

Akıllı Sistemlerin Endüstriyel Uygulaması II

100
TÜRKİYE CUMHURİYETİNİN YÜZÜNCÜ YILI

Editor
Eyyüp GÜLBANDILAR



BİDGE Yayınları

Akıllı Sistemlerin Endüstriyel Uygulaması II

Editör: Prof. Dr. Eyyüp GÜLBANDILAR

ISBN: 978-625-6707-14-6

1. Baskı

Sayfa Düzeni: Gözde YÜCEL

Yayınlama Tarihi: 25.12.2023

BİDGE Yayınları

Bu eserin bütün hakları saklıdır. Kaynak gösterilerek tanıtım için yapılacak kısa alıntılar dışında yayıncının ve editörün yazılı izni olmaksızın hiçbir yolla çoğaltılamaz.

Sertifika No: 71374

Yayın hakları © BİDGE Yayınları

www.bidgeyayinlari.com.tr - bidgeyayinlari@gmail.com

Krc Bilişim Ticaret ve Organizasyon Ltd. Şti.

Güzeltepe Mahallesi Abidin Daver Sokak Sefer Apartmanı No: 7/9 Çankaya /
Ankara



İÇİNDEKİLER

İÇİNDEKİLER	3
Text-to-Speech (TTS) Teknolojisi: Gelişimi, Uygulamaları ve Geleceği	5
Funda AKAR.....	5
Bayram GÜNGÖR	5
Kubernetes ile Küme ve Dağıtım Altyapısı Oluşturma, Dağıtım Güncellemelerini Yönetme ve Servislerin Düğümler Üzerinde Çalıştırılması: Deneysel Uygulamalar	39
Işıl KARABEY AKSAKALLI.....	39
Levent BAYINDIR	39
Gradyan Temelli Kenar Belirleme Yaklaşımları ile Görüntü Enterpolasyonu.....	61
Mahmut KILIÇASLAN	61

B-GAN: Biyomedikal Bilişim Uygulamalarında Yapay Zekanın Üretken Yüzü	80
İsa ATAŞ.....	80
Çelik Boru Kaynaklarında Oluşan Hataların Tespitinde Yeni Yöntemler.....	109
Murat MERAL	109
Kamil ORMAN	109
Kaynak Hata Türleri.....	112
Cisco Packet Tracer ile Ağ Tasarımı ve Simülasyonu.....	122
Muammer AKÇAY	122
Timorshah RAHİMİ	122
Kameraların Kısıtlı Alan Derinliğini Artırmak ve İyileştirilmiş Gece Görüşü Sağlamak için Görüntü Birleştirme Yöntemlerinin Karşılaştırılması	129
Fatma Çakıroğlu	129
Rifat Kurban *	129
Ali Durmuş.....	129
Ercan Karaköse.....	129

BÖLÜM I

Text-to-Speech (TTS) Teknolojisi: Gelişimi, Uygulamaları ve Geleceği

Funda AKAR¹
Bayram GÜNGÖR²

1. Giriş

Günümüzde, sesli iletişim teknolojileri hayatımızın hemen hemen her alanında giderek daha fazla önem kazanıyor. Akıllı cihazlar, dijital asistanlar ve daha birçok platform, metin tabanlı bilgileri sesli bir şekilde sunabilmek için Text to Speech (TTS) (Metinden Konuşma) teknolojisine dayanıyor. TTS, metin tabanlı bilgileri otomatik olarak sesli hale getiren bir dönüşüm sürecini ifade eder. Bu teknoloji, bilgisayarların yazılı metni insan benzeri seslere dönüştürmesini sağlar. TTS, sadece metinleri duyulabilir hale getirmekle kalmaz, aynı zamanda metni anlamakta zorlanabilecek

¹ Dr. Öğr. Üyesi, Erzincan Binalı Yıldırım Üniversitesi, Mühendislik-Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, fakar@erzincan.edu.tr

² Lisans Öğrenci, Erzincan Binalı Yıldırım Üniversitesi, Mühendislik-Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, byrmngng@gmail.com

kişilere yardımcı olur. Özellikle görme engelliler, metin tabanlı içeriklere sesli bir şekilde erişerek bilgilere ulaşabilirler. Bu teknoloji ayrıca metin tabanlı içerikleri daha erişilebilir ve anlaşılabilir hale getirerek geniş bir uygulama yelpazesine sahiptir. Günümüzde TTS teknolojisinin kaynaklarının yetersiz olması ve Türkçe kaynakların sınırlı olmasından dolayı çalışmamız bu teknolojinin tanıtılması için büyük önem taşımaktadır. Bu teknoloji öğrenmek isteyen kişilere de ilham kaynağı olacaktır.

