretici ağlara dayalı modellere odaklanılarak VoxCAE/GAN olarak adlandırılan model üzerine çalışılmıştır. Çekişmeli eğitimin nesne yeniden yapılandırma problemi üzerindeki performansını daha iyi analiz edebilmek üzere ayrımcı ağ modelden çıkarılarak temel model olarak VoxCAE modeli ele alınmıştır. Tez çalışması kapsamında, VoxCAE modelinin farklı yaklaşımlar ile geliştirilmesi ile SkipVoxCAE ve FusedVoxCAE modelleri önerilmiştir. Nesne yeniden yapılandırma problemi için önerilen VoxCAE/GAN, VoxCAE, SkipVoxCAE ve FusedVoxCAE modelleri bu bölümde ele alınmıştır.

5.1. VoxCAE/GAN Modeli

İlk model olarak sunulan VoxCAE/GAN modeli iki boyutlu bir kodlayıcı (E), üç boyutlu bir kodçözücü/üretici (G) ve üç boyutlu bir ayrımcı (D) ağdan oluşmaktadır. AE [23] modeline benzer olarak, kodlayıcı (E) x görüntüsünü z gizli koduna Eş. 5.1'de verildiği gibi $q(z|x)$ ile dönüştürmektedir.

$$z = E(x;\ \theta_e) = q(z|x) \tag{5.1}$$

Eşitlikte x, p_{data} gerçek dağılımdan gelen, üç boyutlu nesnenin tek bir bakış açısından (single-view) elde edilen 64×64 boyutlu ikili derinlik/silüet görüntülerini ifade etmektedir.

Üç boyutlu kodçözücü/üretici ağ (G), z gizli kodu ve çok-kategorili model eğitimi için c etiket bilgisini girdi alarak $G(z,c;\ \theta_g)$ olarak ifade edilen üç boyutlu nesneleri Eş. 5.2'de verildiği şekilde oluşturmaktadır.

$$\hat{y}\ =\ G(E(x;\ \theta_e),c;\ \theta_g)\ =\ p(y|z,c) \tag{5.2}$$

Eş. 5.2 ile elde edilen $\hat{y}$, 32×32×3 iki boyutlu nesneleri ifade etmektedir.

Model ile elde edilen nesnelerin gerçek nesneler ile benzerliğini ölçmek üzere diğer çalışmalarda yeniden yapılandırma maliyet fonksiyonu olarak kullanılan Öklid uzaklığı ya da çapraz entropi (cross-entropy) yerine IoU skoruna dayanan türevlenebilir bir maliyet fonksiyonu amaçlanmıştır. Bu amaçla Eş. 5.3'te verilen denkliğe göre elde edilen IoU skoruna göre maliyet fonksiyonu Eş. 5.4' te verildiği şekilde elde edilmiştir.

$$IoU = \frac{I(X,Y)}{U(X,Y)} = \frac{\cap\,(X,Y)}{\cup\,(X,Y)} = \frac{|X*Y|}{|X+Y-X*Y|}\ \approx \frac{\sum X*Y}{\sum X+Y-X*Y} \tag{5.3}$$

$$L_{IoU} = 1 - IoU \tag{5.4}$$

Elde edilen maliyet fonksiyonuna ve Eş. 5.5 ve Eş. 5.6'da verilen eşitliklere göre geri-besleme Eş. 5.7'de gösterildiği şekilde yapılabilecektir.

$$I(X,Y) = \sum_i x_i * y_i \tag{5.5}$$

$$U(X,Y) = \sum_i x_i + y_i - x_i * y_i \tag{5.6}$$

$$\partial\frac{L_{IoU}}{\partial x_i} = -\frac{\partial L_{IoU}}{\partial x_i}\left[\frac{I(X,Y)}{U(X,Y)}\right] = \frac{-U(X,Y)\frac{\partial I(X,Y)}{\partial x_i} + I(X,Y)\frac{\partial U(X,Y)}{\partial x_i}}{U(X,Y)^2}$$
$$= \begin{cases} -\dfrac{1}{U(X,Y)}, y_i = 1 \\ \dfrac{I(X,Y)}{U(X,Y)^2}, y_i = 0 \end{cases} \tag{5.7}$$

Böylece, Eş. 5.8'de verilen IoU maliyet fonksiyonu tez çalışması kapsamında önerilmiştir.

$$L_{IoU} = 1 - \frac{\sum_{ijk} I(G(E(x_{ijk})) > t) * y_{ijk}}{\sum_{ijk} I(G(E(x_{ijk})) > t) + y_{ijk} - \sum_{ijk} I(G(E(x_{ijk})) > t) * y_{ijk}} \quad (5.8)$$

Eş. 5.8'de verilen y_{ijk} orijinal nesneyi, t eşik değerini (deney çalışmalarında t=0,4 veya t=0,5), I ise indikatör fonksiyonunu ifade etmektedir. İndikatör fonksiyonuna Eş. 5.9'da tanımlanmıştır.

Eşitlikte farklı bir y fonksiyonu için indikatör fonksiyonu $I(y): \Omega \rightarrow \{0,1\}$ olarak tanımlanmıştır.

$$I(y(i)) = \begin{cases} 1, & i \in y \\ 0, & i \notin y \end{cases} \quad (5.9)$$

Üç boyutlu ayrımcı ağ (D) ise $G(E(x;\ \theta_e), c;\ \theta_g)$ veya y nesnelerini c etiket bilgisi ile birlikte girdi olarak alan ve ağ çıktısı olarak nesneler gerçek ya da sahte şeklinde sınıflandıran ikili bir sınıflandırıcıdır. Ayrıca, ayrımcı ağ ile elde edilen yüksek-seviyeli öznitelikler $f_l\left(G\left(E(x;\ \theta_e), c;\ \theta_g\right)\right)$ veya $f_l(y)$ ile ara çıktı olarak ifade edilebilmektedir. Gerçek nesneler ve model ile sentetik verilerden elde edilen öznitelikleri L1 normuna göre karşılaştırmak üzere kullanılan öznitelik tabanlı maliyet fonksiyonu Eş. 5.10'da verilmiştir.

$$L_d^{dis} = \left| f_l(y) - f_l(G(E(x;\ \theta_e), c;\ \theta_g) \right| \quad (5.10)$$

VAE/CPGAN modelinde kullanılan ayrımcı ağ, klasik GAN modelindeki ayrımcı ağdan üç boyutlu evrişimsel bir ağ olması ve nesneler ile birlikte etiket bilgisini de girdi olarak alması yönüyle farklı bir modeldir. Çekişmeli eğitim için tanımlanan çekişmeli maliyet fonksiyonu standart çekişmeli maliyet fonksiyonu ile aynıdır. Üretici ve ayrımcı ağ için tanımlanan çekişmeli maliyet fonksiyonları Eş. 5.11 ve Eş. 5.12'de verilmiştir.

$$L_g^{GAN} = \min_{\theta_e, \theta_g}(\log(1 - D(G\left(E(x;\ \theta_e), c;\ \theta_g\right), c;\ \theta_d))) \quad (5.11)$$

$$L_d^{GAN} = \max_{\theta_d}(\mathbb{E}_{y \sim p_{data}^{3B}(y)}[\log D(y, c;\ \theta_d] + \mathbb{E}_{x \sim p_{data}(x)}(1 - D(G\left(E(x;\ \theta_e), c;\ \theta_g\right), c;\ \theta_d))) \quad (5.12)$$

Üretici ağ için maliyet fonksiyonu L_{IoU}, L_d^{dis} ve L_g^{GAN} fonksiyonlarının kombinasyonu olarak Eş. 5.13'de gösterildiği şekilde ifade edilmiştir. Ayrımcı ağ için maliyet fonksiyonu ise ayrımcı ağ çekişmeli maliyet fonksiyonuna eşittir.

$$L_g = \lambda_1 L_{IoU} + \lambda_2 L_g^{GAN} + \lambda_3 L_d^{dis} \tag{5.13}$$

$$L_d = L_d^{GAN} \tag{5.14}$$

5.2. VoxCAE Modeli

Çekişmeli eğitimin nesne yeniden yapılandırma problemine etkisini inceleyebilmek üzere önceki bölümde sunulan VoxCAE/GAN modelinin, yapısal ve objektifi daha basit hale indirgenerek elde edilen temel model olarak VoxCAE modeli elde edilmiştir. VoxCAE modeli, iki boyutlu bir kodlayıcı (E) ile üç boyutlu kodçözücüden (G) oluşmaktadır. E, x ile ifade edilen iki boyutlu silüet görüntülerini gizli kod olarak tanımlanan z'ye dönüştürmektedir. Eş. 5.15'de verildiği gibi $z, q(z|x)$ ile gösterilen x gerçek dağılımından elde edilmektedir.

$$z = E(x; \theta_e) = q(z|x) \tag{5.15}$$

Üç boyutlu kodçözücü (G) ise gerçek hayattaki verilerin çok sınıflı yapısı nedeniyle çok-kategorili bir model elde edebilmek üzere gizli koda (z) ek olarak c sınıf bilgisini de girdi olarak alan gözetimli öğrenmeye dayalı bir otokodlayıcı olarak kullanılmıştır. G fonksiyonun çıktısı olarak $\hat{y}$ üç boyutlu nesnesi, Eş. 5.16'de gösterildiği şekilde elde edilmektedir.

$$\hat{y} = G(z, c; \theta_g) = p(\hat{y}|z) \tag{5.16}$$

Önerilen model ile elde edilen $\hat{y}$ ve yer gerçekliği verisi y'nin benzerliğini ölçmek üzere Ortalama Kareler Hatası (Mean Square Error) (MSE) kullanılarak Eş. 5.17'de verilen maliyet fonksiyonunu minimize etmek üzere model eğitilmektedir. Eşitlikte N her bir epok sırasında işlenen örnek sayısını ifade etmektedir.

$$L = \frac{1}{N}\sum_{i=1}^{N} \left\| \mathrm{G}(\mathrm{E}(\mathrm{x}_i; \theta_e), \mathrm{c}; \theta_g) - \mathrm{y}_i \right\|^2 \quad (5.17)$$

5.3. SkipVoxCAE Modeli

SkipVoxCAE modeli, U-Net [65] olarak bilinen, görüntü bölütleme probleminde kullanılan modelden esinlenilerek artık bağlantıya [64] dayalı iki boyuttan üç boyuta koşullu otokodlayıcı modeli olarak sunulmuştur. SkipVoxCAE modelinde U-Net modelinden farklı olarak iki boyuttan üç boyuta öznitelik haritaları aktarılmıştır. Bu model ile amaçlanan, nesnelerin gizli kod ve koşul parametresi olan sınıf bilgilerine ek olarak daha düşük seviyeli özniteliklerden de yararlanılarak elde edilebilmesidir.

SkipVoxCAE modeli ile daha önceki modellerin bir kısıtı olan sınıf bilgisine gereksinimi en aza indirgemek üzere zayıf gözetime (weak-supervison) dayalı bir yaklaşıma odaklanılmıştır.

SkipVoxCAE, standart otokodlayıcı modellerinde olduğu gibi bir kodlayıcı (E) ve kodçözücü (D) modelinden oluşmaktadır. Kodlayıcı ağ, Eş. 5.18'de gösterildiği gibi, x görüntülerini alarak en yüksek seviyeli öznitelik haritası olan gizli kod (z_e) ile birlikte ara seviyeli öznitelik haritalarını elde etmektedir. Öznitelik haritaları, n kodlayıcı katman sayısı olmak üzere, $f = [f_2, f_3, \dots, f_n]$ olarak ifade edilmektedir.

$$z_e, f = E(x;\ \theta_e) = q((z_e, f)\ |x) \quad (5.18)$$

Üç boyutlu kodçözücü/üretici ağ ise gizli kod ($z = z_e$ ya da $z = z_p \left(\in \mathbb{N}(0,1)\right)$ ile birlikte kodlayıcının elde ettiği ara seviyeli öznitelik haritaları olarak tanımlanmış ara bağlantılar ile nesnelerin yeniden yapılandırılmasını mümkün kılmaktadır. Çok kategorili tek bir model olarak tasarladığımız SkipVoxCAE kodçözücü/üretici ağı, gizli koda eklenmiş sınıf bilgilerinden $(c = Cat(c|\pi))$ de yararlanmaktadır. Kodçözücü/üretici ile elde edilen $\hat{y}$ nesnelerine ait eşitlik ifadeleri Eş. 5.19'da verilmiştir.

$$\hat{y} = G(E(x;\ \theta_e), c;\ \theta_g) = G\left(z, c, f; \theta_g\right) = p(y|(z, f, c)) \quad (5.19)$$

Ara bağlantılarının üretici ağda nasıl kullanıldığını daha iyi anlatabilmek üzere Eş. 5.20'de üretici ağın 2. katmanına denk fonksiyonu verilmiştir. Eşitlikte, || operatörü birleştirme operatörüne karşılık gelmektedir.

$$g_2(g_1(z, c;\ \theta_g)||f_{n-1}(f_{n-1};\ \theta_{f_{n-1}});\ \theta_{g_2}) \tag{5.20}$$

Daha önce Eş. 5.8'de tanımlanan L_{IoU} skoruna dayalı olan yeniden yapılandırma hatası, model objektif fonksiyonu olarak tanımlanmıştır.

5.4. FusedVoxCAE Modeli

RGB ve silüet görüntülerinden ayrı ayrı yararlanabilmek üzere bu bölümde sunulan FusedVoxCAE modeli, iki boyutlu RGB kodlayıcı (E_{RGB}) ve derinlik kodlayıcı (E_{D}) ile volümetik bir ağ olan kodçözücü/ üretici ağdan (G) oluşmaktadır. Genel olarak ifade edilen kodlayıcı ağ (E), iki boyutlu RGBA görüntülerini, x, gizli kod veya darboğaz olarak tanımlanan Eş. 5.21'de verilen z vektörüne dönüştürür. Eşitlikte verildiği gibi, z gizli kod vektörü $q(z|x)$ çıkarım modelinden elde edilmektedir. E fonksiyonu ise E_D çıktısı ile $x_{RGB}'yi$ girdi olarak alan E_{RGB} fonksiyona denk olacak şekilde Eş. 5.22'de verildiği gibi ifade edilmektedir.

$$z = E(x;\ \theta_e) = q(z|x) \tag{5.21}$$

$$E(x;\ \theta_e) = E_{RGB}(x_{RGB}, E_D(x_D;\ \theta_{e_D})) \tag{5.22}$$

Volümetrik kodçözücü/ üretici ağ (G), Eş. 5.23'te verildiği gibi sınıf etiketlerini temsil eden bir-elemanı-bir olan vektör (one-hot vector), c, ile E kodlayıcı ağının çıktısı olarak elde edilen kodlanmış özellik vektörü ya da Eş. 5.24'de normal dağılımdan elde edilen z vektörünü giridi olarak alarak üç boyutlu $\hat{y}$ yeniden yapılandırma nesnesini elde etmektedir.

Eş. 5.25'te gösterildiği gibi, sınıf etiketleri, M kategori için etiket dağılımı ifade eden, Eş. 5.26'da verilen, π değişkenine bağlı koşullu olasılıktan gelmektedir. Sonuç olarak, üç boyutlu y nesneleri, Eş. 5.27'de verildiği gibi z vektörü ile π değişkenine bağlı bir olasılıktan elde edilmektedir.

$$\hat{y} = G(z, x;\ \theta_g) = p(y|c, z) \tag{5.23}$$

$$p(z) = \mathcal{N}(z|\ 0, I) \tag{5.24}$$

$$p(c|\pi) = Cat(c|\pi) \tag{5.25}$$

$$\pi = (\pi_c)_c, \pi_c = p(c|\pi) \tag{5.26}$$

$$p(y|z, \pi) = \sum_{c=1}^{M} \pi_c\ p(y|c, z) \tag{5.27}$$

Modeli eğitmek üzere, Eş. 5.28'de verilen yeniden yapılandırma hatası kullanılmıştır. Eşitlikte N her bir veri yığınındaki örnek sayısını, y yer gerçekliği nesnelerini ifade etmektedir.

$$L\ =\ -log\ p(y|z)\ = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i) \tag{5.28}$$

Modelin eğitim algoritması Resim 5.4'te verilmiştir.

1: $\theta_e, \theta_{e_D}, \theta_{e_{RGB}}, \theta_g$ ve π ←parametrelerin ilk değer atamaları
2: k ←yığın boyutu
3: N ← yığın örnek sayısı
4: M ←kategori sayısı
5: $c \leftarrow\ \pi[c_1, c_2, \dots, c_M]$ bir-elemanı-bir olan vektör
6: **while** E, G yakınsanmamış **do**
7: $x \leftarrow\ x_i, i = 1, \dots, N$ yığını
8: $[x_{RGB}, x_D]\ \leftarrow x$
9: $y \leftarrow\ y_i, i = 1, \dots, N$ yığını
10: $z \leftarrow E_{RGB}, \left(x_{RGB}, E_D(x_D;\ \theta_{e_D});\ \theta_{e_{RGB}}\right)$ ya da
$z \leftarrow \mathcal{N}(0, I)$
11: $\hat{y} = G(z, c;\ \theta_g)$
12: $\mathcal{L} \leftarrow \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})$
13: **end while**

Resim 5.4. FusedVoxCAE eğitim algoritması

6. DENEYSEL ÇALIŞMALAR

6.1. Görüntü Oluşturma için Veri Kümesi: MNIST

MNIST [66] [11] optik karakter tanıma ve diğer makine öğrenmesi/bilgisayarlı görü problemleri için yaygın olarak kullanılan el yazısı şeklindeki rakamlardan oluşan bir veri kümesidir. Veri kümesi 28×28 boyutlarındaki ikili görüntülerden oluşmaktadır. MNIST veri kümesinde 60,000 görüntü eğitim verisi kalan 10,000 ise test verisi olarak ayrılmıştır.

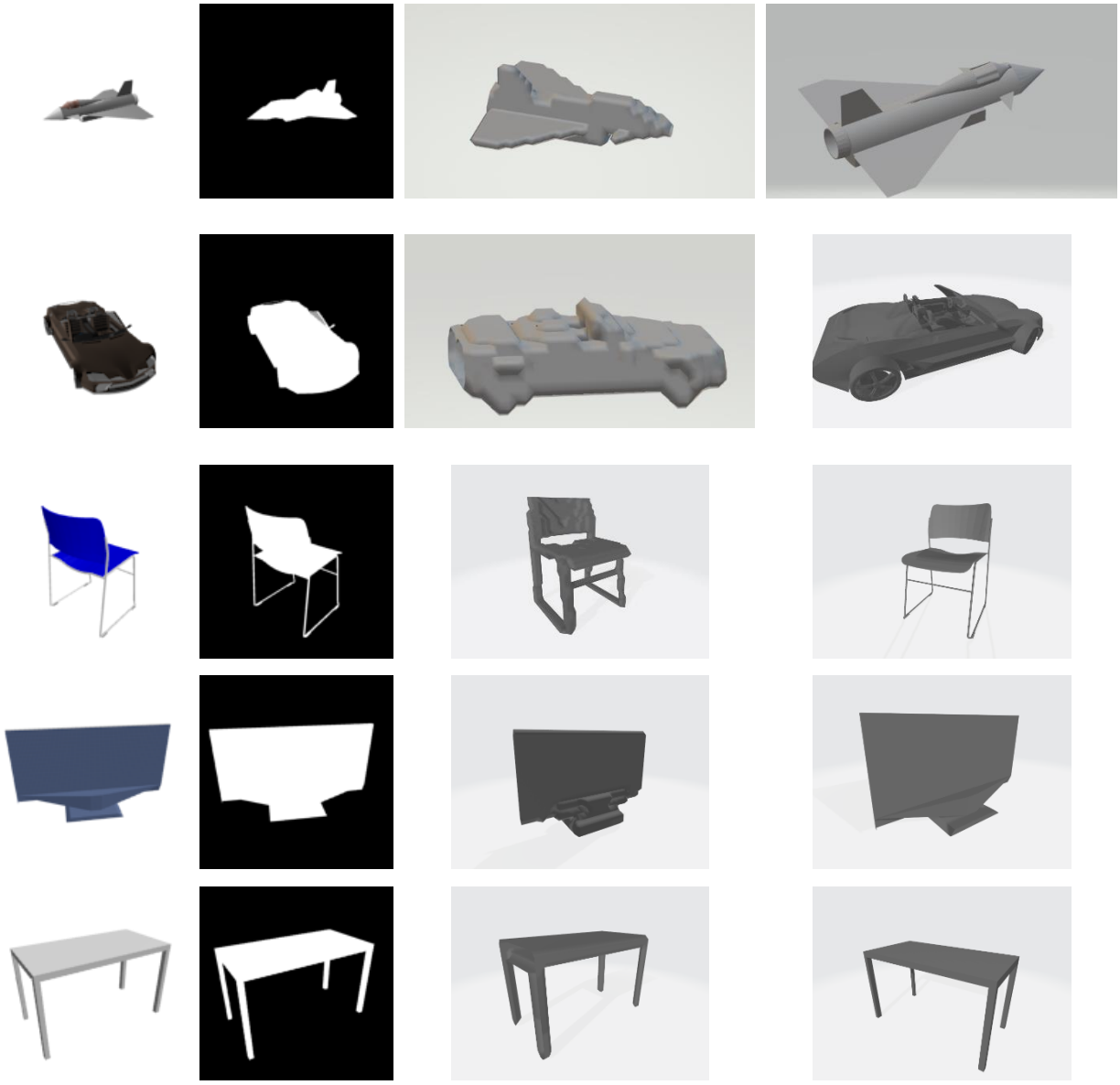
6.2. Görüntüden Nesne Oluşturma için Veri Kümesi: ShapeNetCore

Üç boyutlu nesne oluşturma problemi için tez çalışması kapsamında önerilen VoxCAE/GAN, VoxCAE, SkipVoxCAE ve FusedVoxCAE modelleri için deney çalışmaları ShapeNet verisetinin bir alt kümesi olan ShapeNetCore [63] veriseti üzerinde yürütülmüştür.

ShapeNetCore veriseti 12 nesneden oluşan toplam 34,000 CAD modelinden oluşan sentetik bir veri kümesidir. Veri kümesi CAD modellerine ek olarak, ilgili CAD modellerinin 24 farklı bakış açısından toplanan RGB görüntülerini içermektedir. Veri kümesinde 128×128×128 boyutlarındaki nesnelerin düzenlileştirilerek (normalleştirme) 32×32×32 boyutlarına dönüştürülmüş binvox [168, 169] versiyonlarından elde edilen nesneler ikili vokseller olarak kullanılmıştır. CAD modellerinden elde edilen görüntüler ise 64×64×4 boyutludur.

Çizelge 6.1. ShapeNetCore [63] alt veri kümesinden oluşturulan veri kümeleri

		Kategoriler					
Veriseti		Airplane	Car	Chair	Display	Table	Toplam
70:30 rastsal alt veri kümesi	Eğitim	2864	5263	4722	775	5922	19546
	Test	1181	2233	2056	320	2587	8377
Alt veri kümesi [58]	Eğitim	2832	2502	4612	762	5876	16584
	Geçerleme	405	353	662	112	842	2374
	Test	808	709	1317	219	1679	4732



(a) (b) (c) (d)

Resim 6.1. ShapeNetCore örnek verileri (a) RGB (b) silüet (c) 32x32x32 nesne (d) 128x128x128 nesne

ShapeNetCore veri kümesi içerdiği 12 nesne sınıfı ile kullanılabildiği gibi, literatürde yer alan bazı modellerde ShapeNetCore veri kümesinin bir alt kümesi oluşturularak daha az nesne sınıfı ile çalışılmıştır. 2B ve 3B veriler üzerinde çalışıldığı için 2B modellere göre daha uzun süren eğitim süreleri nedeniyle tez çalışmasında da veri kümesinin en çok örnek içeren "Airplane, Car, Chair, Display, Table" sınıflarına ait örneklerinden oluşturulan alt kümeleri tercih edilmiştir. Yapılan ilk deney grubunda Çizelge 6.1'de örnek sayı dağılımı verilen 70:30 rastsal alt veri kümesi kullanılmıştır. Literatürde yer alan diğer modellerle kıyaslayabilmek üzere ise [58] numaralı çalışmada kullanılan veri kümesinin sadece "Airplane, Car, Chair,

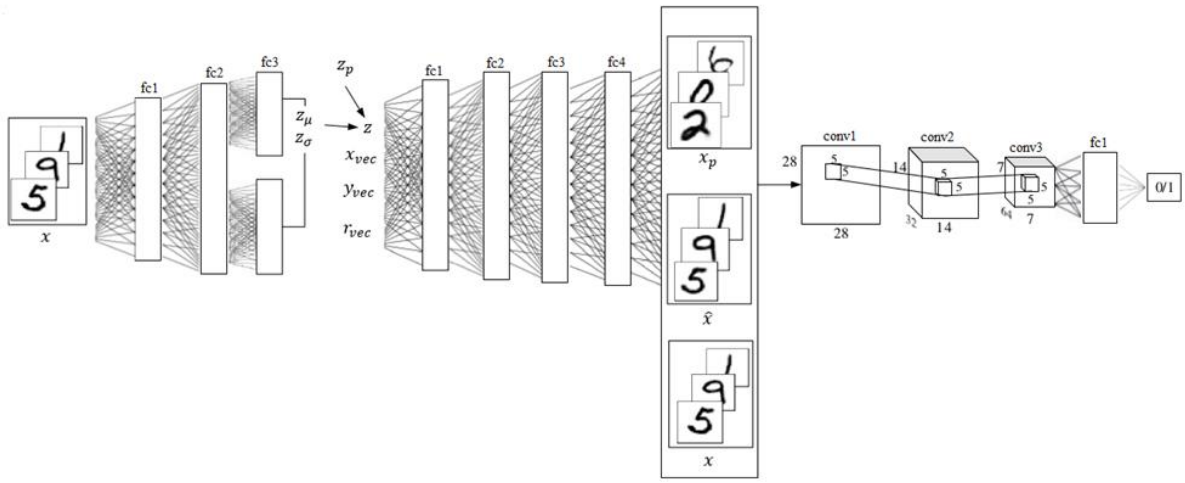
Display, Table" kategorilerinden oluşan alt kümesi çalışmada kullanılan eğitim, geçerleme ve test örnekleri ile birlikte kullanılmıştır. ShapeNetCore veri kümesi örnek görüntüleri ve nesneleri Resim 6.1'de verilmiştir.

Veri kümesinde mevcut olan görüntüler RBGA olarak okunarak, bu görüntülerden RGB ve silüet olarak da ifade edilen derinlik görüntüleri doğrudan elde edilebilmesi mümkün olmuştur.

6.3. Model Mimarileri, Parametreleri ve Uygulama Detayları

6.3.1. VAE/CPGAN mimarisi

Tezin ilk bölümünde sunulan VAE/CPGAN model mimarisi Şekil 6.1'de verilmiştir. Şekilde de görüldüğü gibi 3 ağ modelinin bir araya gelmesinden oluşan modelin kodlayıcı ile üretici ağ kısmı sırasıyla 3 ve 5 katmandan oluşan çok katmanlı perseptron ağı olarak tanımlanmıştır.



Şekil 6.1. Önerilen VAE/CPGAN ağ modeli

Kodlayıcı ağ modelinde son katman dışında yığın normalleştirme yapılarak ReLU aktivasyon fonksiyonu kullanılmıştır. Üretici ağ modelinde benzer olarak son katman dışında yığın normalleştirme ve ReLU aktivasyon fonksiyonu, son katmanda ise yığın normalleştirme yapılmaksızın sigmoid aktivasyon fonksiyonundan yararlanılmıştır. Ayrımcı ağ modeli ise 3 evrişim ve 1 tam-bağlı katmandan oluşan bir evrişimsel sinir ağıdır.

İlk katman ve son katman dışındaki evrişim katmanı ve tam-bağlı katmandan sonra yığın normalleştirme işlemi yapılmıştır. Aktivasyon fonksiyonu olarak ilk 4 katmanda Leaky ReLU [164] son katmanda ise sigmoid fonksiyonu tercih edilmiştir. VAE/CPGAN model parametrelerine Çizelge 6.2 – 6.4'te detaylı olarak yer verilmiştir.

Çizelge 6.2. VAE/CPGAN kodlayıcı ağ parametreleri

Kodlayıcı Ağ (E)	Ağ katmanları		
	fc1	fc2	fc3
Filtre sayısı	512	512	100
Veri normalleştirme	Yığın normalleştirme	Yığın normalleştirme	-
Aktivasyon	ReLU	ReLU	-

Çizelge 6.3. VAE/CPGAN üretici ağ parametreleri

Üretici Ağ (D)	Ağ katmanları				
	fc1	fc2	fc3	fc4	fc5
Filtre sayısı	128	128	128	128	128
Veri normalleştirme	Yığın normalleştirme	Yığın normalleştirme	Yığın normalleştirme	Yığın normalleştirme	-
Aktivasyon	ReLU	ReLU	ReLU	ReLU	Sigmoid

Çizelge 6.4. VAE/CPGAN ayrımcı ağ parametreleri

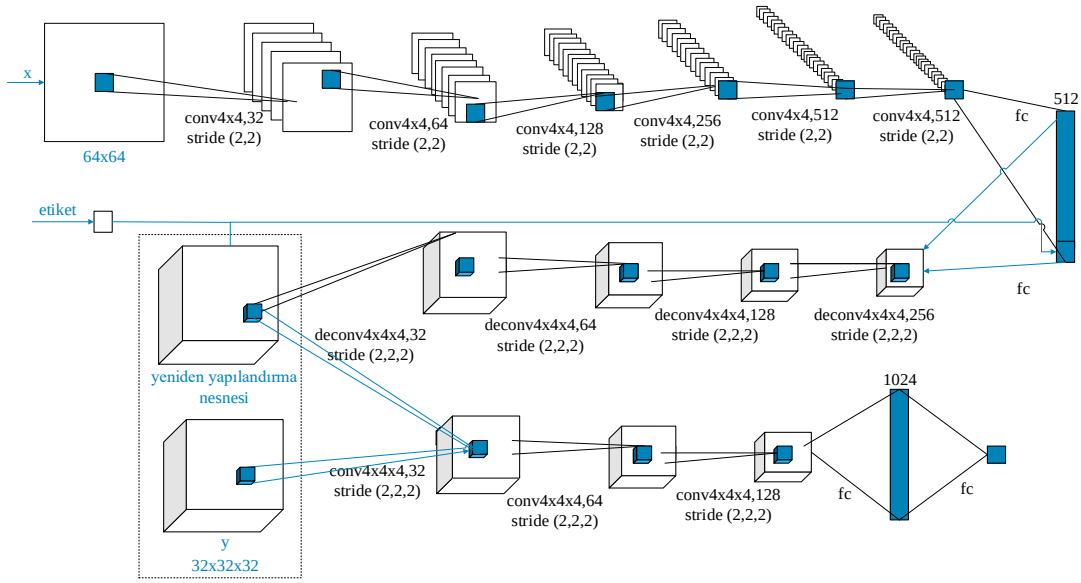
Ayrımcı Ağ (D)	Ağ katmanları				
	conv1	conv2	conv3	fc1	fc2
Filtre sayısı	32	64	128	128	1
Filtre boyutu	5×5	5×5	5×5	-	-
Adım boyutu (Stride)	2	2	2	-	-
Doldurma (Padding)	0	0	0	-	-
Veri normalleştirme	-	Yığın normalleştirme	Yığın normalleştirme	Yığın normalleştirme	-
Aktivasyon fonksiyonu	Leaky ReLU	Leaky ReLU	Leaky ReLU	Leaky ReLU	Sigmoid

Deney çalışmalarında yığın boyutu 100, gizli kod vektör boyutu da 100 olarak belirlenmiştir. Modeli eğitmek üzere Adam eniyileme [170] yaklaşımı, $\beta_1 = 0{,}65$, $\beta_2 = 0{,}999$, $\epsilon = 1e-08$, öğrenme oranı (α) = $1e-03$ parametreleri ile kullanılmıştır. Deney

çalışmalarında, kısa bir eğitim süresindeki model performansları karşılaştırabilmek üzere karşılaştırılan modeller ile VAE/CPGAN modeli 10 epok boyunca eğitilmiştir.

6.3.2. VoxCAE/GAN mimarisi

Tez çalışması kapsamında iki boyutlu silüet görüntülerinde üç boyutlu nesne oluşturmak üzere ilk olarak önerilen VoxCAE/GAN model mimarisine Şekil 6.2'de yer verilmiştir. Şekilde gösterildiği gibi kodlayıcı ağ (E) 7 katmanlı bir ağdır. Son katmanı dışındaki tüm katmanlar iki boyutlu evrişim katmanıdır. Son katmanda ise tam-bağlı katman mevcuttur. Her bir evrişim katmanında 4×4 boyutlarındaki filtreler kullanılmıştır. Evrişim katmanlarını [30] çalışmasında olduğu gibi yığın normalleştirme ve Leaky ReLU $(\alpha = 0{,}2)$ aktivasyon fonksiyonu takip etmektedir. Gizli kod boyutu 512 olmak üzere evrişim katmanlarında kullanılan filtre sayıları ise 32, 64, 128, 256, 512 ve 512 olarak belirlenmiştir.



Şekil 6.2. VoxCAE/GAN model mimarisi

Kodçözücü/üretici ağ (G) 5 katmandan oluşan koşullu bir evrişimsel ağıdır. Gizli koda eklenen etiket bilgisi, tam-bağlı bir katmandan ve bu katmanı takip eden 4 ters-evrişim katmanından sonra üç boyutlu bir nesneye dönüştürülmektedir. Her bir ters-evrişim katmanı 4×4×4 boyutlu filtrelerden oluşmaktadır. Ters-evrişim katmanlarında kullanılan filtre sayıları ise 128, 64, 32 ve 1 dir. Tam-bağlı katman ile bu katmanı takip eden ilk 3 ters-evrişim katmanını [50] numaralı çalışmada önerildiği şekilde yığın normalleştirme ve ReLU

aktivasyon fonksiyonları takip etmektedir. Son evrişim katmanında ise aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılmıştır. Ayrımcı ağ (D), 3 evrişim ve 2 tam-bağlı katmandan oluşan 5 katmanlı bir evrişimsel ağdır. Evrişim katmanları 4×4×4 boyutlu 32, 64 ve 128 adet filtrelerden oluşmaktadır. Ayrımcı ağ modelinde, ilk evrişim katmanı ve son tam-bağlı katman dışındaki katmanlardan sonra yığın normalleştirme işlemi gerçekleştirilmiştir. Aktivasyon fonksiyonu olarak çekişmeli eğitime dayalı modelde [30] olduğu gibi Leaky ReLU ($\alpha = 0{,}2$) tercih edilmiştir.

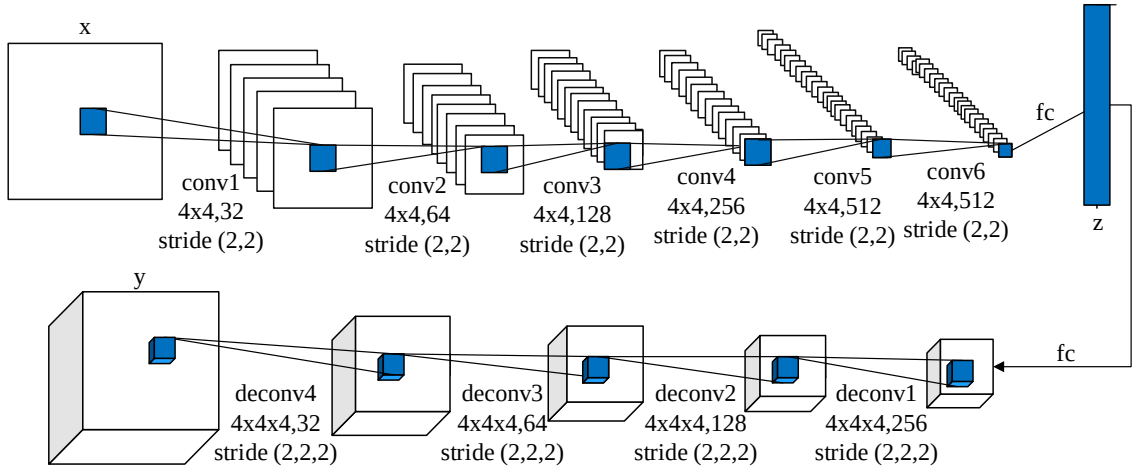
Modeli eniyilemek üzere Adam eniyileme algoritması $\beta_1 = 0{,}95$, $\beta_2 = 0{,}9$ ile 2e-4 ve 1e-5 öğrenme oranı parametreleri ile kullanılmıştır. Tüm deney çalışmaları, yığın sayısı 16 olmak üzere, tek bir NVIDIA TitanX GPU kartı üzerinde gerçekleştirilmiştir. Ağ değişkenleri için [79] numaralı çalışmada önerildiği gibi $\sigma = 0{,}02$ için gauss dağılımına göre ilk değer atamaları yapılmıştır. Objektif fonksiyonları için tanımlanan λ ağırlıkları 5, 95 ve 1 olarak belirlenmiştir. Deney çalışmalarında analiz edilen tüm modeller 50 epok boyunca yani yaklaşık olarak 60,000 iterasyon süresince eğitilmiştir.

6.3.3. VoxAE mimarisi

VoxCAE/GAN model performansını iyileştirmek üzere çekişmeli eğitimin ve koşullu ağ yaklaşımlarının model performanslarına etkisini inceleyebilmek üzere iki boyutlu görüntülerden üç boyutlu nesne oluşturma problemi için VoxAE ve VoxCAE modelleri tasarlanmıştır. Modellerin VoxCAE/GAN modelinden farkı, ayrımcı bir ağ ile çekişmeli eğitime dayalı eğitilmemeleridir. Temel model olarak, VoxAE modeline ait genel mimari gösterimine Şekil 6.3'de yer verilmiştir. Bu modelde, VoxCAE/GAN modelinde olduğu gibi kodlayıcı ağ (E) son katmanı tam-bağlı katman olmak üzere diğer katmanları iki boyutlu evrişim katmanından oluşan 7 katmanlı bir evrişimsel ağdır. 4×4 boyutlarında filtreler kullanılan evrişim katmanları yığın normalleştirme ve Leaky ReLU ($\alpha = 0{,}2$) aktivasyon fonksiyonu takip etmektedir. Evrişim katmanlarında kullanılan filtre sayıları 32, 64, 128, 256, 512 ve 512; gizli kod boyutu ise 512 olarak belirlenmiştir.

VoxCAE modelinde nesneler, sıkıştırılmış gizli koda eklenmiş bir koşul vektöründen elde edilmeye çalışılmaktadır. Bu koşul vektörü, kodçözücü modeline eklenen etiket bilgisini ifade eden vektördür. Genel olarak ifade edilmek istenirse, kodçözücü/üretici ağ (G), gizli kodu nesneye dönüştürmek üzere eğitilen 5 katmandan oluşan koşullu ya da standart bir üretici

ağıdır. Kodçözücü/üretici ağı, 1 tam-bağlı katmandan ve 4×4×4 boyutlu filtrelerden oluşan 4 ters-evrişim katmanından oluşmaktadır. Ters-evrişim katmanlarında kullanılan filtre sayıları sırasıyla 128, 64, 32 ve 1 olarak belirlenmiştir. Tam-bağlı katman ile son katman dışındaki ters-evrişim katmanlarını yığın normalleştirme ve ReLU aktivasyon fonksiyonları takip etmektedir. Son katmanda ise önceki modellerde olduğu gibi sigmoid fonksiyonu aktivasyon fonksiyonu olarak kullanılmıştır.



Şekil 6.3. VoxAE olarak adlandırılan modellere ait genel mimari gösterimi

Deneyler, bir adet NVIDIA TitanX GPU kartı üzerinde gerçekleştirilmiştir. Model, Adam eniyileme algoritması ($\beta_1 = 0{,}95$, $\beta_2 = 0{,}9$, 2e-4 ve 1e-5 öğrenme oranı parametreleri) ile eğitilmiştir. VoxAE değişkenleri için ilk değer atamaları VoxCAE/GAN modelinde olduğu gibi [79] numaralı çalışmada önerildiği şekilde yapılmıştır. Deney çalışmalarında, modeller yığın boyutu 16 olmak üzere 50 epok süresince yaklaşık 60,000 iterasyon boyunca eğitilmiştir.

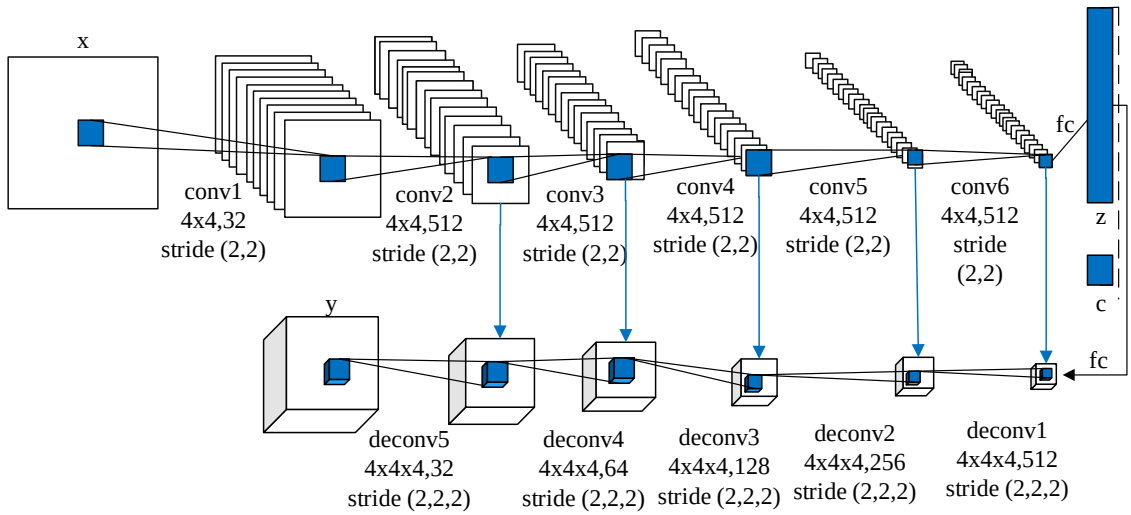
6.3.4. SkipVoxCAE mimarisi

Artık, atlanan bağlantılara dayalı olarak tasarlanan model olan SkipVoxCAE modeli, VoxAE modellerine gibi ayrımcı ağ yapısı ile çekişmeli eğitime dayanmamaktatdır. VoxAE modellerinden farklı olarak ise kodçözücü ve kodlayıcı arasında ara bağlantılar mevcuttur. Atlanan bağlantıya dayalı modeller, literatürde görüntüden-görüntüye geçiş [38], nesne öbekleme [65, 171, 172] süper çözünürlük [173, 174] ve görüntü yenileme [175] gibi problemlerde yaygın olarak kullanılmıştır. Bu modelde kullanılan ara bağlantılar ile sadece yüksek-seviyeli gösterimler yerine daha düşük-seviyeli gösterimlerden de yararlanılması

hedeflenmiştir. SkipVoxCAE olarak adlandırılan model mimarisine Şekil 6.4'te yer verilmiştir.

VoxCAE/GAN modelinde olduğu gibi, kodlayıcı ağ (E) son katmanı tam-bağlı katman diğer katmanları ise iki boyutlu evrişim katmanından oluşan 7 katmanlı bir ağdır. 4×4 boyutlarındaki filtrelerin tanımlandığı evrişim katmanları, yığın normalleştirme ve Leaky ReLU ($\alpha = 0{,}2$) aktivasyon fonksiyonu takip etmektedir. Evrişim katmanlarında kullanılan filtre sayıları 32, 64, 128, 256, 512 ve 512; gizli kod boyutu ise 512 olarak belirlenmiştir.

Gizli kod ve koşul vektörü olarak gizli kod vektörüne eklenen etiket bilgisini girdi olarak alan kodçözücü/üretici ağ (G) 5 katmandan oluşan koşullu bir evrişimsel ağdır. 1 tam-bağlı katman ve 4×4×4 boyutlu filtrelerden oluşan 4 ters-evrişim katmanından oluşmaktadır. Ters-evrişim katmanlarında kullanılan filtre sayıları sırasıyla 128, 64, 32 ve 1 olarak belirlenmiştir. Tam-bağlı katman ile son katman dışındaki ters-evrişim katmanları yığın normalleştirme ve ReLU aktivasyon fonksiyonları takip etmektedir. Son katmanda ise sigmoid fonksiyonu aktivasyon fonksiyonu olarak kullanılmıştır.



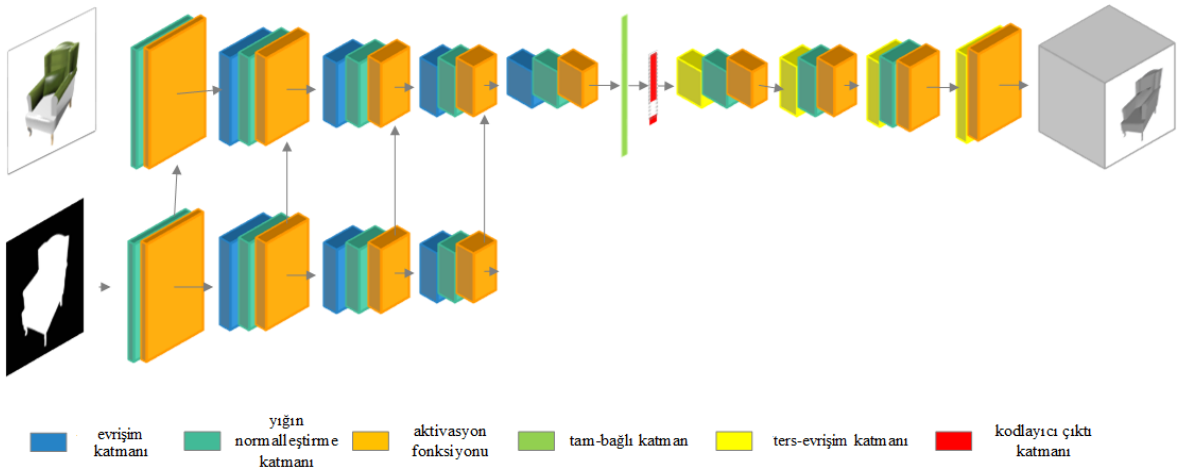
Şekil 6.4. SkipVoxCAE model mimarisi

SkipVoxCAE modelini değerlendirmek üzere yapılan deneyler daha önceki modellerde de olduğu gibi, bir adet NVIDIA TitanX GPU kartı üzerinde Adam eniyileme algoritması ($\beta_1 =$ 0,95, $\beta_2 =$0,9, 2e-4 ve 1e-5 öğrenme oranı parametreleri) ile gerçekleştirilmiştir. Model parametreleri ilk değer atamaları VoxCAE/GAN ve VoxAE modellerinde olduğu gibi [79]

çalışmasında önerildiği şekilde yapılmıştır. Deney çalışmalarında modeller, yığın boyutu 16 olmak üzere, geçerleme kümesi yakınsandığı sürece eğitilmiştir.

6.3.5. FusedVoxCAE mimarisi

Veri kümesinde yer alan görüntülerin RGBA olması dolayısıyla renk bilgisi ile derinlik bilgisinden birlikte yararlanabilmek üzere, mevcut çalışmalardan farklı olarak, bu bölümde sunulan FusedVoxCAE modeli iki kodlayıcı ağ (E) ile bir kodçözücü ağdan (D) oluşmaktadır. FusedVoxCAE model mimarisi Şekil 6.5'te verilmiştir. FusedVoxCAE ile RGB ve silüet görüntülerinden farklı özniteliklerin elde edilmesi hedeflenmiştir. Silüet kodlayıcı ağdan elde edilen öznitelik haritaları ile RGB kodlayıcının aynı seviyeli öznitelik haritalarının füzyon işlemi ile gizli kod vektörü elde edilmektedir. Kodlayıcı ağlar, filtre boyutu $k = 4 \times 4$, adım boyutu $s = 2$ olmak üzere 5 evrişim katmanından oluşmaktadır. Evrişim katmanlarındaki filtre sayıları ise sırasıyla 64, 128, 256, 512 ve 512'dir. RGB kodlayıcı, gizli kod oluşturabilmek üzere ayrıca 1 adet tam bağlı katmandan oluşmaktadır. Kodlayıcı ağlarda, her evrişim katmanından sonra yığın düzenlileştirme ve Leaky ReLU ($\alpha = 0{,}2$) aktivasyon fonksiyonu tanımlanmıştır. Özellik haritalarının füzyon işlemi, ilgili özellik haritalarının eleman eleman toplamı olarak gerçekleştirilmektedir. Volümetrik kodçözücü ağ ise filtre boyutu $k = 4 \times 4 \times 4$, adım boyutu $s = 2$ olan toplam 4 ters-evrişim katmanından oluşmaktadır. Evrişim katmanları filtre sayıları 128, 64, 32 ve 1 olarak belirlenmiştir. İlk 3 ters-evrişim katmanını yığın normalleştirme ve ReLU aktivasyonu takip etmektedir. Son ters-evrişim katmanında ise, evrişim işlemini sigmoid aktivasyon fonksiyonu takip etmiştir.