TTS, dil sınırlarını aşmamıza ve kültürel çeşitliliği vurgulamamıza olanak tanır. Eğitim, erişilebilirlik ve dijital içerik oluşturma alanlarında TTS, bilgiye erişimi ve iletişimi dönüştürmek için güçlü bir araç olarak kabul edilir. Ayrıca, iş dünyası ve endüstriler için de büyük bir potansiyel sunar; sesli reklamlar, sesli kitaplar ve interaktif sanal asistanlar gibi alanlar, işletmelerin ve girişimlerin müşterileriyle daha yakın bir iletişim kurmalarını sağlar. Teknolojinin hızla evrildiği bir çağda, iletişim ve erişilebilirlik alanında ses teknolojileri, önemli bir değişimi tetikliyor. Bu değişimin önemli bir parçası olan TTS, metinden sese dönüşüm konusundaki ilerlemeleri ve uygulamalarıyla insanlar için yeni olanaklar sunuyor. Bu çalışma, TTS teknolojisinin geçmişi, çalışma prensipleri, sentezleme yöntemleri ve gelecekteki potansiyeli detaylı olarak ele alınacaktır.

TTS'nin evrimi, ilk mekanik dönemlerden elektronik süreçlere ve modern yaklaşımlara kadar olan yolculuğunda önemli bir gelişim gösterdi. Bu evrim, sentezleme yöntemlerindeki değişimlerle birlikte, metin analizinden ses sentezine kadar geniş bir yelpazede yeni olanaklar doğurdu. Bu çalışmada TTS teknolojisinin tarihsel gelişimi, sentezleme yöntemleri ve temel işleyişi adım adım incelenecek. Ayrıca, TTS teknolojisinin günümüzdeki uygulamaları da detaylı olarak ele alınacak. Eğitim materyallerinden sesli asistanlara ve sesli kitaplara kadar geniş bir alanda TTS'nin kullanımı, bu teknolojinin insan yaşamına olan etkisini göstermesi açısından önemlidir. Son olarak, bu çalışma TTS teknolojisinin geleceğine odaklanacak. Gelecekteki eğilimler, olası gelişmeler ve bu teknolojinin ilerleyen dönemdeki potansiyeli, iletişim ve

teknoloji dünyasındaki yeri hakkında bir bakış sunacaktır. Bu çalışma, TTS teknolojisinin geniş bir perspektiften ele alınarak, okuyuculara sesin metinle birleştiği bu etkileyici teknoloji dünyasının kapılarını aralamayı amaçlamaktadır.

2. Terimler ve Kısaltmaları

Formant : Formantlar insan sesinin karakteristik özelliklerini oluşturan ses bileşenleridir. İnsanlar konuşurken, seslerin belirli frekansta yoğunlaştığı noktalara formant denir. Formantlar, sesin nasıl duyulduğunu ve anlaşıldığını etkiler. Örneğin, farklı vokal sesleri farklı formant yapılarına sahiptir. Bir vokal sesin yüksekliği ve şekli, formantlar tarafından belirlenir. Aynı şekilde, ünsüz seslerin farklılıkları da formantlar aracılığıyla oluşturulur. Formantlar, insan sesinin anlaşılabilirliğini sağlayan önemli bileşenlerdir.

Vokal Ses : Vokal ses, ses üretirken ses yolunun tıkanmadığı ve ses tellerinin titreşimine izin verdiği bir ses türüdür. Vokal sesler genellikle ünlü sesler olarak bilinir ve konuşma veya müziğin temel öğelerinden biridir. İnsan sesini oluşturan vokal sesler, ses tellerinin titreşimi ve ağız boşluğunun şeklinin değiştirilmesi ile oluşturulmaktadır.