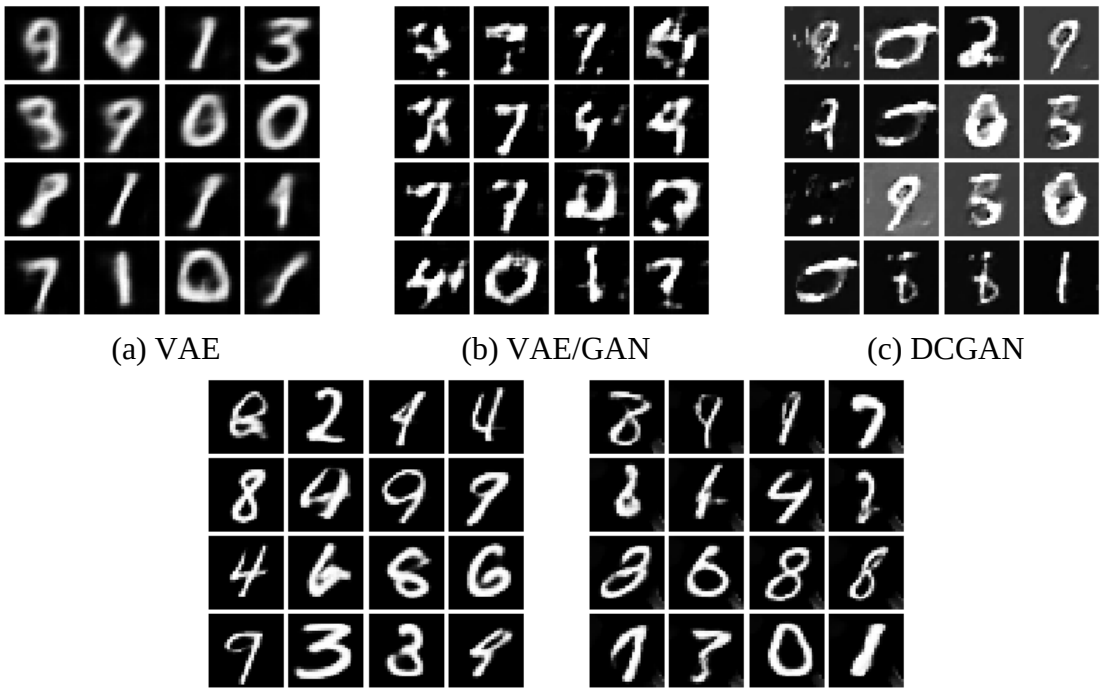
Şekil 6.5. FusedVoxCAE model mimarisi

FusedVoxCAE model eğitimi için Adam eniyileme (β1 = 0,95, β2 = 0,9 ve 2e - 4, 1e - 5 öğrenme oranları parametreleri) ile uçtan uca eğitim gerçekleştirilmiştir. Deneyler NVIDIA TitanX GPU'da Tensorflow kütüphanesi ile gerçekleştirilmiştir. Daha önceki modellerde olduğu gibi ağ ağırlıkları [79]'e göre atanarak yığın boyutu 16 olmak üzere geçerleme kümesine göre model yakınsadığı sürece eğitilmiştir.

6.4. Deney Sonuçları

6.4.1. VAE/CPGAN model karşılaştırmaları

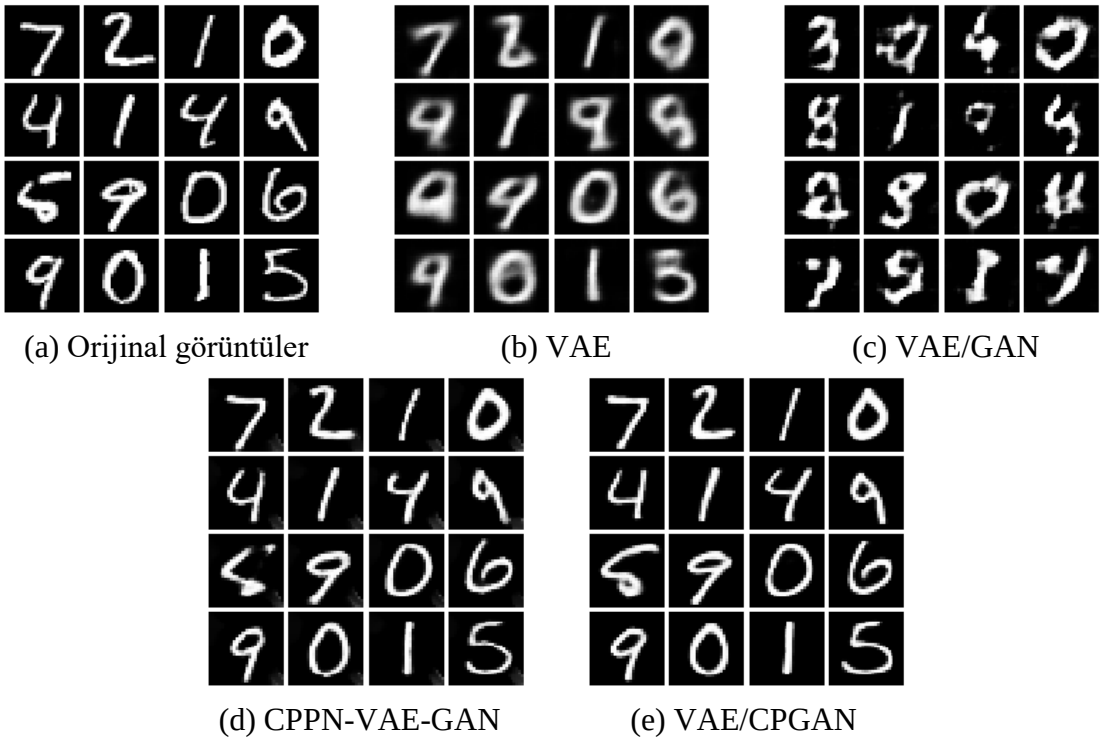
VAE/CPGAN model performansını değerlendirmek üzere modele temel olan VAE, VAE/GAN, DCGAN, CPPN-GAN-VAE modelleri ile karşılaştırmalar yapılmıştır. Bu modeller için aynı derinlikteki ağlar aynı parametreler ile oluşturulmuştur. Modeller, hem niceliksel hem de niteliksel olarak karşılaştırılmıştır. Karşılaştırmalarda kullanılan modeller ile elde edilen örnek görüntüler Resim 6.2'de verilmiştir.



(a) VAE (b) VAE/GAN (c) DCGAN

(d) CPPN-VAE-GAN (e) VAE/CPGAN

Resim 6.2. VAE, VAE/GAN, DCGAN, CPPN-VAE-GAN ve VAE/CPGAN modelleri ile elde edilen sentetik örnekler



(a) Orijinal görüntüler (b) VAE (c) VAE/GAN

(d) CPPN-VAE-GAN (e) VAE/CPGAN

Resim 6.3. Orijinal görüntüler ve VAE, VAE/GAN, CPPN-VAE-GAN ve VAE/CPGAN modelleri ile yeniden yapılandırma görüntüleri

Ayrıca, yeniden yapılandırma ile elde edilen görüntüleri karşılaştırmak üzere AE tabanlı modeller kullanılmıştır. Şekil 6.3'de verilen görüntülere göre, VAE/CPGAN modeli farklı stillerle daha gerçekçi ve bulanık olmayan görüntüler üretebilmiştir. Modelin yeniden yapılandırma performansı da diğer modeller ile elde edilen görüntülere kıyasla gerçek görüntülere daha yakın olmuştur. Tez çalışması kapsamında ele alınan temel Kompozisyonel Örüntü Üretici Değişimsel Otokodlayıcı Ağ (CPPN-VAE/GAN) modelinde yeni örnekler üretebiliyor olsa da yeniden yapılandırma görüntülerinde, önerilen modelimize göre, gürültüler olduğu dikkat çekmektedir. Deney sonuçlarına bakıldığında VAE modeli ile elde edilen görüntüler diğer modeller ile elde edilen görüntülerden daha bulanık olmuştur.

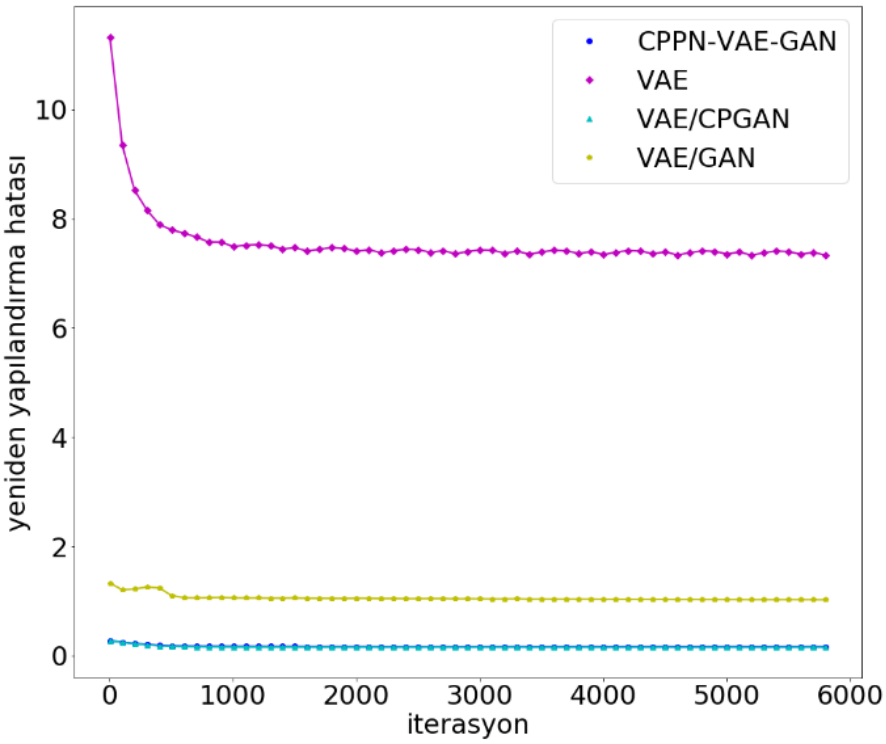
Niceliksel olarak modelleri karşılaştırmak üzere literatürde tercih edilen Inception skoru (IS) [113] kullanılmıştır. Ayrıca, otokodlayıcı modellerinde yeniden yapılandırma performansları MSE tabanında değerlendirilmiştir. Bu skorlara göre yapılan karşılaştırmalar Çizelge 6.5'te verilmiştir. Çizelgede * ile belirtilen modeller otokodlayıcı tabanlı modeller olmadığı için yeniden yapılandırma hatası hesaplanamamaktadır. Verilen sonuçlara göre yeniden yapılandırma hatası en düşük önerilen VAE/CPGAN modeli ile elde

edilmiştir. Elde edilen sentetik görüntüler ise IS skoruna göre değerlendirildiğinde en yüksek skorun 1.8538±0,0337 olarak VAE/CPGAN modeli ile elde edildiği görülmektedir. CPPN-VAE/GAN modeli ise performans olarak IS skoru ve MSE metriğine göre modelimizi takip etmektedir. Bu durum, CPPN modelinin bir üretici ağ olarak performansını ortaya koymaktadır.

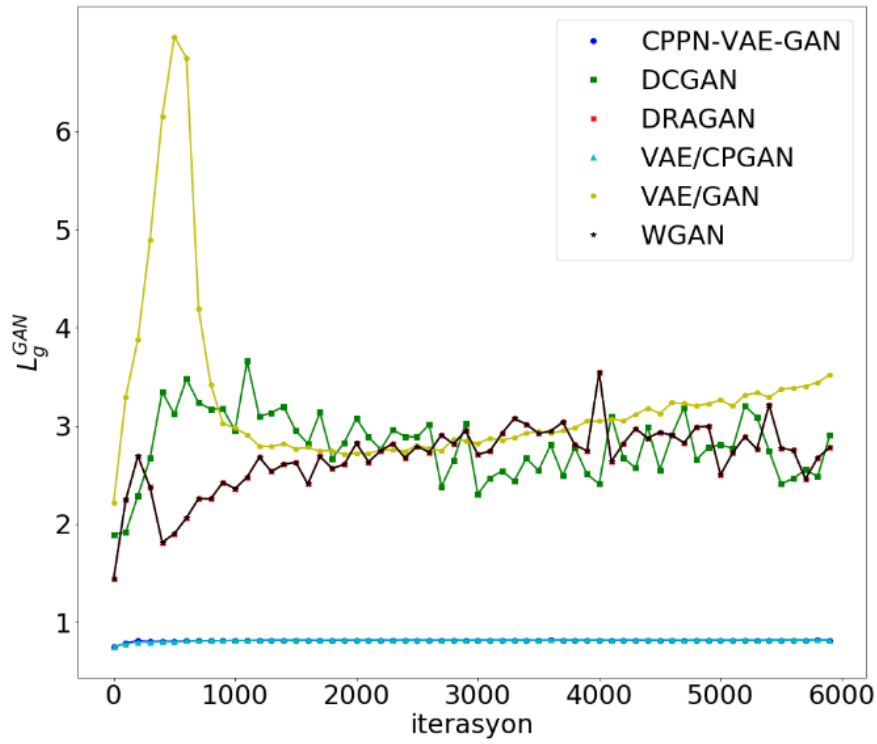
Çizelge 6.5. MSE metriği ve IS skoru tabanında model karşılaştırmaları

Model	MSE	IS	
		Ortalama	Standart sapma
VAE	3,6940	1,5830	0,0153
VAE/GAN	11,1919	1,3587	0,0335
DCGAN	*	1,6141	0,0249
WGAN	*	1,0060	0,0004
DRAGAN	*	1,0500	0,0016
CPPNVAEGAN	2,0118	1,6266	0,0518
VAECPGAN	1,7512	1,8538	0,0337

Şekil 6.6’da 6000 iterasyon boyunca elde edilen yeniden yapılandırma ve üretici ağ hata eğri grafikleri verilmiştir. Şekil 6.6 (a)’da verilen eğrilere göre VAE ve VAE/GAN modellerinin CPPN tabanlı modeller de olduğu gibi yakınsanamadığı görülmektedir. Şekil 6.6 (b)’de ise literatürde dikkat çekilen GAN tabanlı modellerde sık karşılaşılan bir problem olan eğitimde stabil olamama durumunu analiz edebilmek üzere GAN tabanlı modellerin üretici ağ hata grafikleri ele alınmıştır. GAN modellerinin eğitiminin stabil olamama problemine çözüm getirmek üzere literatürde daha önce önerilen DRAGAN [176] ve WGAN modelleri ile de karşılaştırmalara yer verilmiştir. Şekil 6.6 (b)’ye göre DCGAN, WGAN ve DRAGAN modellerinin eğitim süresince stabil performans sergileyemediği ve VAE/GAN modelinin yakınsanamadığı dikkat çekmektedir. Bu nedenle, VAE/CPGAN modelinin, diğer modellere göre daha stabil bir model olmakla birlikte IS skoruna göre test verisi üzerindeki sonuçlara göre de daha performanslı olduğu görülmektedir. Sonuçlara göre, çekişmeli eğitime dayalı modellerde görüldüğü üzere, çekişmeli eğitim ile daha gürbüz ve performanslı modeller elde edilirken diğer GAN modellerinin eğitiminin önerdiğimiz VAE/CPGAN modeli gibi stabil performans sergileyemediği değerlendirilmiştir.



(a)



(b)

Şekil 6.6. İterasyon sayısına göre (a) VAE tabanlı modellerin yeniden yapılandırma hata eğrileri (b) GAN tabanlı modellerin üretici ağ hata eğrileri

(a) Orijinal LR görüntüleri

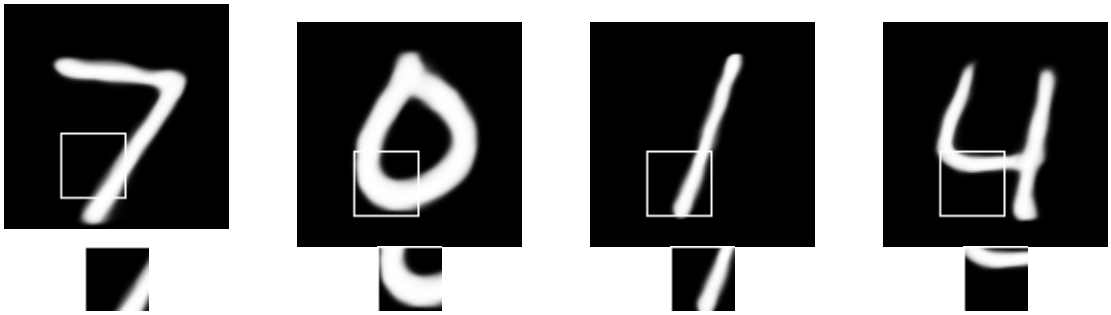
(b) En yakın komşu interpolasyon SR görüntüleri (s=2)

(c) Model SR görüntüleri (s=2)

(d) En yakın komşu interpolasyon SR görüntüleri (s=3)

(e) Model SR görüntüleri (s=3)

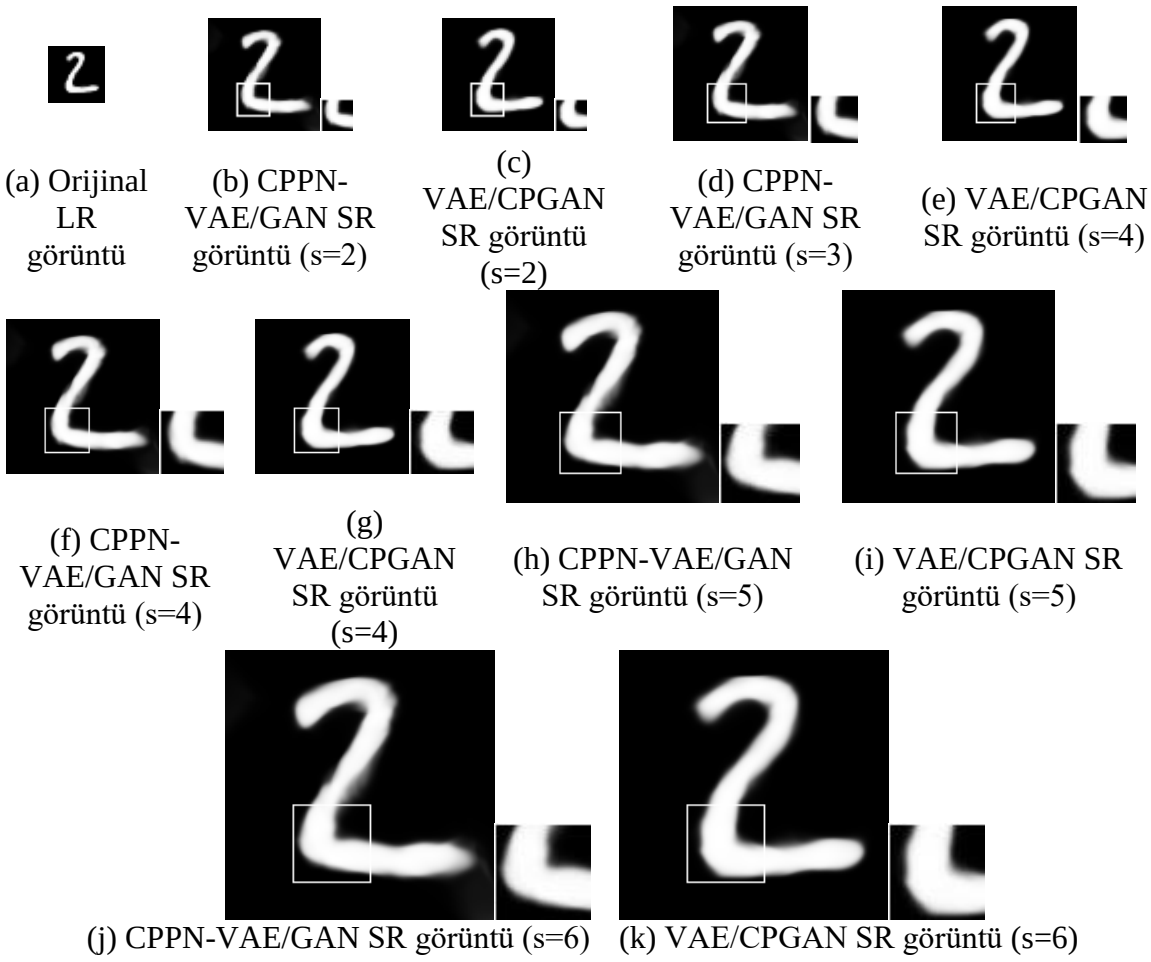
(f) En yakın komşu interpolasyon SR görüntüleri (s=4)



(g) Model SR görüntüleri (s=4)

Resim 6.4. VAE/CPGAN ve en yakın komşu interpolasyon ile elde edilen SR görüntüleri

CPPN tabanlı modellerin ölçeklenebilirlik özelliği sayesinde, VAE/CPGAN modelimiz örnekleri yüksek çözünürlüklü olarak üretebilmeyi de mümkün kılmaktadır. VAE/CPGAN modeli ile elde edilen yüksek çözünürlüklü görüntüler farklı ölçeklerde (2, 3 ve 4) en yakın komşu interpolasyon örnekleri ile Resim 6.4'te karşılaştırılmıştır. Bu örnekler incelendiğinde farklı ölçeklerde herhangi bir eğitim yapılmaksızın önerilen modelin yüksek çözünürlüklü görüntüler üretebildiği görülmektedir.



Resim 6.5. CPPN tabanlı modeller ile elde edilen SR görüntüleri

Yüksek çözünürlüklü görüntüleri karşılaştırabilmek üzere, farklı ölçeklerde herhangi bir eğitim yapılmaksızın yüksek çözünürlüklü görüntü oluşturabilen modeller karşılaştırılmak istenmiştir. Bu nedenle, literatürde mevcut olan süper çözünürlük modelleri bu karşılaştırmalara dahil edilememiştir. Bu nedenle, diğer CPPN tabanlı model olan CPPN-VAE/GAN ile önerilen VAE/CPGAN modeli farklı ölçeklerde (2, 3, 4, 5 ve 6) karşılaştırılmıştır. Resim 6.5'te karşılaştırılan görüntüler incelendiğinde, diğer modelin modelimize kıyasla yüksek çözünürlüklü görüntülere bazı gürültüler eklediği sonucuna

ulaşılabilmektedir. Önerilen model ile ölçek 6 gibi çok yüksek çözünürlüklerde bile daha net görüntüler elde edebildiği görülmektedir.

Süper çözünürlük analizinde kullanabilmek için MNIST veri kümesini oluşturan el yazısı görüntülerinin yüksek çözünürlüklü versiyonları mevcut olmadığı için, elde edilen yüksek çözünürlüklü görüntülerin bir metriğe göre nasıl değerlendirilebileceği araştırılmıştır. Görüntülerin netlik bilgisini karşılaştırmak üzere Laplasların varyansı netlik skoru olarak kullanılarak netlik oranları hesaplanmıştır. Deneylerde kullanmak üzere netlik için eşik değeri, λ_{th}, 100 olarak ayarlanmıştır. Çizelge 6.6'da elde edilen netlik skorları ve oranları verilmiştir. Bu skorlara göre önerilen model ile diğer CPPN tabanlı model olan CPPN-VAE/GAN karşılaştırılmıştır. Bu sonuçlara göre CPPN-VAE/GAN modeli Laplasların varyansına göre hesaplanan netlik skoruna göre modelimizden biraz daha net görüntüler oluşturabilen model olarak değerlendirilmiştir.

Çizelge 6.6. CPPN tabanlı modeller ile elde edilen SR görüntülerinin netlik skoru ve oranına göre karşılaştırmaları

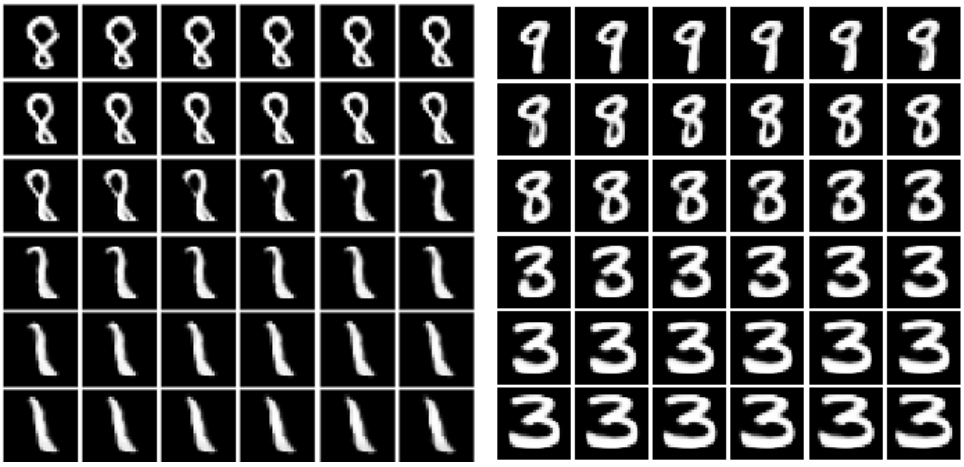
Model		CPPN-VAE-GAN	VAE/CPGAN
ölçek (s) =2	netlik skoru	1647,0256±432,4643	1538,5852±428,1420
	netlik oranı	1,0000	1,0000
ölçek (s) =3	netlik skoru	509,7392±137,2906	450,7175±129,2560
	netlik oranı	1,0000	1,0000
ölçek (s) =4	netlik skoru	203,0867±55,8707	175,8523±52,6528
	netlik oranı	0,9901	0,9586

Diğer bir deney, el yazısı stillerinin gizli kod olarak ifade edilen vektörler aracılığıyla transferini incelemek üzere yapılmıştır. Literatürde sunulan GAN modellerinde olduğu gibi istenen bir el yazı stili gizli kodların aritmetiği ile daha farklı bir stile dönüştürülebilmiştir. Bu çalışmalarda olduğu gibi, model ile gizli kod aritmetiği yapılarak stil transferi gizli kodlar aracılığıyla gerçekleştirilmeye çalışılmıştır. Resim 6.6'da verilen örneklerde olduğu gibi bir yazı stilindeki örneğin gizli kod vektörüne elde etmek istenen rakama ait gizli kod vektörü eklenerek istenmeyen stil vektörü ise çıkarılarak elde edilen gizli kod vektörü kodçözücü ağa girdi olarak verilerek istenen stilde rakamın elde edilebildiği görülmüştür. 7 sayısının istenen stilde elde edilmek istendiği ilk örnekte, italik 1 sayısına ait vektöre italik

olmayan bir 7 vektörü eklenmiş, italik olmayan 1 vektörü ise çıkarılarak hedeflenen gizli vektör elde edilmiştir. Bu aritmetik işlem sonucunda elde edilen vektörden kodçözücü ağ ile italik 7 rakamına ait el yazısı görüntüsü sentetik olarak elde edilebilmiştir.

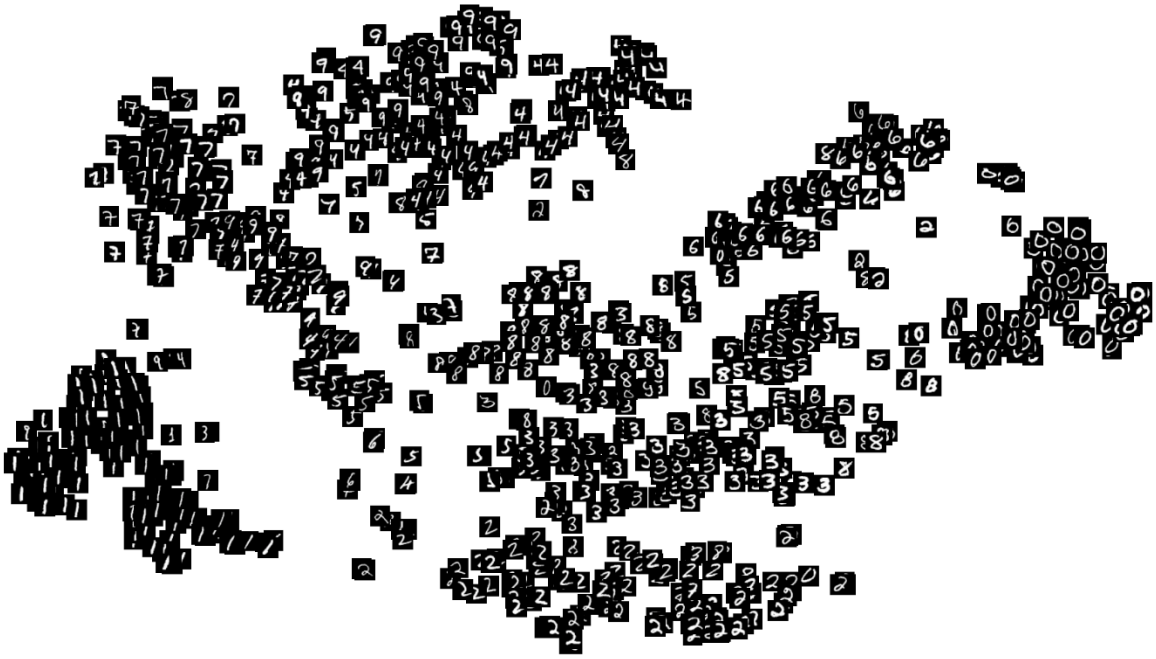
Resim 6.6. Gizli kod aritmetiği örnekleri

Son deney olarak, gizli kod vektörü üzerinde doğrusal interpolasyon yapılarak elde edilen yeni gizli kodlardan kodçözücü ağ kullanılarak adım adım farklı rakama ait el yazısı görüntüleri elde edilmiştir. Resim 6.7'de verilen görüntüler incelendiğinde, ilk interpolasyon örneğinde 8 rakamından 1 rakamına nasıl adım adım geçildiği gözlemlenebilmektedir. Benzer şekilde, 9 rakamından 8 ve 3 rakamlarının adım adım nasıl elde edildiği gösterilmektedir.



Resim 6. 7. Gizli kod interpolasyon örnekleri

Bu durumun, gizli kodlarının birbirine yakın dağılıma sahip örnekler arasında yapılabildiği gözlemlenmiştir. Gizli kodların t-SNE ile izdüşümleri Şekil 6.7'de görselleştirilmiştir. Bu izdüşümlere göre interpolasyon örneklerinin yakın dağılıma sahip örnekler olduğu dikkat çekmektedir.



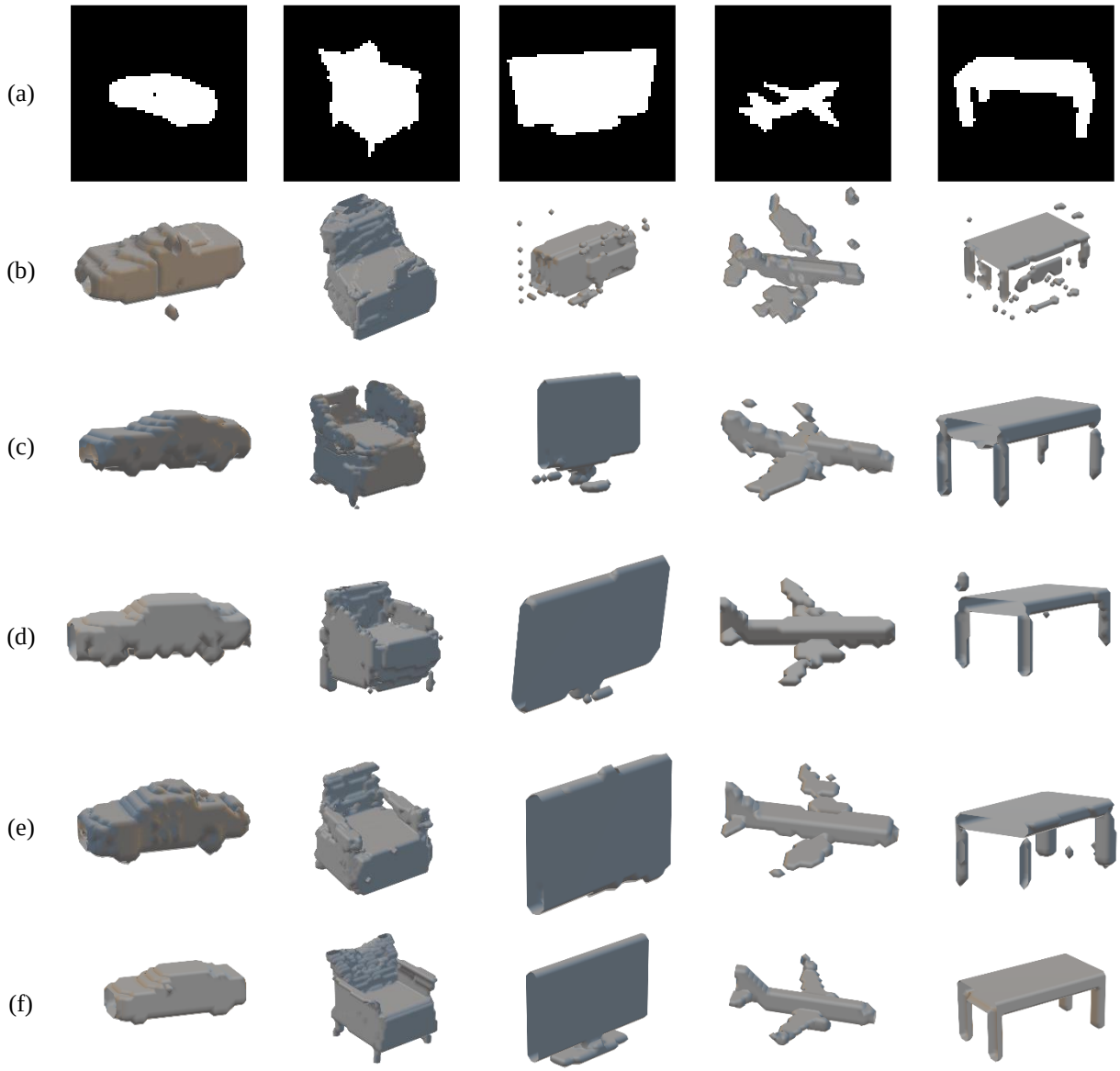
Resim 6.8. MNIST gizli kod t-SNE izdüşümleri [177]

Bu örneklerden anlaşılabileceği gibi önerilen VAE/CPGAN modeli ile hedeflendiği gibi veri dağılımı öğrenilebilmiş ve öğrenilen veri dağılımından model ile istenen veriler örneklenebilmektedir.

6.4.2. VoxCAE/GAN model karşılaştırmaları

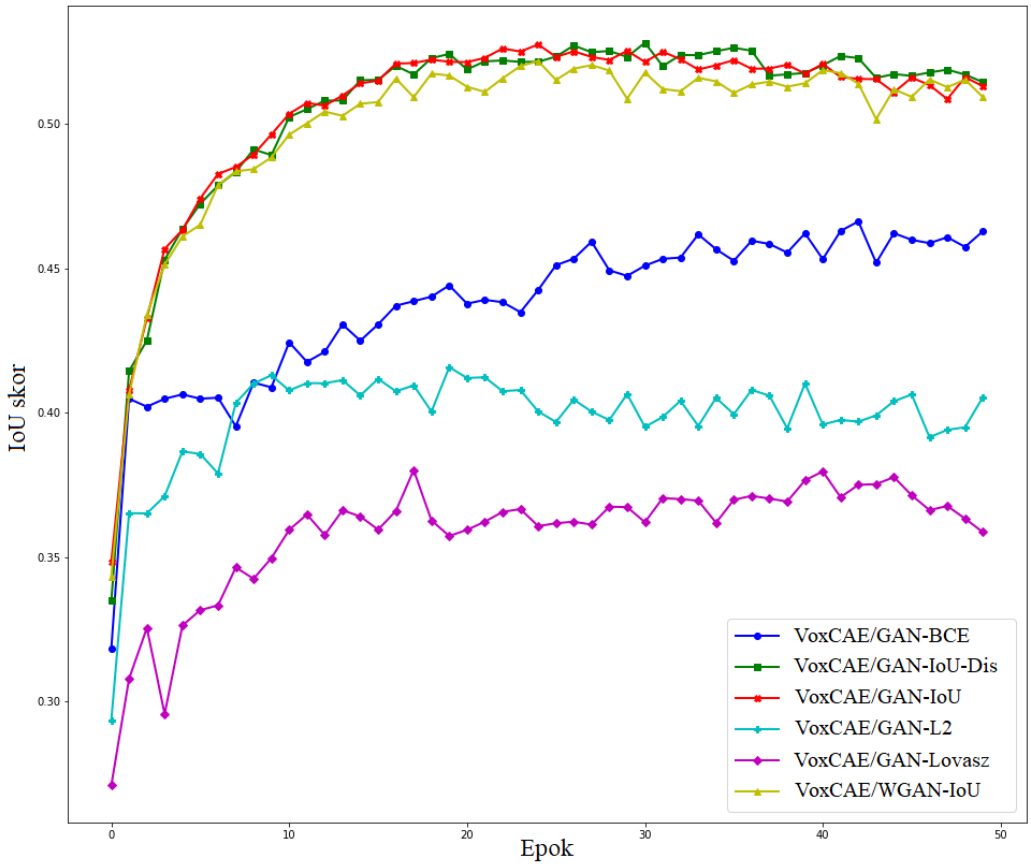
VoxCAE/GAN modeliyle hedeflenen silüetten nesneye dönüşüm problemi için deneysel çalışmalarda farklı amaç fonksiyonlarının nesne oluşturma performansına etkisi analiz edilmiştir. Bu nedenle, geri-çatım maliyet fonksiyonları üzerinde çalışılmıştır. Literatürde mevcut olan problemlerde geri-çatım maliyet fonksiyonu için MSE ya da L2 olarak ifade edilen metrik ile ikili çapraz entropi (BCE) kullanılmıştır. Bu çalışmalardan farklı olarak, sunulan modelde IoU maliyet fonksiyonu tabanlı bir eniyileme yapılması hedeflenmiştir. Oluşturulan sentetik verileri değerlendirmek üzere kullanılan IoU metriğinin türevi alınamaz yapısı nedeniyle daha önce nesne bölütleme problemi için önerilen türevlenebilir versiyonu [178] VoxAE/GAN modeline adapte edilmiştir. Deney çalışmalarında, bu şekilde elde edilen model VoxCAE/GAN-IoU olarak adlandırılmıştır. Amaç fonksiyonuna ayrımcı ağ ile üç boyutlu nesnelerden çıkarılan yüksek seviyeli özniteliklerin benzerlik ölçütü eklenerek ise VoxCAE/GAN-IoU-Dis modeli elde edilmiştir. IoU skorunu eniyilemek üzere sunulan yeni yaklaşımlardan biri olan Lovasz-hinge maliyet fonksiyonu [179], nesne geri-çatım maliyet

fonksiyonu olarak kullanılmak üzere bu çalışmayla analiz edilmiştir. Lovasz-hinge maliyet fonksiyonu tabanlı oluşturulan model ise VoxCAE/GAN-Lovasz olarak adlandırılmıştır. Ayrıca, yakın zamanda sunulan çalışmalarda [56, 180] tercih edilmesi sebebiyle Gradyan Cezalı Wasserstein Üretici Çekişmeli Ağ (WGAN-GP) [181] olarak adlandırılan Wasserstein uzaklığı tabanında iyileştirilen çekişmeli eğitim yaklaşımını analiz etmek üzere VoxCAE/WGAN modeli de oluşturulmuştur. Böylece, tez çalışması kapsamında üç boyutlu nesne oluşturma problemi için ilk kez IoU tabanlı objektife dayalı eniyileme çalışması yapılmıştır.



Resim 6.9. VoxCAE/GAN modelleri ile yeniden oluşturulan nesneler a) silüet görüntü b) VoxCAE/GAN-L2 c) VoxCAE/GAN-IoU d) VoxCAE/GAN-IoU-Dis e) VoxCAE/WGAN-IoU f) Yer gerçekliği

Deneysel çalışmalarda yeniden oluşturulan nesneleri niteliksel olarak analiz etmek üzere 5 nesne sınıfı için oluşturulan nesneler ile nesneleri oluşturulmak üzere modele girdi olarak verilen test silüet görüntüleri Resim 6.8'de gösterilmiştir. Resim 6.8'de görüldüğü gibi silüet görüntüleri farklı bakış açılarından elde edilmiştir. Şekilde de görüldüğü gibi nesnelerin oldukça düşük boyutlu olması nesnelerin anlaşılması ve nesne ilgili detayların yakalanarak gösterimlerin çıkarılmasını daha zor kılmaktadır. Şekildeki niteliksel sonuçlara bakılarak IoU tabanlı modellerin diğer modellerden daha iyi sonuç verdiği görülmektedir. Literatürde tercih edilen L2 tabanında geri-çatım maliyetine göre eğitilen model ile diğer modellere göre gürültü içeren nesneler oluşturulduğu gözlemlenmiştir.



Şekil 6.7. VoxCAE/GAN modellerinin eğitim süresince IoU skor grafiği

Niceliksel analiz yapmak üzere literatürdeki diğer çalışmalarda olduğu gibi IoU skoru tabanlı değerlendirmelere yer verilmiştir. IoU skoru 0 ile 1 arasında değişebilen bir değerdir. IoU değeri 1'e ne kadar yakın ise yeniden oluşturma performansı o kadar iyi kabul edilmektedir. IoU skoru Eş. 6.1'de verildiği gibi nesnelerin kesişimlerinin bileşimlerine oranı şeklinde hesaplanmaktadır.

$$IoU = \frac{\cap (y, \hat{y})}{\cup (y, \hat{y})} \tag{6.1}$$

Şekil 6.7'de analiz edilen VoxCAE/GAN modellerinin eğitim süresince her bir iterasyonda IoU skor değerlerine ait grafiği verilmiştir. Grafik incelendiğinde IoU tabanlı modellerin diğer yaklaşımlara göre daha hızlı ve iyi yakınsandığı görülmektedir.

VoxCAE/GAN modellerinin test IoU skoruna göre karşılaştırmalarına Çizelge 6.7'de yer verilmiştir. Çizelgede test verileri üzerinde tüm kategoriler için IoU skor ile ortalama IoU skor karşılaştırmaları mevcuttur. Çizelge incelendiğinde en iyi IoU skorlarının IoU ve yüksek seviyeli öznitelikler tabanında (öznitelik tabanlı maliyet fonksiyonu) eğitilen model olan VoxCAE/GAN-IoU-Dis ile elde edildiği görülmektedir. IoU maliyet fonksiyonu ile L2 ve BCE tabanında maliyet fonksiyonlarından tüm kategoriler için daha iyi performans elde edilebilmiştir.

Çizelge 6.7. Kategori tabanında ortalama test IoU karşılaştırmaları

	Kategori tabanlı IoU					
Metot	Airplane	Car	Chair	Display	Table	Ort. IoU
VoxCAE/GAN-BCE	0,3150	0,1618	0,0055	0,1045	0,0764	0,1285
VoxCAE/GAN-L2	0,3098	0,6834	0,2138	0,1689	0,2098	0,3490
VoxCAE/GAN-Lovasz	0,3426	0,5362	0,1947	0,1973	0,1550	0,2940
VoxCAE/GAN-IoU	0,3269	0,7450	0,3294	0,3473	0,3357	0,4419
VoxCAE/GAN-IoU-Dis	0,4010	0,7038	0,3309	0,3584	0,3526	0,4474
VoxCAE/WGAN-IoU	0,3718	0,7199	0,2847	0,3212	0,3478	0,4334

VoxCAE/GAN-IoU modelinin performansını mevcut modeller ile karşılaştırmak üzere bu alandaki en yüksek sonuçların kaydedildiği modeller olarak nitelendirilen 3d-recon ve 3DR2N2 modelleri tercih edilmiştir. Karşılaştırmalarda kullanılan model mimarileri ve özellikleri Çizelge 6.8 ve Çizelge 6.9'da verilmiştir. Bu modeller ile sunulan VoxCAE/GAN-IoU modeli tek açıdan görüntüden çok kategorili modelleme ile nesne oluşturma problemini çözebilmek üzere karşılaştırılmıştır.

Çizelge 6.8. Karşılaştırılan model mimarileri

Metot	Model mimarisi				Objektif
	Kodlayıcı ağ	Kodçözücü ağ	Ayrımcı ağ	Diğer	
3d-recon	2B conv	3B conv	2B conv	Perspektif projeksiyon	• Yeniden yapılandırma hatası • Poz-bağımsız gösterim hatası • Poz-bağımsız volüm
3DR2N2	2B conv	3B conv LSTM	-	-	3B voksel tabanlı softmax
VoxCAE/GAN-IoU	2B conv	3B conv	2B conv	-	• IoU tabanlı yeniden yapılandırma hatası • Çekişmeli eğitim hatası

Çizelge 6.9. Karşılaştırılan model özellikleri

Metot	Girdi	Poz/sınıf bilgisi	Eğitim	Parametre sayısı
3d-recon	32 × 32	poz	2B	37M
3DR2N2	64 × 64 × 3	-	3B	34M
VoxCAE/GAN-IoU	64 × 64	sınıf	3B	28M

Model mimarileri ve özellikleri incelendiğinde, 3d-recon modelinin, karşılaştırılan diğer modellerin aksine 3B gözetimli öğrenmeye dayanan bir model olmadığı, bir poz vektörünü ayrıca girdi olarak alan ve bir projeksiyon katmanından oluşan 2B gözetimli öğrenmeye dayalı bir model olduğu dikkat çekmektedir. Projeksiyona dayalı bir model olması sebebiyle, 3d-recon modelinde yazarlar, kamera parametrelerini (azimut, yükseklik, dönüş açısı ve mesafe gibi) kullanabilmek üzere nesnelerden görüntüleri kendileri elde ederek 3DR2N2 modelinde ve diğer çalışmalarda kullanılan görüntülerden farklı görüntüler üzerinde çalışmışlardır. Bu çalışmada kullanılan görüntüler diğer modellerden farklı olarak 32×32 boyutlarına ölçeklendirilmiştir.

3DR2N2 modeli ise kodçözücü ağ mimarisi bakımından 3d-recon ve VoxCAE/GAN-IoU modelimizden farklı olarak 3B evrişimsel LSTM ağ mimarisine dayanmaktadır. Ayrıca, 3DR2N2 modelinde, silüet görüntüleri yerine RGB görüntüleri ile nesneler elde edilmektedir.

Karşılaştırılan modeller ve modelimiz arasındaki parametre sayısında önemli farklar olduğu dikkat çekmektedir. Bu durum, diğer modellerin daha uzun zamanda eğitilebilmeleri nedeniyle daha maliyetli modeller olarak değerlendirebileceğini ortaya koymaktadır.

Çizelge 6.10. 3d-recon, 3DR2N2 ve VoxCAE/GAN-IoU modelleri kategori bazlı ve ortalama IoU skorları

		Metot		
		3d-recon [57]	3DR2N2 [5]	VoxCAE/GAN-IoU
Kategori	Airplane	0,3957	0,4325	0,4027
	Car	0,5894	0,7452	0,7552
	Chair	0,3130	0,4217	0,3803
	Display	0,2174	0,3651	0,3782
	Table	0,2895	0,4115	0,3619
	Ortalama	0,3558	0,4659	0,4334

Çizelge 6.10'da modellerin kategori bazlı ortalama IoU skorları verilmiştir. Verilen sonuçlara göre, VoxCAE/GAN-IoU modelimiz ile tüm kategoriler için 3d-recon modelinden daha yüksek IoU skorları ile daha iyi performans elde edebilmiştir. 3DR2N2 modeli ile karşılaştırmalara bakıldığında, "Airplane, Chair ve Table" kategorilerinde 3DR2N2 modelinin daha iyi performans sergilediği, diğer kategorilerde ise önerdiğimiz VoxCAE/GAN-IoU modeli ile en yüksek IoU skorlarının elde edildiği görülmektedir. Bu durumun, 3DR2N2 model performansının nesne şekilleri hakkında çok daha fazla bilgi sağlayabilen RGB görüntüleri ile eğitime dayalı bir model olmasından kaynaklandığı değerlendirilmiştir.

6.4.3. VoxAE modellerine ait karşılaştırmalar

Bu bölümde çekişmeli eğitim ve sınıf bilgisi gibi etkenlerin nesne oluşturma problemindeki etkisini irdelemek üzere deney çalışmaları yürütülmüştür. Bu amaçla, VoxAE olarak adlandırılan voksel otokodlayıcı modeli ile sınıf bilgisinin kullanıldığı Koşullu Otokodlayıcı

(CAE) modeline dayalı VoxCAE ve Koşullu GAN (CGAN) benzeri sınıf bilgisi ile çekişmeli eğitime dayanan, önceki alt bölümde ele alınan VoxCAE/GAN modelleri karşılaştırılmıştır. Çok-kategorili eğitim ile elde edilen test sonuçları, kategori bazında IoU metriğine göre Çizelge 6.11'de verilmiştir. Çizelgeye göre tüm kategoriler bazında elde edilen en yüksek IoU skoru sınıf bilgisine dayalı koşullu üç boyutlu otokodlayıcı modeli (VoxCAE) ile elde edilmiştir. Bu model ile ortalama 0,5550 IoU skoru kaydedilmiştir. Bu yaklaşımı üç boyutlu otokodlayıcıya dayalı model olan VoxAE modeli ortalama 0,5169 IoU skoru ile takip etmektedir.

Çizelge kategori tabanlı incelendiğinde elde edilen en yüksek sonuçların 'car' kategorisinde kaydedilmiştir. Bu durum, örnek sayısının diğer kategorilerdeki örnek sayılarından fazla olması ile bu sınıfa ait örnekler arasındaki varyansın diğer kategorilere göre daha az olması ile ilişkilendirilmiştir. Tüm kategoriler incelendiğinde ise en yüksek IoU skorlarının VoxCAE modeli ile elde edildiği görülmektedir.

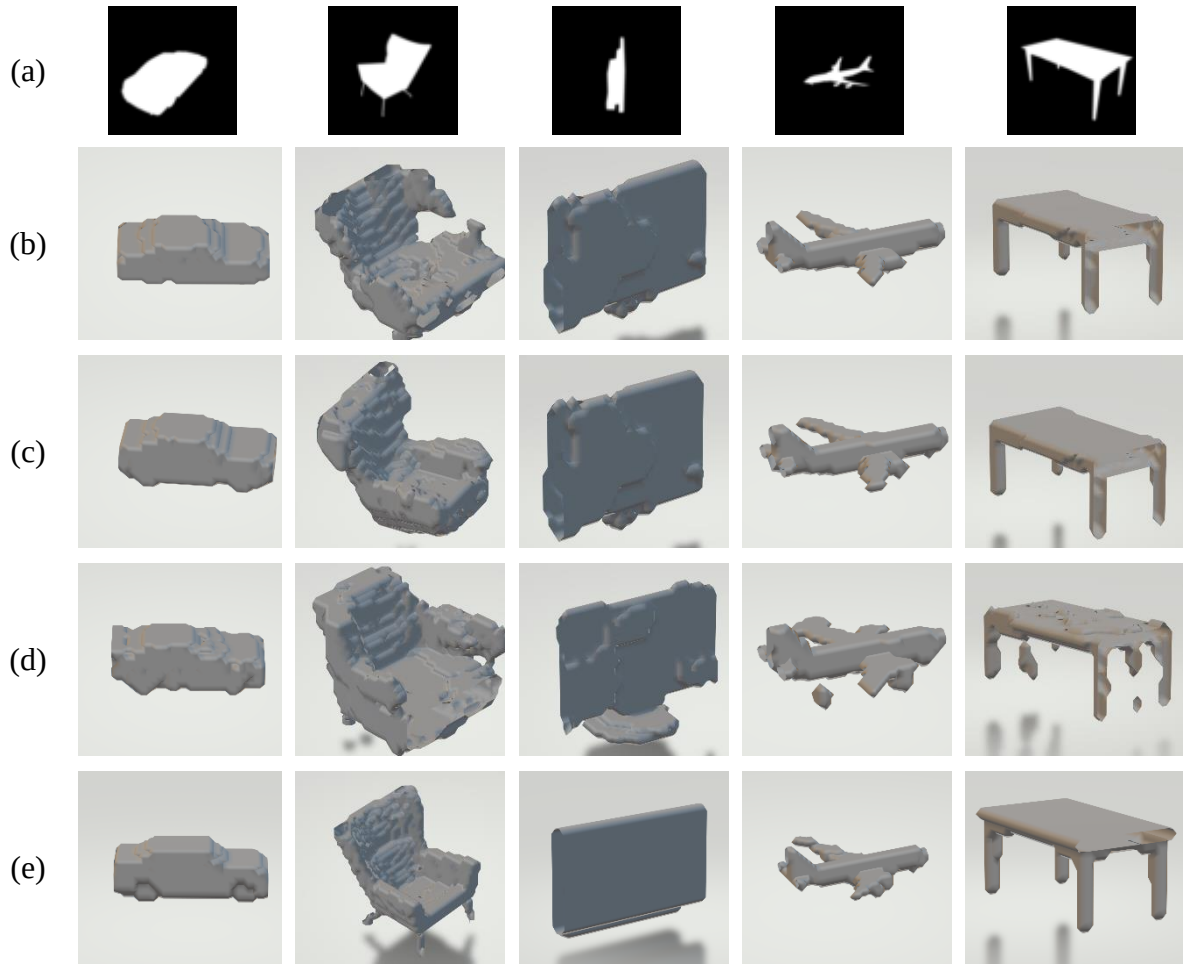
Çizelge 6.11. Çekişmeli eğitim ve sınıf bilgisine dayalı 3B yeniden yapılandırma performans karşılaştırmaları

	Kategori tabanlı IoU					
Metot	Airplane	Car	Chair	Display	Table	Ort. IoU
VoxAE	0,5010	0,7756	0,3785	0,4045	0,4272	0,5169
VoxCAE	0,5343	0,8140	0,4320	0,4340	0,4572	0,5550
VoxCAE/GAN	0,3965	0,7230	0,3099	0,3393	0,3519	0,4458

VoxAE, VoxCAE ve VoxCAE/GAN modelleri ile elde edilen nesneler ise Resim 6.10'da verilmiştir. Resim 6.10 (a)'da verilen tek bir silüet görüntüsünden bu modeller ile elde edilen nesneler gösterilmiştir. Resim 6.10 (e) 'de ise yer gerçekliği nesnelerine yer verilmiştir. Resimde verilen nesneler incelendiğinde, yer gerçekliği nesnelerine en yakın nesnelerin Resim 6.10 (c) ile gösterilen VoxCAE modeli ile elde edilebildiği görülmektedir.

Çekişmeli eğitime dayalı VoxCAE/GAN modeli ile elde edilen nesnelerin (Resim 6.10 (d)) diğer modeller ile elde edilen nesnelere göre (table nesnesinde olduğu gibi) gürültü ve (chair nesnesindeki gibi) bazı eksik nesne parçaları içerdikleri gözlemlenmiştir. Çekişmeli eğitime dayalı modellerin diğer modellerden daha zor ve uzun eğitim süreleri ve daha çok parametreye gereksinim duymaları ile birlikte global optimuma daha zor ulaşabilmesi

nedeniyle sadece otokodlayıcya dayalı modellerin performans olarak gerisinde kaldığı değerlendirilmiştir.



Resim 6.10. a) Farklı kategorilere ait silüet görüntüleri b) VoxAE nesneleri c) VoxCAE nesneleri d) VoxCAE/GAN nesneleri e) orijinal nesneler

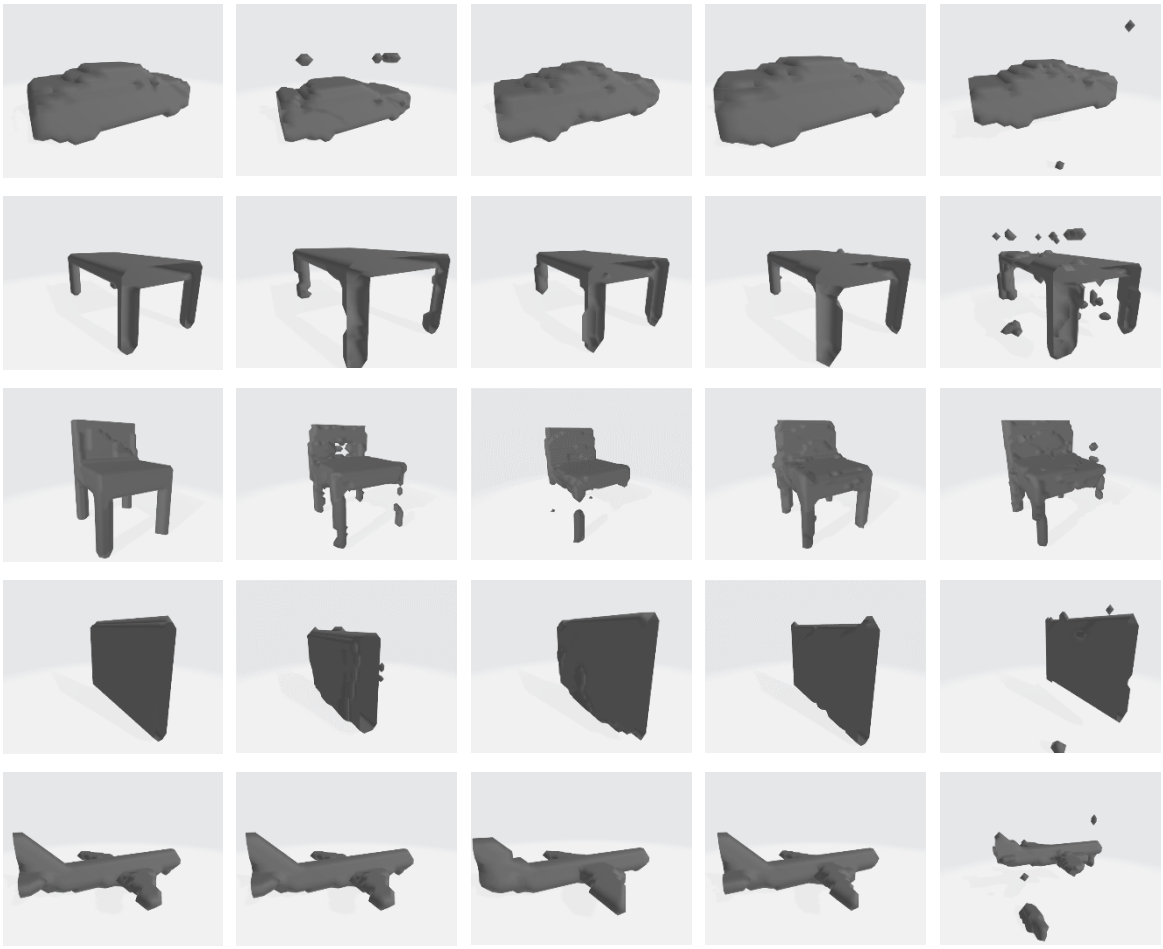
6.4.4. SkipVoxAE modellerine ait karşılaştırmalar

Bu bölümde önceki bölümlerde ele alınan VoxAE, VoxCAE ve VoxCAE/GAN modelleri ile SkipVoxAE olarak adlandırılan model ile karşılaştırmalara yer verilmiştir. Ayrıca, modelin farklı versiyonları tasarlanarak girişi verisinin tipinin, her bir örnek için kullanılan görüntü sayısının, etiket verisine dayalı olarak gözetimli öğrenmenin, çekişmeli eğitimin ve yeniden yapılandırma maliyet fonksiyonu için kullanılan metriklerin karşılaştırmalarına yer verilmiştir.

Çizelge 6.12. SkipVoxAE modelleriyle VoxAE, VoxAE/GAN model karşılaştırmaları

Model	Girdi	Görüntü sayısı	Model bileşenleri		Maliyet fonksiyonu	Ort. IoU	
			Otokodlayıcı	Ayrımcı ağ		Eğitim	Test
VoxAE-L2	2B Silüet	1	2B-3B CAE	-	L2	0,5897	0,5169
VoxAE-IoU	2B Silüet	1	2B-3B AE	-	IoU	0,5750	0,5344
VoxCAE-L2	2B Silüet	1	2B-3B CAE	-	L2	0,6011	0,5555
VoxCAE-IoU	2B Silüet	1	2B-3B CAE	-	IoU	0,5950	0,5391
VoxAE/GAN-L2	2B Silüet	1	2B-3B AE	3B CNN	• L2 • Çekişmeli eğitim	0,3739	0,3494
VoxAE/GAN-L2-Acc	2B Silüet	1	2B-3B AE	3B CNN	• L2 •Çekişmeli eğitim • Doğruluk oranı • Öznitelik tabanlı maliyet	0,2768	0,3689
VoxCAE/GAN-L2	2B Silüet	1	2B-3B AE	3B CNN	• L2 • Çekişmeli eğitim	0,3739	0,3494
VoxCAE/GAN-L2-Acc	2B Silüet	1	2B-3B AE	3B CNN	• L2 • Çekişmeli eğitim	0,2768	0,3689
VoxCAE/GAN-IoU	2B Silüet	1	2B-3B AE	3B CNN	• IoU • Çekişmeli eğitim	0,5058	0,4419
VoxCAE/GAN-Dis	2B Silüet	1	2B-3B AE	3B CNN	• IoU • Çekişmeli eğitim	0,5130	0,4474
VoxCAE/DGAN-L2	2B Silüet	1	2B-3B AE	3B gürültülü CNN	• L2 • Çekişmeli eğitim	0,4732	0,4458
VoxAE/DGAN-L2-Dis	2B Silüet	1	2B-3B AE	3B gürültülü CNN	• L2 •Çekişmeli eğitim • Doğruluk oranı	0,4805	0,4527
VoxAE/DGAN-L2-Acc-Dis	2B Silüet	1	2B-3B AE	3B gürültülü CNN	• L2 • Çekişmeli eğitim • Doğruluk oranı • Öznitelik tabanlı maliyet	0,4482	0,4143
SkipVoxAE-L2	2B Silüet	1	2B-3B AE	-	L2	0,6251	0,4605
SkipVoxAE-IoU	2B Silüet	1	2B-3B AE	-	IoU	0,6316	0,5404
SkipVoxCAE/GAN-L2	2B Silüet	1	2B-3B CAE	3B CNN	• L2 • Çekişmeli eğitim	0,3386	0,3311
SkipVoxCAE/GAN	2B Silüet	1	2B-3B CAE	3B CNN	IoU	0,4707	0,4617
SkipVoxCAE-L2	2B Silüet	1	2B-3B CAE	-	L2	0,6663	0,5513
SkipVoxCAE-IoU	2B Silüet	1	2B-3B CAE	-	IoU	0,6399	0,5557
SkipVoxCAE-IoU	2B Silüet	2	2B-3B CAE	-	IoU	0,6511	0,5811
SkipVoxCAE-IoU	2B Silüet	3	2B-3B CAE	-	IoU	0,6635	0,5848
SkipVoxCAE-IoU	2B Silüet	4	2B-3B CAE	-	IoU	0,6626	0,5580
SkipVoxCAE-IoU	2B RGB	1	2B-3B CAE	-	IoU	0,6516	0,5388
SkipVoxCAE-BCE	2B Silüet	1	2B-3B CAE	-	BCE	0,6399	0,5484

Ele alınan modeller Çizelge 6.12'de detaylı bir şekilde verilmiştir. Bu modeller ile daha önceki bölümlerde ele alınan VoxAE, VoxCAE/GAN tabanlı modellere ait karşılaştırmalara da yer verilmiştir. Çizelgede görülebildiği gibi SkipVoxAE modelinden çok sayıda model türetilebilmiştir. Verilen sonuçlar incelendiğinde, öngörülebildiği gibi, en yüksek performansın tek açıdan görüntü yerine birden çok açıdan görüntü kullanılarak elde edilebildiği görülmüştür. Tek açıdan görüntüler kullanıldığında ise bu performansa oldukça yakın sonuçlar elde edilebildiği gözlemlenmektedir.



(a) Yer gerçekliği nesneleri (b) VoxCAE-IoU nesneleri (c) VoxCAE/GAN-IoU nesneleri (d) SkipVoxCAE-IoU nesneleri (e) SkipVoxCAE/GAN-IoU nesneleri

Resim 6.11. ShapeNetCore yer gerçekliği nesneleri ile VoxCAE-IoU, VoxAE/GAN-IoU, SkipVoxCAE-IoU ve SkipVoxCAE/GAN-IoU yeniden yapılandırma ile elde edilen nesneler

Test verisi üzerinde elde edilen ortalama IoU skoruna göre en iyi performans veren modelin IoU metriğine dayalı SkipVoxCAE-IoU olduğu görülmektedir. Bu bölümde ele alınan, artık bağlantıya dayalı üretici model ile daha önceki bölümlerde ele alınan modellerde kaydedilen

performansın da iyileştirilmesi mümkün olmuştur. Önerilen SkipVoxCAE-IoU modeline en yakın performans sergileyen VoxCAE-L2 modeli ile kategori tabanında karşılaştırmalara ise Çizelge 6.12'de yer verilmiştir.

En yüksek performans elde edilen VoxCAE-IoU, VoxCAE/GAN-IoU, SkipVoxCAE-IoU ve SkipVoxCAE/GAN-IoU modelleri ile elde edilen üç boyutlu yeniden yapılandırma nesneleri Resim 6.11'de karşılaştırılmıştır. Resimde görüldüğü üzere yer gerçekliği nesnelerine en yakın nesneler SkipVoxCAE-IoU ile elde edilmiştir. Diğer modeller ile eksik nesneler oluşturulabildiği ve nesnelerin gürültülü veriler içerdiği görülmektedir.

Çizelge 6.13 ile Çizelge 6.16 arasında 3d-recon modeli [148] ile IoU ve AP tabanında performans karşılaştırmalarına yer verilmiştir. Her iki model, 3d-recon modelinde kullanılan veri kümesinde tanımlanan veri örneklerine göre, bölüm 6.1'de ayrıntılı şekilde verildiği gibi, 5 nesne ile tek bir model olarak eğitilmiştir. 3d-recon modelinde, 32x32x1'lik görüntüler ile birlikte 0,1 poz oranıyla poz bilgileri kullanılmış iken bizim modelimizde 64x64x1 görüntüler 1,0 oranında etiket bilgisi ile birlikte model eğitilmiştir.

Çizelge 6.13. Kategori-spesifik SkipVoxCAE ve 3d-recon model karşılaştırmaları

Kategori	Model	IoU			AP
		Maks.	Ortalama		
			t=0,4	t=0,5	
Airplane	3d-recon	0,4190	0,4040	0,4043	0,5319
	SkipVoxCAE	0,8741	0,6806	0,6800	0,8047
Car	3d-recon	0,6012	0,5893	0,5855	0,7000
	SkipVoxCAE	0,9478	0,8126	0,8122	0,8994
Chair	3d-recon	0,3382	0,3316	0,3315	0,4155
	SkipVoxCAE	0,8607	0,3967	0,3960	0,4964
Display	3d-recon	0,2511	0,2414	0,2393	0,2958
	SkipVoxCAE	0,9470	0,4338	0,4305	0,6288
Table	3d-recon	0,3113	0,3019	0,3013	0,4150
	SkipVoxCAE	0,9577	0,4765	0,4761	0,5829

Çizelge 6.13'de verilen sonuçlara göre, kategori-spesifik modeller arasında, modelimizin tüm kategori modelleri için 3d-recon modelini geride bıraktığı görülmektedir. Beklenildiği gibi, "Chair, Table, Display" gibi bazı kategoriler için modellerin bulguları incelendiğinde, bu kategorilerin modelleme için yeterli sayıda örnek içermemeleri ile model dağılımını

öğrenerek genelleştirmeyi zorlaştıran, örnekler arası çeşitlilikleri nedeniyle henüz hedeflenen sonuçlara erişilemediği görülmektedir.

Kategori-bağımsız olarak literatürde ifade edilen birden çok kategorili model değerlendirmeleri için, tüm kategorilerden elde edilen sınıf bilgileri kullanılarak, SkipVoxCAE tek bir model olarak eğitilmiştir. SkipVoxCAE modelimizin sınıf bilgisine gereksinim duyması en büyük kısıtlarından biridir. Bu kısıtı ortadan kaldırabilmek üzere yarı gözetimli öğrenmeye dayalı bir yaklaşım hedeflenmiştir. Bu amaçla, sınıf bilgileri 0,1 olasılıkta kullanılarak yarı gözetimli eğitim gerçekleştirilmiştir.

Çizelge 6.14. Tek açıdan görüntülere dayalı birden çok kategorili modellerin IoU ve AP skor karşılaştırmaları

Kategori	Model	Poz/ Sınıf bilgisi	Poz/Sınıf oranı	IoU			AP
				Maks.	Ortalama		
					t=0,4	t=0,5	
Airplane	3d-recon	Poz	0,1	0,3599	0,3113	0,1418	0,4895
			1,0	0,4022	0,3957	0,3959	0,5117
	SkipVoxCAE	Sınıf	0,1	0,8978	0,7029	0,7028	0,8471
			1,0	0,86131	0,7063	0,7061	0,8062
Car	3d-recon	Poz	0,1	0,5540	0,5303	0,2883	0,7142
			1,0	0,5980	0,5894	0,5894	0,7107
	SkipVoxCAE	Sınıf	0,1	0,9560	0,8196	0,8196	0,8681
			1,0	0,9519	0,8110	0,8110	0,8657
Chair	3d-recon	Poz	0,1	0,3086	0,2832	0,1093	0,3715
			1,0	0,3212	0,3130	0,3128	0,3854
	SkipVoxCAE	Sınıf	0,1	0,8905	0,4159	0,4156	0,4949
			1,0	0,8507	0,4321	0,4319	0,4962
Display	3d-recon	Poz	0,1	0,2133	0,2021	0,0946	0,2613
			1,0	0,2208	0,2174	0,2175	0,2581
	SkipVoxCAE	Sınıf	0,1	0,9908	0,4596	0,4594	0,3841
			1,0	0,9752	0,4738	0,4736	0,3523
Table	3d-recon	Poz	0,1	0,2850	0,2506	0,1021	0,3699
			1,0	0,2964	0,2895	0,2892	0,3754
	SkipVoxCAE	Sınıf	0,1	0,9571	0,4801	0,4799	0,6149
			1,0	0,9373	0,4823	0,4822	0,6018
Ort.	3d-recon	Poz	0,1	0,3442	0,3155	0,1472	0,4413
			1,0	0,3677	0,3610	0,3610	0,4483
	SkipVoxCAE	Sınıf	0,1	0,9908	0,5498	0,5496	0,7874
			1,0	0,9752	0,5574	0,5573	0,7790

Yarı gözetimli öğrenme ve gözetimli öğrenmeye dayalı modeller ile elde edilen sonuçlara Çizelge 6.14'te yer verilmiştir. Sonuçlar, yarı gözetimli ve gözetimli öğrenmeye dayalı model arasında anlamlı bir fark olmadığını göstermektedir. Bu durum, modelin sınıf bilgisine duyduğu gereksinim sınırlamasını ortadan kaldırarak yarı-gözetimli öğrenme ile benzer performans elde edilmesi sağlanmıştır. Karşılaştırmalarda kullanılan 3d-recon modeli, bizim model kısıtımıza benzer olarak 2B gözetimli bir model olması nedeniyle silüet görüntülerin yeniden elde edilmesi için poz bilgilerine gereksinim duymaktadır. Bu modelin limitli poz bilgisine dayalı versiyonu ise 0,1 oranında poz bilgileri kullanılarak elde edilmiştir.

Tüm modeller, Çizelge 6.14'te kategori bazlı olarak karşılaştırılmıştır. Tek açıdan görüntülere dayalı eğitim söz konusu olduğunda, modelimizin tüm kategoriler için 3d-recon modelinden çok daha iyi performans sergilediği tabloda yer alan bulgulara göre değerlendirilmiştir. 3d-recon modelinin tüm pozlara dayalı (poz oranı = 1,0) olarak eğitilen versiyonunun önerilen SkipVoxCAE modelin yarı-gözetimli halinden (sınıf bilgisi oranı = 0,1) çok daha düşük performans elde ettiği görülmektedir.

Tez çalışmasında asıl hedeflenen problem olan tek görüntüden üç boyutlu nesne oluşturma üzerine analizlere ek olarak birden çok açıdan görüntüden nesne oluşturma üzerine de değerlendirmeler yapılmıştır. Bu amaçla, her bir nesne örneği için birden çok poz görüntüleri kullanılarak nesne oluşturmak üzere modeller elde edilmiştir. Karşılaştırılan 3d-recon model için nesnenin 4 farklı poz görüntüleri kullanılmıştır. Önerilen SkipVoxCAE modeli için ise 2, 3 ve 4 farklı poz görüntüsü kullanılarak elde edilen modellerin yeniden yapılandırma performanslarına Çizelge 6.15'te yer verilmiştir. Tüm modeller için maksimum IoU ve farklı eşik değeri (t = 0,4 ve 0,5) için ortalama IoU skorları ile AP skor değerleri çizelgede verilmiştir. "Car" kategorisi dışındaki tüm kategorilerde en yüksek performansın SkipVoxCAE modelleri ile elde edildiği görülmektedir.

Çizelge 6.15. Birden çok açıdan görüntüye dayalı modellerin yeniden yapılandırma performans karşılaştırmaları

Kategori	Model	Poz/ Sınıf bilgisi	Poz/ Sınıf oranı	Objektif fonk.	Görüntü sayısı	IoU			AP
						Maks.	Ortalama		
							t=0,4	t=0,5	
Airplane	SkipVoxCAE	Sınıf	1,0	L_{recon}^{IoU}	2	0,9463	0,5839	0,5838	0,6705
					3	0,9801	0,5902	0,5900	0,6833
					4	0,9617	0,5922	0,5919	0,6865
	3d-recon	Poz	0,1	• L_{recon}^{L1+L2} • L_{pinv}^{L2} • L_{vinv}^{L1}	4	0,4222	0,3971	0,3193	0,5328
			1,0			0,5058	0,4816	0,4836	0,7105
Car	SkipVoxCAE	Sınıf	1,0	L_{recon}^{IoU}	2	0,9597	0,5768	0,5766	0,6656
					3	0,9898	0,5784	0,5782	0,6738
					4	0,9906	0,5848	0,5845	0,6782
	3d-recon	Poz	0,1	• L_{recon}^{L1+L2} • L_{pinv}^{L2} • L_{vinv}^{L1}	4	0,6975	0,6071	0,3884	0,8286
			1,0			0,7709	0,7540	0,7454	0,9185
Chair	SkipVoxCAE	Sınıf	1,0	L_{recon}^{IoU}	2	0,9702	0,5804	0,5802	0,6679
					3	1,0000	0,5829	0,5827	0,6768
					4	0,9920	0,5936	0,5933	0,6886
	3d-recon	Poz	0,1	• L_{recon}^{L1+L2} • L_{pinv}^{L2} • L_{vinv}^{L1}	4	0,3686	0,3231	0,2071	0,4602
			1,0			0,4627	0,4114	0,3757	0,5877
Display	SkipVoxCAE	Sınıf	1,0	L_{recon}^{IoU}	2	0,9525	0,5976	0,5975	0,6809
					3	0,9469	0,5898	0,5894	0,6804
					4	0,9605	0,6088	0,6084	0,7023
	3d-recon	Poz	0,1	• L_{recon}^{L1+L2} • L_{pinv}^{L2} • L_{vinv}^{L1}	4	0,2652	0,2328	0,1606	0,3147
			1,0			0,4316	0,3966	0,3853	0,5834
Table	SkipVoxCAE	Sınıf	1,0	L_{recon}^{IoU}	2	0,9550	0,5787	0,5785	0,6652
					3	0,9916	0,5848	0,5845	0,6795
					4	0,9745	0,5929	0,5925	0,6859
	3d-recon	Poz	0,1	• L_{recon}^{L1+L2} • L_{pinv}^{L2} • L_{vinv}^{L1}	4	0,3329	0,2633	0,1587	0,4440
			1,0			0,4157	0,3492	0,3074	0,5549

Çizelge 6.16. Çok kategorili modellerin kategori dışı örneklerdeki performans karşılaştırmaları

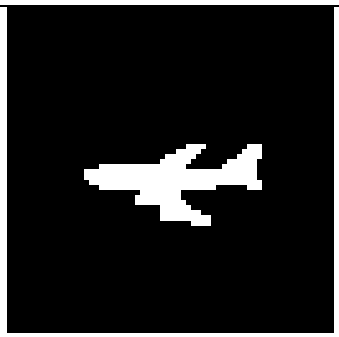
Kategori	Model	Poz/Sınıf bilgisi	Poz/Sınıf oranı	Görüntü sayısı	IoU			AP
					Maks.	Ortalama		
						t=0,4	t=0,5	
Bench	3d-recon	Poz	0,1	1	0,1937	0,1878	0,1870	0,2324
				4	0,2097	0,1570	0,0860	0,2572
			1,0	1	0,2331	0,2287	0,2285	0,2865
	SkipVoxCAE	Sınıf	0,1	1	0,9083	0,2217	0,2214	0,2644
			1,0	1	0,6878	0,2103	0,2100	0,2467
				2	0,7302	0,3416	0,3409	0,4305
				3	0,8037	0,3273	0,3264	0,4322
				4	0,9410	0,3449	0,3440	0,4462
Cabinet	3d-recon	Poz	0,1	1	0,3989	0,3766	0,3729	0,4854
				4	0,4519	0,3743	0,1537	0,5664
			1,0	1	0,4407	0,4295	0,4288	0,5283
	SkipVoxCAE	Sınıf	0,1	1	0,7732	0,4265	0,4259	0,5541
			1,0	1	0,7239	0,4006	0,4000	0,5131
				2	0,8184	0,3400	0,3395	0,4284
				3	0,7762	0,3306	0,3298	0,4334
				4	0,9410	0,3317	0,3308	0,4378
Vessel	3d-recon	Poz	0,1	1	0,3308	0,3215	0,3216	0,4093
				4	0,4096	0,3828	0,3130	0,5066
			1,0	1	0,3740	0,3657	0,3664	0,4489
	SkipVoxCAE	Sınıf	0,1	1	0,7165	0,3563	0,3561	0,3954
			1,0	1	0,7205	0,3704	0,3702	0,4024
				2	0,8287	0,3517	0,3511	0,4301
				3	0,7935	0,3169	0,3160	0,4145
				4	0,8023	0,3304	0,3295	0,4308

Karşılaştırılan model poza dayalı bir objektif fonksiyonuna (L_{pinv}^{L2}) daha dayalı eğitilmesine rağmen, verilen sonuçlara göre, önerilen modelin her kategoride birden çok poza görüntüsünden nesne oluşturmada en iyi performansı gösterdiği görülmektedir. Elde edilen sonuçlara göre, görüntü sayısı arttıkça modellerin nesne yapısını daha iyi anlayarak modelleyebildiği görülmektedir. Bu sonuçlar, Çizelge 6.15'de verilen tek görüntüye dayalı model sonuçları ile karşılaştırıldığında "Airplane, Car" gibi bazı kategoriler için tek görüntüden nesne oluşturma ile daha iyi sonuçlar elde edildiği görülmektedir. Bu durum,

bazı sınıflara ait örneklerin daha karmaşık yapıda olması nedeniyle birden çok poz görüntüsüne gereksinim duyması, bazı nesnelerin ise benzer geometrik yapıları nedeniyle daha kolay modellenmesi ile açıklanabilmektedir.

Ayrıca, kategori dışındaki örnek görüntülerden elde edilen yeniden yapılandırma nesnelerinin performans sonuçları Çizelge 6.16'de verilmiştir. Eğitimde kullanılan her bir nesneye ait görüntü sayısı, 0,1 ya da 1,0 olmak üzere poz/sınıf bilgisi oranları ile modellerin IoU ve AP skorlarına göre elde ettiği sonuçlara çizelgede yer verilmiştir. Öngörüldüğü gibi örnek sayısı model performansını olumlu şekilde etkilemektedir. “Bench” kategorisi dışında örnek sayısı artıkça modellerin yeniden yapılandırma kabiliyetlerinin arttığı görülmektedir. Karşılaştırılan 3d-recon modeli, yeniden yapılandırma hatası dışında poz tabanlı bir objektife daha dayalı olduğu için kategori dışındaki örneklerden “Cabinet” sınıfı için daha iyi performans elde etmiştir. Modelde hedeflenen poza dayalı objektife rağmen diğer kategori dışı sınıflar için benzer sonuçlar elde edilmiştir.

Yapılan niceliksel analizler ile elde edilen deney bulgularına ek olarak, yeniden yapılandırılan nesnelerin niteliksel olarak değerlendirilmesine de yer verilmiştir. Resim 6.11 ile Resim 6.15 arasında tek bir görüntüden elde edilen yeniden yapılandırma nesneleri verilmiştir. Oluşturulan sentetik üç boyutlu nesnelerin niteliksel olarak daha iyi analiz edilebilmesi için her bir nesnenin tek açıdan görüntülerinin karşılaştırılması yerine 4 farklı açıdan görüntüleri karşılaştırılmıştır. Yer gerçekliği nesnelerine ait görüntüler de karşılaştırmak üzere şekilde verilmiştir. Elde edilen yeniden yapılandırma nesneleri incelediğinde, "Airplane", "Car" gibi örnekler arası farklılıkların az olduğu kategorilerde model sonuçlarının oldukça gerçekçi olduğu değerlendirilmektedir. "Chair" ve "Table" gibi objelerin bazı parçalarında (özellikle bacak kısmında) eksiklikler dikkat çekmektedir. Bu durumun, bu kategori grubundaki objelerin kendi içerisindeki çeşitliliğinden kaynaklı olduğu değerlendirilmiştir. Önerilen SkipVoxCAE modelinin tek açıdan bir silüet görüntüsünden dahi üç boyutlu nesne oluşturabildiği görülmektedir. Resim 6.13’te olduğu gibi, nesnelerin özellikle arka açılardan elde edilen görüntüleri, nesnelerin şekli ve yapısı hakkında oldukça az bilgi verebilmektedir. Ayrıca, şekilde verilen, girdi olarak kullanılan silüet görüntüleri incelendiğinde, bu şekildeki düşük çözünürlüklü görüntülerden nesnelerin geometrik yapısının çıkarılabilmesinin insanlar için dahi zor bir problem olduğu görülmektedir.

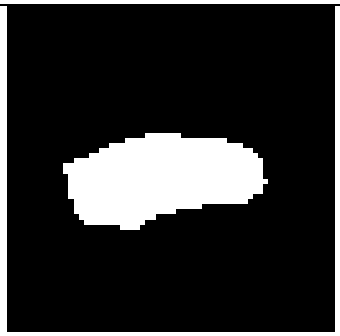


(a) Airplane objesi silüet giriş görüntüsü

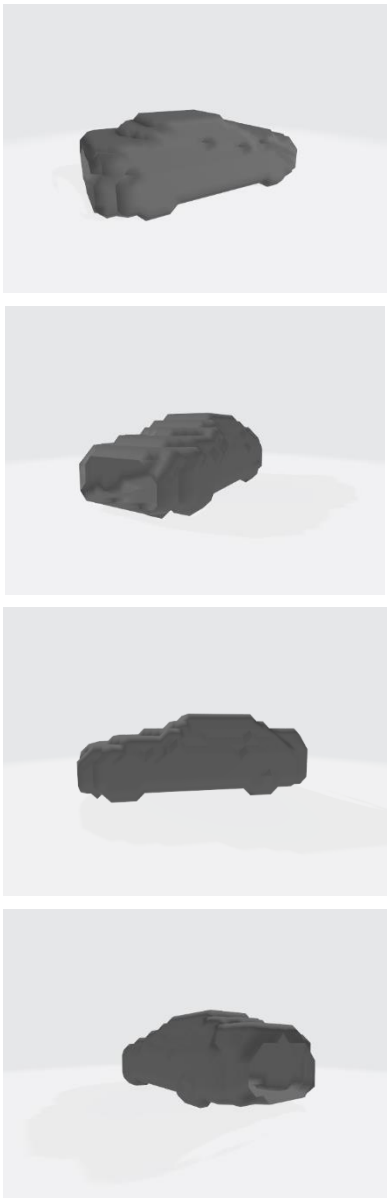
(b) Airplane yeniden yapılandırma nesne görüntüleri

(c) Airplane yer gerçekliği nesne görüntüleri

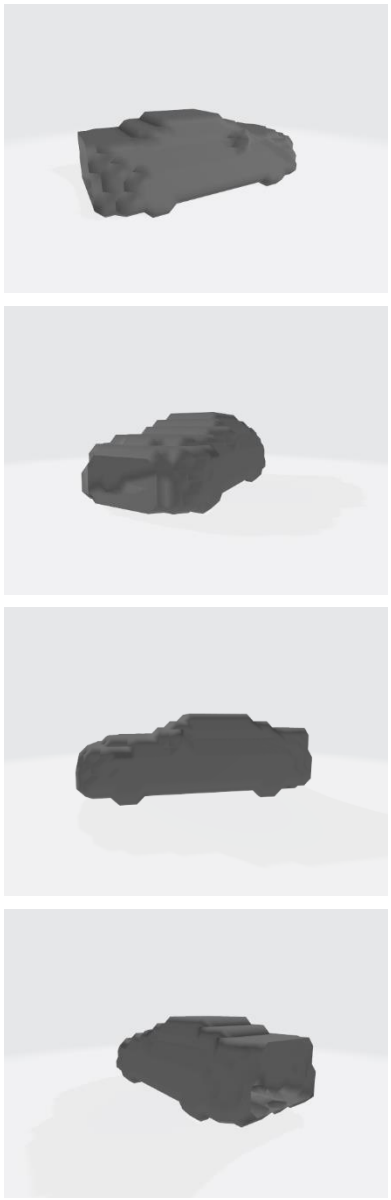
Resim 6.12. Airplane objesi yeniden yapılandırma ve yer gerçekliği nesne karşılaştırmaları



(a) Car objesi silüet giriş görüntüsü

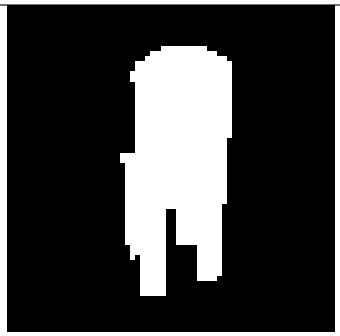


(b) Car yeniden yapılandırma nesne görüntüleri

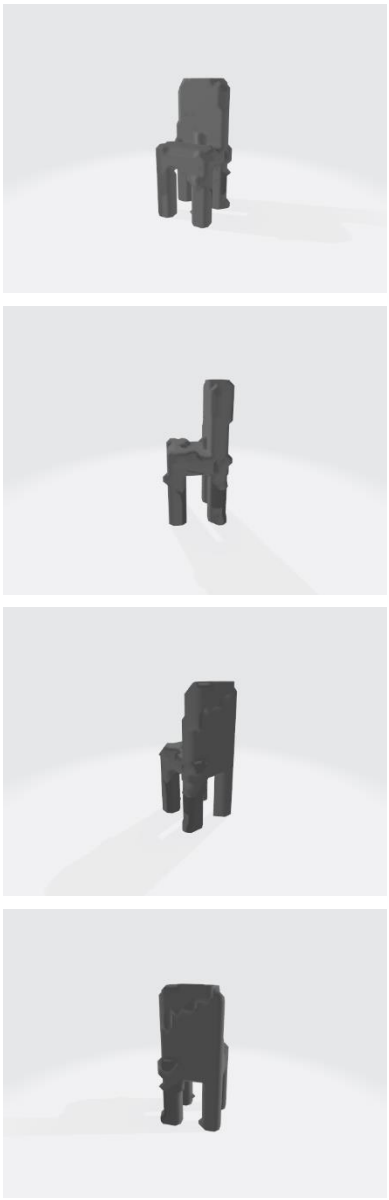


(c) Car yer gerçekliği nesne görüntüleri

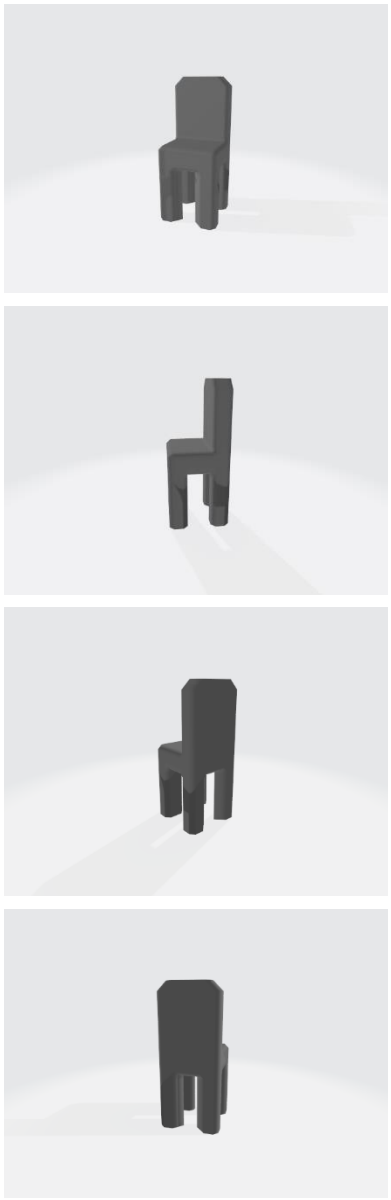
Resim 6.13. Chair objesi yeniden yapılandırma ve yer gerçekliği nesne karşılaştırmaları



(a) Chair objesi silüet giriş görüntüsü



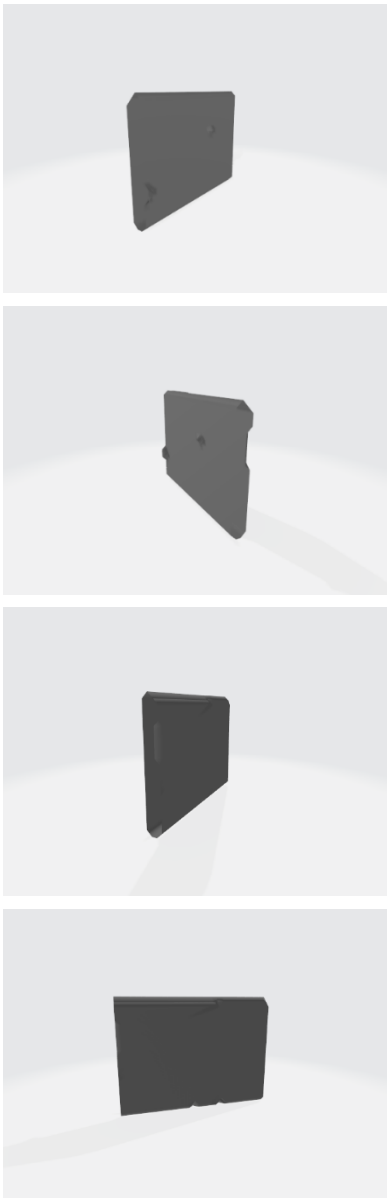
(b) Chair yeniden yapılandırma nesne görüntüleri



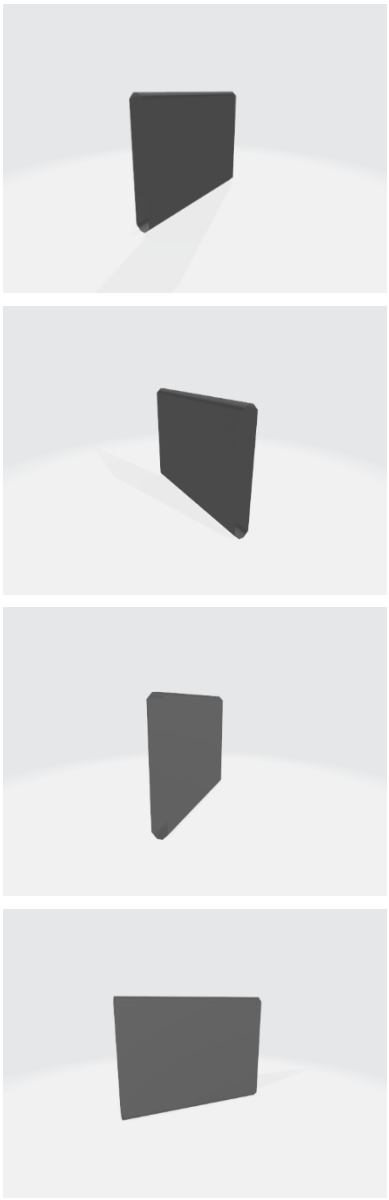
(c) Chair yer gerçekliği nesne görüntüleri

Resim 6.14. Chair objesi yeniden yapılandırma ve yer gerçekliği nesne karşılaştırmaları

(a) Display objesi silüet giriş görüntüsü

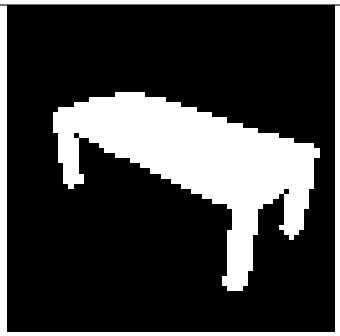


(b) Display yeniden yapılandırma nesne görüntüleri



(c) Display yer gerçekliği nesne görüntüleri

Resim 6.15. Display objesi yeniden yapılandırma ve yer gerçekliği nesne karşılaştırmaları



(a) Table objesi silüet giriş görüntüsü

(b) Table yeniden yapılandırma nesne görüntüleri

(c) Table yer gerçekliği nesne görüntüleri

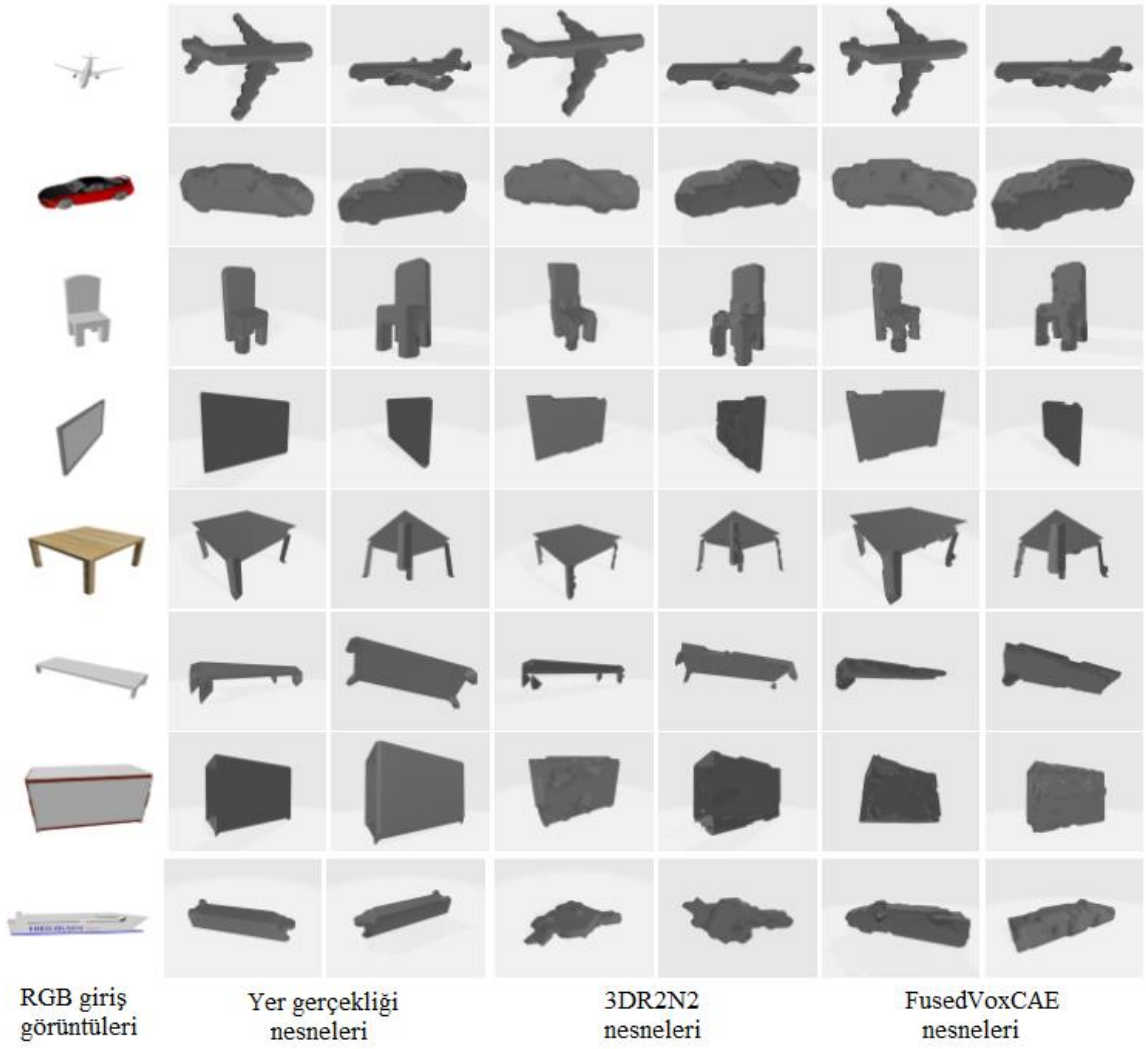
Resim 6.16. Table objesi yeniden yapılandırma ve yer gerçekliği nesne karşılaştırmaları

Bu model ile elde edilen bulgular genel olarak değerlendirildiğinde, önerilen modelin nesnenin birden çok görüntüsüne gereksinim duymadan herhangi bir bakış açısından elde edilen görüntüsünden dahi nesnelerin oluşturulabileceği görülmektedir. SkipVoxCAE modelinin, karşılaştırmalarda kullanılan silüet görüntüsüne dayalı popüler 3d-recon modelinden daha üstün olduğu deneysel sonuçlar ile ortaya koyulmuştur. Bu sonuçlar neticesinde, hem kategori-spesifik hem de kategori-bağımsız modellemede önerilen modelin performansı dikkat çekmektedir. Modelin kategori-bağımsız modellemede için sınıf bilgisine gereksinim duyması kısıtını ortadan kaldırmak üzere zayıf gözetimli öğrenme yaklaşımı benimsenmiştir. Bu yaklaşım, gözetimli öğrenme ile tüm sınıf bilgilerinin kullanıldığı model ile oldukça yakın performans elde edilmesini mümkün kılmıştır. Böylece, önerilen model ile zayıf gözetimli öğrenme ile dahi üç boyutlu yeniden yapılandırma performansının arttırılabildiği ortaya koyulmuştur. Elde edilen nesneler ve sonuçlar incelendiğinde, üretici modellere dayalı üç boyutlu yeniden yapılandırmanın neden umut vadeden bir yaklaşım olduğu daha iyi anlaşılmaktadır.

6.4.5. FusedVoxCAE model karşılaştırmaları

FusedVoxCAE modeli, literatürde oldukça ses getirilen, popüler ve en yüksek performanslı modellerden biri olarak değerlendirilen 3DR2N2 ile karşılaştırılmıştır. Ayrıca, modelin sadece derinlik (silüet) kodlayıcı ve sadece RGB kodlayıcıdan oluşan versiyonları da ele alınmıştır. Modellerin yeniden yapılanma performansı, daha önceki modellerde olduğu gibi IoU metriğine göre değerlendirilmiştir.

Modelin niteliksel karşılaştırmalarına Şekil 6.17'de yer verilmiştir. Modeller ile elde edilen nesnelerin iki farklı açıdan görüntüleri modelleri daha iyi analiz edebilmek üzere sunulmuştur. Resimde verilen görüntüler incelendiğinde modellerin yakın performans sergilediği dikkat çekmektedir. Eğitim dışı kategorilere ait nesneleri oluşturmada ise bazı kategorilerde FusedVoxCAE modeli daha başarılı görünür iken (vessel gibi) bazı kategorilerde (bench) ise 3DR2N2 ile daha yakın nesneler oluşturulabilmiştir.



Resim 6.17. FusedVoxCAE ve 3DR2N2 modelleri ile oluşturulan nesneler

Model versiyonları ile 3DR2N2 model karşılaştırmalarına Çizelge 6.17'de yer verilmiştir. Çizelgede sadece eğitimde kullanılan kategorilere ait görüntüler üzerinde değil farklı kategoriden görüntülerden de üç boyutlu nesne oluşturarak hesaplanan IoU skorları verilmiştir. Bu sonuçlara göre, RGB ve derinlik kodlayıcı tabanlı modelimizin (FusedVoxCAE), tüm kategoriler için karşılaştırılan 3DR2N2 modelinden çok daha iyi performans elde ettiği görülmektedir. Bu bölümde önerdiğimiz model ile üç boyutlu yeniden yapılandırma kapasitesinin kayda değer bir şekilde iyileştirildiği görülmektedir. Model versiyonları arasında ise, RGB ve derinlik kodlayıcı tabanlı modeli (FusedVoxCAE), "Display" ve "Vessel" kategorileri dışında en iyi sonuçları elde edebilmiştir. "Display" ve "Vessel" kategorilerinde ise derinlik kodlayıcı tabanlı modelimiz daha iyi sonuç vermiştir. Bu sonuçlar neticesinde, silüet özelliklerinin "Display" ve "Vessel" kategorilerini modellemek için yeterli olabileceği sonucuna varılabilmektedir.

Çizelge 6.17. FusedVoxCAE model karşılaştırmaları

Kategori	3DR2N2 [8]	Derinlik Kodlayıcı	RGB Kodlayıcı	RGB+Derinlik Kodlayıcı
Airplane	0,4410	0,6933	0,6894	0,6992
Car	0,7423	0,8142	0,8049	0,8172
Chair	0,4177	0,4151	0,4133	0,4252
Display	0,3652	0,4303	0,3801	0,4180
Table	0,4092	0,4840	0,4661	0,4872
Bench	0,2207	0,2418	0,2332	0,2553
Cabinet	0,3984	0,4473	0,4419	0,4667
Vessel	0,2546	0,3627	0,3013	0,3608

7. SONUÇ VE ÖNERİLER

Günümüzde derin öğrenme yaklaşımları, bilgisayarlı görü alanındaki birçok uygulamaları ile dikkat çekmektedir. Özellikle çekişmeli eğitim başta olmak üzere üretici ağların, makine öğrenmesinin en ilgi çekici konularından biri olarak kaydedildiği görülmüştür. Son yıllarda sunulan modellerin ağırlıklı olarak çekişmeli eğitime dayalı üretici modellerden oluştuğu görülmektedir. Üretici modellerin görüntü oluşturma probleminden süper çözünürlüğe kadar birçok bilgisayarlı görü problemlerindeki uyarlamaları nedeniyle tez çalışması kapsamında derin üretici ağ modellerine odaklanılmıştır. Çekişmeli eğitim yaklaşımı ile yeniden popülerliğini kazanan otokodlayıcılara göre üstünlüklerine rağmen otokodlayıcı yaklaşımlarında bulunan çıkarım mekanizmalarından yoksun olmaları nedeniyle çekişmeli eğitime dayalı üretici modeller ile otokodlayıcılar bir araya getirilmeye çalışılmıştır. Bu nedenle, VAE ve GAN hibrit modelleri üzerine çalışmalara bir eğilim söz konusu olmuştur.

Tezin ilk çalışmasında, literatürde yer alan hibrit modellerden etkilenilerek, sentetik görüntü oluşturabilmek üzere VAE ve GAN tabanlı bir modele odaklanılmıştır. Üretici modellerde elde edilen görüntülerin ölçeklenebilir olmaması nedeniyle CPPN modelinin piksel tabanında görüntü oluşturabilmesi nedeniyle üretici model olarak otokodlayıcı ve evrişimsel ayrımcı ağ ile modellenmiştir. Bu yaklaşımlar ile ölçeklenebilir bir üretici çekişmeli ağ modeli oluşturmaya çalışılmıştır. Bu üretici ağ ile veri dağılımının modellenebilmesi ile birlikte düşük boyutlu görüntünün gizli bir kodundan yüksek çözünürlüklü görüntülerin oluşturulmasını sağlayan görüntü yoğunluğunun da öngörülebilmesi hedeflenmiştir. Çekişmeli eğitime dayalı modellerde kullanılan eleman tabanlı yeniden yapılandırma ile birlikte özellik tabanında yeniden yapılandırma yaklaşımı benimsenerek ağın üretim kapasitesi arttırılmaya çalışılmıştır. Değişimsel bir kodlayıcı ve komposizyonel çekişmeli üretici ağ modeli ile daha çeşitli, net, gerçekçi ve yüksek çözünürlüklü görüntüler elde edilebilmiştir. Önerilen model ile karşılaştırma yapmak üzere kullanılan modele temel olan modeller ile IS skoruna göre yapılan karşılaştırmalarda, modelin performansı ortaya konmuştur. Önerilen model ile karşılaştırılan modellerin eğitim süresince performansları incelendiğinde ise önerilen modelin, GAN modellerinin eğitimde stabil olabilmesi için literatürde daha önce sunulan modellerine göre daha stabil performans sergilediği değerlendirilmiştir. Yapılan diğer deneysel çalışmalarda ise modelin veri dağılımını öğrenme konusunda da etkinliği ortaya konmuştur.

Görüntü oluşturma problemi için üretici ağlar ile kaydedilen yüksek performanslar, bu modellerin üç boyuta aktarılmasına yönelik çalışmalara neden olmuştur. Literatürde iki buçuk ve üç boyutlu olarak nitelendirilen verilerden yeniden üç boyutlu nesneler yüksek hesaplama maliyetlerine rağmen oluşturulabilmiş iken bu uygulamaların gerçek hayata uyarlanabilmesi için yer gerçekliği verilerinin mevcut olması gerekmektedir.

Tez çalışması kapsamında gerçek hayattan alınan tek bir görüntünün üç boyutlu nesne olarak modellenmesi hedeflenmiştir. Bu amaçla iki boyuttan üç boyuta dönüşüm problemine odaklanılmıştır. 3 boyuta dönüştürülmek üzere kullanılan görüntülerin sentetik olması ve gerçek görüntülerin aydınlanma, çevresel faktörler ve çok nesneli yapıları nedeniyle sentetik verilerden oldukça farklılık göstermesi nedeniyle doğrudan sentetik verilerden üç boyutlu nesneler oluşturmak yerine silüet verilerinden oluşturmaya odaklanılmıştır. Silüet görüntülerinden öznitelik çıkarmak ve üç boyutlu nesne oluşturmanın daha kolay olacağı öngörülürken nesnenin görsel şeklinin çıkarılabilmesi renkli görüntülerden daha zor olacaktır. Bu zorluklara rağmen, sentetik veriler için silüet görüntülerinden üç boyutlu nesneler oluşturulabilir ise gerçek görüntülerden bir bölütleme ile silüet görüntüler elde edilerek silüetten nesne oluşturabilen ağ modeli ile üç boyutlu nesnelerin elde edilmesi mümkün olacaktır. Nesnelerin şekil, tür gibi yapısal farklılıkları nedeniyle ise daha önceki çalışmalarda tercih edilen kategori-spesifik modeller yerine kategori-bağımsız model oluşturmak hedeflenmiştir.

Bu amaçla, son yıllarda oldukça popüler olan, GAN ve AE modelleri tabanlı hibrit bir modele odaklanılarak AE modellerin çıkarım mekanizması ile çekişmeli eğitim bir arada kullanılarak iki boyuttan üç boyuta dönüşüm problemi çözülmeye çalışılmıştır. VoxCAE/GAN olarak adlandırılan bu model için amaç fonksiyonları öncelikli olarak daha önceki çalışmalarda olduğu gibi L2 ya da çapraz entropi metriğine dayalı yeniden yapılandırma maliyet fonksiyonu ve çekişmeli eğitime dayalı olacak şekilde VoxCAE/GAN tasarlanmıştır. Bu şekilde tanımlanan modelin performansını arttırmak üzere VoxCAE/GAN modeli iyileştirmeye çalışılmıştır. Üretici modeller ile oluşturulan nesnelerin IoU metriğine dayalı olarak karşılaştırılması türevi alınabilir bir IoU fonksiyonuna dayalı objektifi eniyilemeye odaklanılmasına neden olmuştur. Bu amaçla, literatürde daha önce nesne bölütleme problemi için tanımlanan bir IoU objektif fonksiyonu yeniden yapılandırma maliyet fonksiyonu olarak uyarlanmış şekilde elde edilen model ise VoxCAE/GAN-IoU olarak adlandırılmıştır. Diğer yaklaşımlar ile IoU tabanlı model niceliksel ve niteliksel olarak ShapeNetCore veri

kümesinden elde edilen alt veri kümeleri üzerinde analiz edilmiştir. Yapılan analizlere göre IoU tabanlı yeniden yapılandırma ve yüksek seviyeli öznitelik benzerlik ölçütü dikkate alınarak yapılan eniyileme ile en iyi performansın elde edildiği görülmüştür. Elde edilen sonuçlara göre düşük çözünürlüklü ikili sayı şeklinde ifade edilen silüet görüntülerinde nesne yapısının öğrenilerek üç boyutlu nesne haline dönüştürülebildiği görülmüştür.

Son yıllarda oldukça önem kazanan ve bu alandaki son çalışmalarda da yararlanıldığı görülen çekişmeli eğitim yaklaşımının üç boyutlu nesne yapılandırma ve oluşturma problemine katkısı da incelenmiştir. Bu amaçla ele alınan VoxCAE ve VoxAE modelleri ile daha iyi performans sergilendiği kaydedilmiştir. Bu durum, çekişmeli eğitimine dayalı yaklaşım ile eğitimin stabil olamama durumu modellerin eniyilenmesinin daha zor olmasından kaynaklandığı şeklinde değerlendirilmiştir. Ele alınan modellerde sınıf bilgisinin kullanılmasının performansı arttırdığı görülmekle birlikte gerçek uygulamalar için bu bilgiye gereksinim duyulmasının modelin önemli bir kısıtı olmasından dolayı kısıtlı sınıf bilgisine (%10) dayalı bir yaklaşım ile görüntülerin daha düşük seviyeli özniteliklerinden de yararlanabilmek üzere sunulan SkipVoxCAE modeli ile performansın önemli derecede arttırılabildiği görülmüştür. Sentetik eğitim görüntüleri ile gerçek hayattan elde edilen görüntüler arasındaki farklılıklar nedeniyle bu verilere dayalı eğitimin gerçek verilerde başarısız olacağının öngörülmesi nedeniyle öncelikli olarak silüet tabanlı modellere odaklanılmıştır. Tek açıdan silüet görüntüsünden bir nesnenin geometrik olarak çıkarılabilmesinin güç olması nedeniyle ise silüet görüntüleri ile birlikte RGB görüntülerinden de faydalanabilmek üzere iki farklı kodlayıcı ile farklı katmanlardan elde edilen özniteliklerin füzyonundan nesne oluşturabilmek üzere son olarak FusedVoxCAE modeli önerilmiştir. Bu model ile RGB yerine siluet görüntülerinden nesne oluşturmanın etkinliği ortaya konulmuştur. En iyi performansın ise ise her iki tip görüntülerin füzyonu şeklindeki modelimiz ile kaydedilebildiği görülmüştür.

Sonuç olarak, tez çalışmasında öncelikli olarak, ölçeklenebilir bir GAN modeline odaklanılarak literatürdeki çalışmalardan farklı olarak tezin ilk çalışması olarak önerilen VAE/CPGAN modeli ile istenen boyutta ikili görüntülerin yeniden yapılandırılabildiği ve sentetik ikili görüntülerin oluşturabildiği gösterilmiştir. GAN modellerini 3 boyuta aktarabilmek üzere tezin devam eden kısmında ise tek açıdan görüntülerden nesne oluşturma ve yeniden yapılandırma problemi, literatürde bulunan çalışmalardan farklı bir objektif fonksiyonu ile çözümlenmiştir. Tez çalışması kapsamında önerilen VoxCAE/GAN, VoxAE,

VoxCAE, SkipVoxCAE ve FusedVoxCAE modelleri ile adım adım yeniden yapılandırma performansı iyileştirilerek alanın önde gelen çalışmalarına yakın ve daha iyi sonuçların elde edilebildiği gösterilmiştir. Elde edilen kategori-bağımsız modeller ile eğitim kategorileri dışındaki kategorilerde bile başarılı olduğu görülmüştür.

Doktora sonrası çalışmalarda, ölçeklenebilir GAN yaklaşımımız 3 boyuta transfer edilmeye çalışılarak hesaplama maliyeti arttırılmadan düşük çözünürlüklerde eğitim yapılarak düşük çözünürlüklü tek açıdan bir görüntüden daha yüksek çözünürlükte nesnelerin elde edilebilmesi hedeflenmektedir. Ayrıca, görüntülerin döşeme ve renk bilgilerinin de nesnelere transfer edilebilmesi amaçlanmaktadır. Tek objeden oluşan görüntüler yerine birden fazla obje içeren görüntülerin 3 boyuta dönüştürülmesi üzerine çalışmalar gerçekleştirilecektir. Nesne tamamlama problemi de ele alınarak nesne yapılandırma performansı daha da iyileştirilerek gerçek problemler için kullanılabilecek modellerin geliştirilmesi hedeflenmekted

KAYNAKLAR

1. LeCun, Y., Bengio, Y. ve Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

2. McCulloch, W. S. ve Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115-133.

3. Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386-408.

4. Hubel, D. H. ve Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3), 574-591.

5. Werbos, P. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science. Doctoral dissertation. Harward University.

6. Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4), 193-202.

7. Yann LeCun, B. B., John S. Derker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, Lawrence D. Jackel (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4), 541-551.

8. LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E. ve Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network, *Advances in Neural Information Processing Systems*, 396-404.

9. Salakhutdinov, R. ve Hinton, G. (2009). Deep boltzmann machines. *Artificial intelligence and statistics*, 448-455.

10, Krizhevsky, A., Sutskever, I. ve Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, Amerika Birleşik Devletleri, 1097-1105.

11. LeCun, Y., Bottou, L., Bengio, Y. ve Haffner, P. (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 2278-2324.

12. Simard, P. Y., Steinkraus, D. ve Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. *International Conference on Document Analysis and Recognition*, 3, 958-962.

13. Internet: Chellapilla, K., Puri, S. ve Simard, P. (2006). High performance convolutional neural networks for document processing. URL: https://hal.inria.fr/inria-00112631/document. Son Erişim Tarihi: 03.12.2018.

14. Zamir, A. R. ve Shah, M. (2014). Image geo-localization based on multiplenearest neighbor feature matching usinggeneralized graphs. *IEEE transactions on pattern analysis machine intelligence*, 36(8), 1546-1558.

15. Frome, A., Cheung, G., Abdulkader, A., Zennaro, M., Wu, B., Bissacco, A., Adam, H., Neven, H. ve Vincent, L. (2009). Large-scale privacy protection in street-level imagery, *International Conference on Computer Vision*, 2373-2380.

16. Garcia, C. ve Delakis, M. (2004). Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE transactions on pattern analysis machine intelligence*, 26(11), 1408-1423.

17. LeCun, Y., Kavukcuoglu, K. ve Farabet, C. (2010). Convolutional networks and applications in vision, *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 253-256.

18. Goodfellow, I. J., Shlens, J. ve Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv:1412.6572*.

19. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. ve Fei-Fei, L. (2015). ImageNet: Large scale visual recognition challenge. *International journal of computer vision*, 115(3), 211-252.

20, İnternet: Goodfellow, I. J. Deep learning adversarial examples clarifying misconception. URL: https://www.kdnuggets.com/2015/07/deep-learning-adversarial-examples-misconceptions.html. Son Erişim Tarihi: 21.12.2018.

21. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. ve Bengio, Y. (2014). Generative adversarial nets, *Advances in Neural Information Processing Systems*, Montréal, Kanada, 2672-2680.

22. İnternet: LeCun, Y. Yann LeCun Quora session overview. URL: http://www.kdnuggets.com/2016/08/yann-lecun-quora-session.html. Son Erişim Tarihi: 21.12.2018.

23. Hinton, G. E. ve Zemel, R. S. (1994). Autoencoders, minimum description length and helmholtz free energy, *Advances in Neural Information Processing Systems*, Denver, Colorado, Amerika Birleşik Devletleri, 3-10.

24. Kingma, D. P. ve Welling, M. (2013). Auto-encoding variational bayes. arXiv:1312.6114.

25. Larsen, L., Boesen, A., Sønderby, S. K., Larochelle, H. ve Winther, O. (2015). *Autoencoding beyond pixels using a learned similarity metric*. arXiv:1512.09300.

26. Lamb, A., Dumoulin, V. ve Courville, A. (2016). Discriminative regularization for generative models. *arXiv:1602.03220.*

27. Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M. ve Courville, A. (2016). Adversarially learned inference. *arXiv:1606.00704*.

28. Dosovitskiy, A. ve Brox, T. (2016). Generating images with perceptual similarity metrics based on deep networks, *Advances in Neural Information Processing Systems*, Barselona, İspanya, 658-666.

29. Gauthier, J. (2014). Conditional generative adversarial nets for convolutional face generation. URL: http://cs231n.stanford.edu/reports/2015/pdfs/jgauthie_final_report.pdf. Son Erişim Tarihi: 02.03.2019.

30, Radford, A., Metz, L. ve Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv:1511,06434*.

31. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I. ve Abbeel, P. (2016). InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets, *Advances in Neural Information Processing Systems*, Barselona, İspanya, 2172-2180.

32. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B. ve Lee, H. (2016). Generative adversarial text to image synthesis. *arXiv:1605.05396.*

33. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X. ve Metaxas, D. (2017). StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks, *IEEE International Conference on Computer Vision*, Venedik, İtalya, 5907-5915.

34. Yeh, R. A., Chen, C., Lim, T. Y., Schwing, A. G., Hasegawa-Johnson, M. ve Do, M. N. (2017). Semantic image inpainting with deep generative models, *IEEE conference on computer vision and pattern recognition*, Honolulu, Hawaii, 5485-5493.

35. Iizuka, S., Simo-Serra, E. ve Ishikawa, H. (2017). Globally and locally consistent image completion. *ACM Transactions on Graphics*, 36(4), 1-14.

36. Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O. ve Lik, H. (2017). High-resolution image inpainting using multi-scale neural patch synthesis, *IEEE conference on computer vision and pattern recognition*, Honolulu, Hawaii, Amerika Birleşik Devletleri, 6721-6729.

37. Liu, M.-Y., Breuel, T. ve Kautz, J. (2017). Unsupervised image-to-image translation networks, *Advances in Neural Information Processing Systems*, Kaliforniya, Amerika Birleşik Devletleri, 700-708.

38. Isola, P., Zhu, J.-Y., Zhou, T. ve Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks, *IEEE conference on computer vision and pattern recognition*, Honolulu, Hawaii, Amerika Birleşik Devletleri, 1125-1134.

39. Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O. ve Shechtman, E. (2017). Toward multimodal image-to-image translation, *Advances in Neural Information Processing Systems*, 465-476.

40. Aldoma, A., Marton, Z.-C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R. B., Gedikli, S. ve Vincze, M. (2012). Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. *IEEE Robotics & Automation Magazine*, 19(3), 80-91.

41. Guo, Y., Zhang, J., Lu, M., Wan, J. ve Ma, Y. (2014). Benchmark datasets for 3D computer vision, *9th IEEE Conference on Industrial Electronics and Applications*, Xi'an, Çin, 1846-1851.

42. Xie, H., Yao, H., Sun, X., Zhou, S., Zhang, S. ve Tong, X. (2019). Pix2Vox: Context-aware 3D Reconstruction from Single and Multi-view Images. *arXiv:1901.11153*.

43. Han, X.-F., Laga, H. ve Bennamoun, M. (2019). Image-based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era. *arXiv:1906.06543*.

44. Denton, E., Chintala, S., Szlam, A. ve Fergus, R. (2015). Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks, *Advances in Neural Information Processing Systems*, Montréal, Kanada, 1486-1494.

45. Makhzani, A., Jonathon, S., Navdeep, J., Goodfellow, I. ve Frey, B. (2015). Adversarial autoencoders. *arXiv:1511,05644*.

46. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. ve Chen, X. (2016). Improved techniques for training GANs, *Advances in Neural Information Processing Systems*, Barselona, İspanya, 2234-2242.

47. Arjovsky, M., Chintala, S. ve Bottou, L. e. (2017). Wasserstein GAN. *arXiv:1701,07875*.

48. Gregor, K., Danihelka, I., Graves, A., Jimenez, D. ve Wierstra, R. D. (2015). DRAW: A recurrent neural network for image generation. *arXiv:1502.04623*.

49. Tolstikhin, I., Bousquet, O., Gelly, S. ve Schölkopf, B. (2018). Wasserstein Auto-Encoders. *arXiv:1711,01558*.

50, Wu, J., Zhang, C., Xue, T., Freeman, W. T. ve Tenenbaum, J. B. (2016). Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling, *Advances in Neural Information Processing Systems*, Barselona, İspanya, 82-90.

51. Sharma, A., Grau, O. ve Fritz, M. (2016). VConv-DAE: Deep volumetric shape learning without object labels, *European Conference on Computer Vision*, Amsterdam, Hollanda, 236-250.

52. Girdhar, R., Fouhey, D. F., Rodriguez, M. ve Gupta, A. (2016). Learning a Predictable and Generative Vector Representation for Objects, *European Conference on Computer Vision*, Amsterdam, Hollanda, 484-499.

53. Brock, A., Lim, T., Ritchie, J. M. ve Weston, N. (2016). Generative and discriminative voxel modeling with convolutional neural networks. arXiv:1608.04236.

54. Liu, J., Yu, F. ve Funkhouser, T. (2017). Interactive 3D modeling with a generative adversarial network, International Conference on 3D Vision (3DV), Kopenhag, Danimarka, 126-134.

55. Smith, E. J. ve Meger, D. (2017). Improved Adversarial Systems for 3D Object Generation and Reconstruction. arXiv:1707.09557.

56. Yang, B., Wen, H., Wang, S., Clark, R., Markham, A. ve Trigoni, N. (2017). 3D object reconstruction from a single depth view with adversarial learning, *IEEE International Conference on Computer Vision*, Venedik, İtalya, 679-688.

57. Yang, B., Rosa, S., Markham, A., Trigoni, N. ve Wen, H. (2019). Dense 3D object reconstruction from a single depth view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12), 2820-2834.

58. Choy, C. B., Xu, D., Gwak, J., Chen, K. ve Savares, S. (2016). 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction, *European Conference on Computer Vision*, Amsterdam, Hollanda, 628-644.

59. Soltani, A. A., Huang, H., Wu, J., Kulkarni, T. D. ve Tenenbaum, J. B. (2017). Synthesizing 3D shapes via modeling multi-view depth maps and silhouettes with deep generative networks, *IEEE conference on computer vision and pattern recognition*, Honolulu, Hawaii, Amerika Birleşik Devletleri, 1511-1519.

60, Gadelha, M., Maji, S. ve Wang, R. (2017). 3D shape induction from 2D views of multiple objects, *2017 International Conference on 3D Vision (3DV)*, Qingdao, Çin, 402-411.

61. Gwak, J., Choy, C. B., Chandraker, M., Garg, A. ve Savarese, S. (2017). Weakly supervised 3D reconstruction with adversarial constraint, *2017 International Conference on 3D Vision (3DV)*, Qingdao, Çin, 263-272.

62. Wiles, O. ve Zisserman, A. (2017). SilNet : Single- and multi-view reconstruction by learning from silhouettes. *arXiv:1711,07888*.

63. Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L. ve Yu, F. (2015). ShapeNet: An information-rich 3D model repository. *arXiv:1512.03012*.

64. He, K., Zhang, X., Ren, S. ve Sun, J. (2016). Deep residual learning for image recognition, *IEEE conference on computer vision and pattern recognition*, Las Vegas, Nevada, Amerika Birleşik Devletleri, 770-778.

65. Ronneberger, O., Fischer, P. ve Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation, *International Conference on Medical image computing and computer-assisted intervention*, Münih, Almanya, 234-241.

66. İnternet: LeCun, Y., Cortes, C. ve Burges, C. J. C. The MNIST database of handwritten digits. URL: http://yann.lecun.com/exdb/mnist/. Son Erişim Tarihi: 10.02.2019

67. Lin, M., Chen, Q. ve Yan, S. (2013). Network in network. *arXiv:1312.4400*.

68. Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. University of Toronto, 1.

69. İnternet: Krizhevsky, A., Nair, V. ve Hinton, G. CIFAR-100 (Canadian Institute for Advanced Research). URL: http://www.cs.toronto.edu/~kriz/cifar.html. Son Erişim Tarihi: 06.05.2019

70. İnternet: Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B. ve Ng, A. Y. (2011). Reading Digits in Natural Images with Unsupervised Feature Learning. URL: http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf. Son Erişim Tarihi: 08.05.2019.

71. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. ve Rabinovich, A. (2015). Going deeper with convolutions, IEEE conference on computer vision and pattern recognition, Boston, Massachusetts, Amerika Birleşik Devletleri, 1-9.

72. Chatfield, K., Simonyan, K., Vedaldi, A. ve Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. arXiv:1405.3531.

73. İnternet: Everingham, M., Van Gool, L., Williams, C., Winn, J. ve Zisserman, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. URL: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. Son Erişim Tarihi: 01.05.2019.

74. İnternet: M. Everingham, C. K. I. W., J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. URL: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. Son Erişim Tarihi: 01.05.2019.

75. İnternet: Perona, G. G. a. A. H. a. P. (2007). Caltech-256 Object Category Dataset. California Institute of Technology. URL: https://authors.library.caltech.edu/7694/1/CNS-TR-2007-001.pdf. Son Erişim Tarihi: 02.03.2019.

76. Simonyan, K. ve Zisserman, A. (2014). Very Deep Convolutional Networks for Large-scale Image Recognition. arXiv:1409.1556.

77. Zeiler, M. D. ve Fergus, R. (2014). Visualizing and understanding convolutional networks, *European Conference on Computer Vision*, Zürih, İsviçre, 818-833.

78. L. Fei-Fei, R. F., P. Perona (2007). Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding,* 106(1), 59-70.

79. He, K., Zhang, X., Ren, S. ve Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, *IEEE International Conference on Computer Vision*, Las Condes, Şili, Amerika Birleşik Devletleri, 1026-1034.

80, Girshick, R., Donahue, J., Darrell, T. ve Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, *IEEE conference on computer vision and pattern recognition*, Columbus, Ohio, Amerika Birleşik Devletleri, 580-587.

81. Girshick, R. (2015). Fast R-CNN, *International conference on computer vision* Santiago, Şili, Amerika Birleşik Devletleri, 1440-1448.

82. Ren, S., He, K., Girshick, R. ve Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks, *Advances in Neural Information Processing Systems*, Montréal, Kanada, 91-99.

83. Long, J., Shelhamer, E. ve Darrell, T. (2015). Fully convolutional networks for semantic segmentation, *IEEE conference on computer vision and pattern recognition*, Boston, Massachusetts, Amerika Birleşik Devletleri, 3431-3440.

84. Simonyan, K. ve Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos, *Advances in Neural Information Processing Systems*, Montréal, Kanada, 568-576.

85. Soomro, K., Zamir, A. R. ve Shah, M. (2012). UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild. *arXiv:1212.0412.*

86. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T. ve Serre, T. (2011). HMDB: a large video database for human motion recognition, *International Conference on Computer Vision*, Barselona, İspanya, 2556-2563.

87. Gers, F. A., Schmidhuber, J. ve Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural computation*, 12(10), 2451-2471.

88. Donahue, J., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K. ve Darrell, T. (2017). Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4), 677-691.

89. Xu, K., Ba, J. L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R. S. ve Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention, *International conference on machine learning*, Lille, Fransa, 2048-2057.

90, Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. ve Zitnick, C. L. (2014). Microsoft coco: Common objects in context, *European Conference on Computer Vision*, Zürih, İsviçre, 740-755.

91. Agrawal, A., Lu, J., Antol, S., Mitchell, M., Zitnick, C. L., Batra, D. ve Parikh, D. (2015). VQA: Visual question answering, *IEEE International Conference on Computer Vision*, Şili, Amerika Birleşik Devletleri, 2425-2433.

92. Liang, M. ve Hu, X. (2015). Recurrent convolutional neural network for object recognition, *IEEE conference on computer vision and pattern recognition*, Boston, Massachusetts, Amerika Birleşik Devletleri, 3367-3375.

93. Salakhutdinov, R., Mnih, A. ve Hinton, G. (2007). Restricted boltzmann machines for collaborative filtering, *24th International conference on Machine learning*, Corvallis, Oregon, Amerika Birleşik Devletleri, 791-798.

94. Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood, Computing science and statistics: *Proceedings of 23rd Symposium on the Interface Interface Foundation*, Virginia, Amerika Birleşik Devletleri, 156-163.

95. Hinton, G. E., Osindero, S. ve Teh, Y.-W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computing*, 18(7), 1527-1554.

96. Vincent, P., Larochelle, H., Bengio, Y. ve Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders, *25th International conference on Machine learning*, Helsinki, Finlandiya, 1096-1103.

97. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. ve Manzagol, P. A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11, 3371-3408.

98. Kingma, D. P., Rezende, D. J., Mohamed, S. ve Welling, M. (2014). Semi-supervised learning with deep generative models, *Advances in Neural Information Processing Systems*, Montréal, Kanada, 3581-3589.

99. Bengio, Y., Thibodeau-Laufer, E. ve Alain, G. (2014). Deep generative stochastic networks trainable by backprop, *International Conference on Machine Learning*, Beijing, Çin, 226-234.

100. Bengio, Y., Yao, L., Alain, G. ve Vincent, P. (2013). Generalized denoising auto-encoders as generative models, *Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, Amerika Birleşik Devletleri, 899-907.

101. Im, D. J., Ahn, S., Memisevic, R. ve Bengio, Y. (2017). Denoising criterion for variational auto-encoding framework, *Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, Kaliforniya, Amerika Birleşik Devletleri, 2059-2065.

102. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S. ve Lerchner, A. (2017). B-VAE: Learning Basic Visual Concepts with A Constrained Variational Framework, *International Conference on Learning Representations*, Toulon, Fransa, 6.

103. Germain, M., Gregor, C. K., Murray, I. ve Larochelle, H. (2015). MADE: Masked Autoencoder for Distribution Estimation, *International Conference on Machine Learning*, Lille, Fransa, 881-889.

104. van den Oord, A., Kalchbrenner, N. ve Kavukcuoglu, K. (2016). Pixel recurrent neural networks. *arXiv:1601,06759*.

105. van den Oord, A., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A. ve Kavukcuoglu, K. (2016). Conditional image generation with PixelCNN decoders, *Advances in Neural Information Processing Systems*, Barselona, İspanya, 4790-4798.

106. Salimans, T., Karpathy, A., Chen, X. ve Kingma, D. P. (2017). PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications. *arXiv:1701,05517*.

107. Gulrajani, I., Kumar, K., Ahmed, F., Taiga, A. A., Visin, F., Vazquez, D. ve Courville, A. (2016). PixelVAE: A Latent Variable Model for Natural Images. *arXiv:1611,05013*.

108. Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I. ve Abbeel, P. (2017). Variational lossy autoencoder. *arXiv:1611,02731*.

109. Im, D. J., Kim, C. D., Jiang, H. ve Memisevic, R. (2016). Generating images with recurrent adversarial networks. *arXiv:1602.05110.*

110, Donahue, J., Krähenbühl, P. ve Darrell, T. (2016). Adversarial feature learning. *arXiv:1605.09782.*

111. Chen, M. ve Denoyer, L. (2017). Multi-view generative adversarial networks, *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Skopje, Makedonya, 175-188.

112. Durugkar, I., Gemp, I. ve Mahadevan, S. (2016). Generative Multi-adversarial Networks. *arXiv:1611.01673.*

113. Huang, X., Li, Y., Poursaeed, O., Hopcrof, J. ve Belongie, S. (2017). Stacked generative adversarial networks, *IEEE conference on computer vision and pattern recognition*, Honolulu, Hawaii, Amerika Birleşik Devletleri, 5077-5086.

114. Nowozin, S., Cseke, B. ve Tomioka, R. (2016). f-GAN: Training generative neural samplers using variational divergence minimization, *Advances in Neural Information Processing Systems*, Barselona, İspanya, 271-279.

115. Yu, R., Russell, C., Campbell, N. D. ve Agapito, L. (2015). Direct, dense, and deformable: Template-based non-rigid 3d reconstruction from rgb video, *Proceedings of the IEEE International Conference on Computer Vision*, 918-926.

116. Perriollat, M., Hartley, R. ve Bartoli, A. (2011). Monocular template-based reconstruction of inextensible surfaces. *International journal of computer vision*, 95(2), 124-137.

117. Cheon, S.-U. ve Han, S. (2008). A template-based reconstruction of plane-symmetric 3D models from freehand sketches. *Computer-Aided Design*, 40(9), 975-986.

118. Socher, R., Huval, B., Bhat, B., Manning, C. D. ve Ng, A. Y. (2012). Convolutional-recursive deep learning for 3D object classification, *Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, Amerika Birleşik Devletleri, 656-664.

119. Maturana, D. ve Scherer, S. (2015). VoxNet: A 3D convolutional neural network for real-time object recognition, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Almanya, 922-928.

120, Su, H., Maji, S., Kalogerakis, E. ve Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3D shape recognition, *IEEE International Conference on Computer Vision*, Las Condes, Şili, Amerika Birleşik Devletleri, 945-953.

121. Alexandre, L. A. (2016). 3D object recognition using convolutional neural networks with transfer learning between input channels. *Intelligent Autonomous Systems*, 13 889-898.

122. Riegler, G., Ulusoy, A. O., Bischof, H. ve Geiger, A. (2017). OctNetFusion: Learning depth fusion from data, 2017 International Conference on 3D Vision (3DV), Qingdao, Çin, 57-66.

123. Fan, H., Su, H. ve Guibas, L. (2017). A point set generation network for 3D object reconstruction from a single image, IEEE conference on computer vision and pattern recognition, Honolulu, Hawaii, Amerika Birleşik Devletleri, 605-613.

124. Jiang, L., Shi, S., Qi, X. ve Jia, J. (2018). GAL: Geometric adversarial loss for single-view 3D-object reconstruction, European Conference on Computer Vision (ECCV), Münih, Almanya, 802-816.

125. Yu, L., Li, X., Fu, C.-W., Cohen-Or, D. ve Heng, P.-A. (2018). PU-Net: Point cloud upsampling network, IEEE conference on computer vision and pattern recognition, Salt Lake City, Utah, Amerika Birleşik Devletleri, 2790-2799.

126. Qi, C. R., Su, H., Mo, K. ve Guibas, L. J. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation, IEEE conference on computer vision and pattern recognition, Honolulu, Hawaii, Amerika Birleşik Devletleri, 652-660.

127. Qi, C. R., Yi, L., Su, H. ve Guibas, L. J. (2017). PointNet++: Deep hierarchical feature learning on point sets in a metric space, Advances in Neural Information Processing Systems, Kaliforniya, Amerika Birleşik Devletleri, 5099-5108.

128. Achlioptas, P., Diamanti, O., Mitliagkas, I. ve Guibas, L. (2017). Learning representations and generative models for 3D point clouds. arXiv:1707.02392.

129. Tatarchenko, M., Dosovitskiy, A. ve Brox, T. (2016). Multi-view 3D Models from Single Images with a Convolutional Network, European Conference on Computer Vision, Münih, Almanya, 322-337.

130, Tatarchenko, M., Dosovitski, A. ve Brox, T. (2017). Octree Generating Networks: Efficient convolutional architectures for high-resolution 3D outputs, IEEE International Conference on Computer Vision, Venedik, İtalya, 2088-2096.

131. Riegler, G., Ulusoy, A. O. ve Geiger, A. (2017). OctNet: Learning deep 3D representations at high resolutions, IEEE conference on computer vision and pattern recognition, Honolulu, Hawaii, Amerika Birleşik Devletleri, 3577-3586.

132. Wu, J., Xue, T., Lim, J. J., Tian, Y., Tenenbaum, J. B., Torralba, A. ve Freeman, W. T. (2016). Single image 3D interpreter network, European Conference on Computer Vision, Amsterdam, Hollanda, 365-382.

133. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W. ve Jiang, Y.-G. (2018). Pixel2Mesh: Generating 3D mesh models from single RGB images, European Conference on Computer Vision (ECCV), Münih, Almanya, 52-67.

134. Kato, H., Ushiku, Y. ve Harada, T. (2018). Neural 3d mesh renderer, IEEE conference on computer vision and pattern recognition, Salt Lake City, Utah, Amerika Birleşik Devletleri, 3907-3916.

135. Kanazawa, A., Tulsiani, S., Efros, A. A. ve Malik, J. (2018). Learning category-specific mesh reconstruction from image collections, *European Conference on Computer Vision (ECCV)*, Münih, Almanya, 371-386.

136. Tan, Q., Gao, L., Lai, Y.-K. ve Xia, S. (2018). Variational autoencoders for deforming 3D mesh models, *IEEE conference on computer vision and pattern recognition*, Salt Lake City, Utah, Amerika Birleşik Devletleri, 5841-5850.

137. Pontes, J. K., Kong, C., Sridharan, S., Lucey, S., Eriksson, A. ve Fookes, C. (2018). Image2Mesh: A learning framework for single image 3D reconstruction, *Asian Conference on Computer Vision*, Perth, WA, Avustralya, 365-381.

138. Dai, A., Qi, C. R. ve Nießner, M. (2017). Shape completion using 3D-encoder-predictor CNNs and shape synthesis, *IEEE conference on computer vision and pattern recognition*, Honolulu, Hawaii, Amerika Birleşik Devletleri, 5868-5877.

139. Sinha, A., Unmesh, A., Huang, Q. ve Ramani, K. (2017). SurfNet: Generating 3D shape surfaces using deep residual networks, *IEEE conference on computer vision and pattern recognition*, Honolulu, Hawaii, Amerika Birleşik Devletleri, 6040-6049.

140, Lun, Z., Gadelha, M., Kalogerakis, E., Maji, S. ve Wang, R. (2017). 3D Shape Reconstruction from Sketches via Multi-view Convolutional Networks, *International Conference on 3D Vision (3DV)*, Qingdao, Çin, 67-77.

141. Rezende, D. J., Eslami, S. M. A., Mohamed, S., Battaglia, P., Jaderberg, M. ve Heess, N. (2016). Unsupervised learning of 3D structure from images, *Advances in Neural Information Processing Systems*, Barselona, İspanya, 4996-5004.

142. Shin, D., Fowlkes, C. C. ve Hoiem, D. (2018). Pixels, voxels, and views: A study of shape representations for single view 3D object shape prediction, *IEEE conference on computer vision and pattern recognition*, Salt Lake City, Utah, Amerika Birleşik Devletleri, 3061-3069.

143. Niu, C., Li, J. ve Xu, K. (2018). Im2Struct: Recovering 3D shape structure from a single RGB image, *IEEE conference on computer vision and pattern recognition*, Salt Lake City, Utah, Amerika Birleşik Devletleri, 4521-4529.

144. Smith, E., Fujimoto, S. ve Meger, D. (2018). Multi-View Silhouette and Depth Decomposition for High Resolution 3D Object Representation, *Advances in Neural Information Processing Systems*, Vancouver, Kanada, 6478-6488.

145. Di, X. ve Yu, P. (2017). 3D Reconstruction of simple objects from a single view silhouette image. *arXiv:1701.4752*.

146. Yan, X., Yang, J., Yumer, E., Guo, Y. ve Lee, H. (2016). Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision, *Advances in Neural Information Processing Systems*, Barselona, İspanya, 1696-1704.

147. Wang, M., Wang, L. ve Fang, Y. (2017). 3DensiNet: A robust neural network architecture towards 3D volumetric object prediction from 2D image, *ACM on Multimedia Conference (MM)*, 961-969.

148. Yang, G., Cui, Y., Belongie, S. ve Hariharan, B. (2018). Learning single-view 3D reconstruction with limited pose supervision, *The European Conference on Computer Vision (ECCV)*, Münih, Almanya, 86-101.

149. Sun, Y., Liu, Z., Wang, Y. ve Sarma, S. E. (2018). Im2Avatar: Colorful 3D reconstruction from a single image. *arXiv:1804.06375*.

150, Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J. B. ve Freeman, W. T. (2018). Pix3D: Dataset and methods for single-image 3D shape modeling, *IEEE conference on computer vision and pattern recognition*, Salt Lake City, Utah, Amerika Birleşik Devletleri, 2974-2983.

151. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. ve Xiao, J. (2015). 3D ShapeNets: A Deep Representation for Volumetric Shapes, *IEEE conference on computer vision and pattern recognition*, Boston, Massachusetts, Amerika Birleşik Devletleri, 1912-1920.

152. Internet: Krizhevsky, A. ve Hinton, G. E. (2010). Convolutional deep belief networks on cifar-10.URL: https://www.cs.toronto.edu/~kriz/conv-cifar10-aug2010.pdf. Son Erişim Tarihi: 03.02.2019.

153. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. ve Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The journal of machine learning research*, 15(1), 1929-1958.

154. Goodfellow, I., Bengio, Y. ve Courville, A. (2016). Deep learning, MIT press.

155. İnternet: Nerve Tissue. URL: https://training.seer.cancer.gov/anatomy/nervous/tissue.html. Son Erişim Tarihi: 29.07.2019.

156. İnternet: Perceptron, the building block of modern AI. URL: https://iq.opengenus.org/perceptron-ai/. Son Erişim Tarihi: 29.07.2019.

157. İnternet: Ng, A. (2011). CS294A: Sparse autoencoder lecture notes. URL: https://web.stanford.edu/class/cs294a/sparseAutoencoder_2011new.pdf. Son Erişim Tarihi: 02.08.2019.

158. Sietsma, J. ve Dow, R. J. F. (1991). Creating artificial neural networks that generalize. *Neural Networks*, 4(1), 67-79.

159. Tang, Y. ve Eliasmith, C. (2010). Deep networks for robust visual recognition, 27th International Conference on Machine Learning (ICML) Haifa, Israil, 1055-1062.

160, İnternet: Huang, T. Convolutional neural network, CNN. URL: medium.com/@chih.sheng.huang821/convolutional-neural-network-cnn. Son Erişim Tarihi: 03.08.2019.

161. İnternet: Convolution. URL: https://en.wikipedia.org/wiki/Convolution. Son Erişim Tarihi: 01.09.2019.

162. İnternet: Gwardys, G. Convolutional Neural Networks backpropagation: from intuition to derivation. URL: https://grzegorzgwardys.wordpress.com/2016/04/22/8/ Son Erişim Tarihi: 07.09.2019.

163. Collins, R. T. (1996). A Space-Sweep Approach to True Multi-Image Matching, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, Kaliforniya, Amerika Birleşik Devletleri, 358-363.

164. Maas, A. L., Hannun, A. Y. ve Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models, *International Conference on Machine Learning*, Atlanta, Georgia, Amerika Birleşik Devletleri, 3.

165. Ren, H., Chen, D. ve Wang, Y. (2018). RAN4IQA: Restorative adversarial nets for no-reference image quality assessment, *Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, Amerika Birleşik Devletleri, 7308-7314.

166. Hausman, K., Chebotar, Y., Schaal, S., Sukhatme, G. ve Lim, J. J. (2017). Multi-Modal Imitation Learning from Unstructured Demonstrations using Generative Adversarial Nets, *Advances in Neural Information Processing Systems*, Kaliforniya, Amerika Birleşik Devletleri, 1235-1245.

167. Vukotić, V., Raymond, C. ve Gravier, G. (2017). Generative adversarial networks for multimodal representation learning in video hyperlinking, *ACM on International Conference on Multimedia Retrieval (ICMR)*, Bükreş, Romanya, 416-419.

168. Nooruddin, F. S. ve Turk, G. (2013). Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics*, 9(2), 191-205.

169. İnternet: Min, P. Binvox 3D mesh voxelizer. URL: http://www.patrickmin.com/binvox. Son Erişim Tarihi: 01.10.2019.

170, Kingma, D. P. ve Ba, J. L. (2014). ADAM: A method for stochastic optimization. *arXiv:1412.6980.*

171. Kohl, S. A. A., Romera-Paredes, B., Meyer, C., Fauw, J. D., Ledsam, J. R., Maier-Hein, K. H., Eslami, S. M. A., Rezende, D. J. ve Ronneberger, O. (2018). A probabilistic U-Net for segmentation of ambiguous images, *Advances in Neural Information Processing Systems*, Montréal, Kanada, 6965-6975.

172. Pontes, J. K., Kong, C., Sridharan, S., Lucey, S., Eriksson, A. ve Fookes, C. (2018). H-DenseUNet: Hybrid densely connected UNet for liver and tumor segmentation from CT volumes. *IEEE Transaction on Medical Imaging*, 37(12), 2663-2674.

173. Yamanaka, J., Kuwashima, S. ve Kurita, T. (2017). Fast and accurate image super resolution by deep CNN with skip connection and network in network, *International Conference on Neural Information Processing*, Guangzhou, Çin, 217-225.

174. Tong, T., Li, G., Liu, X. ve Gao, Q. (2017). Image super-resolution using dense skip connections, *IEEE International Conference on Computer Vision*, Venedik, İtalya, 4799-4807.

175. Mao, X.-J., Shen, C. ve Yang, Y.-B. (2016). Image restoration using convolutional auto-encoders with symmetric skip connections. *arXiv:1606.08921*.

176. Kodali, N., Abernethy, J., Hays, J. ve Kira, Z. (2017). On convergence and stability of gans. *arXiv:1705.07215.*

177. İnternet: Despois, J. Latent space visualization — Deep Learning bits #2. URL: https://medium.com/hackernoon/latent-space-visualization-deep-learning-bits-2-bd09a46920df. Son Erişim Tarihi: 05.10.2019

178. Rahman, M. A. ve Wang, Y. (2016). Optimizing intersection-over-union in deep neural networks for image segmentation, *International symposium on visual computing*, Lake Tahoe, Nevada, Amerika Birleşik Devletleri, 234-244.

179. Berman, M., Triki, A. R. ve Blaschko, M. B. (2018). The Lovász-Softmax loss A tractable surrogate for the optimization of the intersection-over-union measure in neural networks, *IEEE conference on computer vision and pattern recognition*, Salt Lake City, Utah, Amerika Birleşik Devletleri, 4413-4421.

180, Yang, B., Rosa, S., Markham, A., Trigoni, N. ve Wen, H. (2017). 3D object dense reconstruction from a single depth view, *IEEE International Conference on Computer Vision*, Venedik, İtalya, 679-688.

181. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. ve Courville, A. (2017). Improved Training of Wasserstein GANs, *Advances in Neural Information Processing Systems*, Kaliforniya, Amerika Birleşik Devletleri, 5767-5777.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : Güzel Turhan, Ceren
Uyruğu : T.C.
Doğum tarihi ve yeri : 11,04.1988, Ankara
Medeni hali : Evli
Telefon : 0 554 403 90 34
e-mail : cerenguzel@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Doktora	Gazi Üniversitesi / Bilgisayar Mühendisliği	Devam ediyor
Yüksek lisans	Gazi Üniversitesi / Bilgisayar Mühendisliği	2014
Lisans	Çankaya Üniversitesi / Bilgisayar Mühendisliği	2011
Lise	Ömer Seyfettin Lisesi	2005

İş Deneyimi

Yıl	Yer	Görev
2011-Halen	Gazi Üniversitesi	Araştırma Görevlisi

Yabancı Dil

İngilizce

Yayınlar

1. Turhan, C. G. ve Bilge, H. S. (Baskıda). Fused voxel autoencoder for single image to 3D object reconstruction. *Electronics Letters.*

2. Turhan, C. G. ve Bilge, H. S. (2020). Scalable image generation and super resolution using generative adversarial networks. *Journal of the Faculty of Engineering and Architecture of Gazi University,* 35(2), 953-966.

3. Turhan, C. G. ve Bilge, H. S. (2019). Single Image to 3D reconstruction Using Generative Networks. 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Türkiye, 24-26 Nisan.

4. Turhan, C. G. ve Bilge, H. S. (2018). Recent Trends in Deep Generative Models: a Review. 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, Bosna, 20-23 Eylül.

5. Turhan, C. G. ve Bilge, H. S. (2018). Variational Autoencoded Compositional Pattern Generative Adversarial Network for Handwritten Super Resolution Image Generation. 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, Bosna, 20-23 Eylül.

6. Turhan, C. G. and Bilge, H. S. (2018). Single image super resolution using deep convolutional generative neural networks. 26th Signal Processing and Communications Applications Conference (SIU), İzmir, Türkiye, 2-5 Mayıs.

7. Guzel Turhan, C. and Bilge, H. S. (2017). Novel Class-wise Two-dimensional Principal Component Analysis Method for Face Recognition. *IET Computer Vision*, 11(4), 286-300,

8. Turhan, C. G. ve Bilge, H. Ş. (2017). Generating word images using deep generative adversarial networks. 25th Signal Processing and Communications Applications Conference (SIU), Antalya, Türkiye, 15-18 Mayıs.

9. Güzel, C. ve Bilge, H. S. (2015). Analysis for feature extraction on block based spatial domain. *Global Journal on Technology*, 7, 65-70,

10. Turhan, C. G., Yildirim Okay, F., Yildiz, O. ve Bilge, H. S. (2014). Dimensionality reduction with PCA in block based DCT domain. International Conference on Advanced Technology & Sciences (ICAT), Antalya, Turkey, 12-15 Ağustos.

11. Turhan, C. G. ve Bilge, H. S. (2014). kNN algorithm based on axis characteristic on Lorentzian space. 22nd Signal Processing and Communications Applications Conference (SIU), Trabzon, Türkiye, 23-25 Nisan.

12. Bilge, H. S. ve Guzel, C. (2013). Face recognition on Lorentzian manifold. 21st Signal Processing and Communications Applications Conference (SIU), Haspolat, Kıbrıs, 24-26 Nisan.

13. Güzel, C. Kaya, M., Yıldız, O. ve Bilge, H. S. (2013). Breast Cancer Diagnosis Based on Naïve Bayes Machine Learning Classifier with KNN Missing Data Imputation, AWER Procedia Information Technology & Computer Science, Antalya, Türkiye, 26-27 Nisan.

Hobiler

Film izlemek, seyahat etmek, pilates.

GAZİ GELECEKTİR..