Rezonans : rezonans, bir sistemin veya nesnenin titreşimlerin veya dalgaların etkisi altında özel bir frekansta tepki verme durumunu ifade eder.

Rezanotör : Rezonatör, bir nesnenin veya sistemin titreşimlerini veya ses dalgalarını belirli bir şekilde etkileyen bir yapı veya bileşen olarak düşünülebilir.

Fonotik : Fonotik, konuşma seslerini inceleyen bir dil bilimi dalıdır. Fonotik dalı, konuşma seslerinin fiziksel özelliklerini (ses titreşimleri, ağız ve boğaz anatomisi gibi) ve nasıl üretildiğini inceler. Ayrıca, farklı seslerin nasıl duyulduğunu ve insanlar arasında nasıl farklılaştığını araştırır.

Akustik : Akustik sesin üretimi, yayılması ve algılanmasıyla ilgilenen bilim dalıdır. Ayrıca sesin fiziksel özellikleri, titreşimleri, yankıları ve ses dalgalarının yayılma şekli gibi konuları da içerir. Akustik sesin nasıl oluştuğunu, nasıl taşındığını ve insanlar veya diğer canlılar tarafından nasıl algılandığını incelemekle ilgilenir.

TTS’de kullanılan terimlerin kısaltmaları ve açıklamaları Tablo 1’de verilmiştir.

Tablo 1. *TTS’de Kullanılan Terimlerin Kısaltmalarının Açılımı*

Kısaltma	Açılımı
TTS	Text To Speech (Yazıdan Konuşmaya Çevirme)
STT	Speech To Text (Konuşmadan Yazıya Çevirme)
PAT	Parametric Artificial Talker (Parametrik Yapay Konuşmacı)
DAVO	Dynamic Analog of the VOcal tract (VOcal kanalının Dinamik Analogu)
HMM	Hidden Markov Model (Gizli Markov Model)
DNN	Deep Neural Network (Derin Sinir Ağı)
SPSS	Statistical Parametric Speech Synthesis (İstatiksel Parametrik Konuşma Sentezleme)
RNN	Recurrent Neural Network (Tekrarlayan Sinir Ağı)
LSTM	Long Short-Term Memory (Uzun Kısa Süreli Bellek)
NPL	Naturel Language Processing (Doğal Dil İşleme)

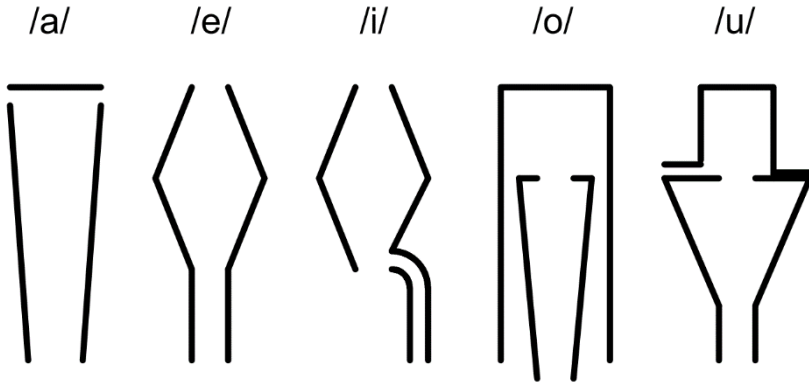
3. TTS Teknolojisinin Tarihsel Gelişimi

Günümüzde neredeyse ayırt edilemeyecek seviyede yapay konuşma üretebilen TTS sistemlerinin tarihi 18. yüzyıla dayanmaktadır. Mekanik cihazlarla başlayan evrim süreci günümüzde yapay zekâ kullanılarak devam etmektedir. Bu süreç, farklı dönemlerdeki teknolojik ilerlemeler, yazılım gelişmeleri ve

kullanım alanlarının genişlemesiyle şekillenerek oluşmuştur. TTS teknolojisinin tarihsel gelişimini dönemlere ayırmak gerekirse;

3.1 Mekanik Dönemler

TTS teknolojisinin ilk adımlarının atıldığı mekanik dönemde, konuşma beceresi yaratıcının sağladığı bir yetenek mi yoksa insanlar tarafından geliştirilen bir beceri mi olduğu konusunda tartışmalar sürmekteydi (Brackhane, 2015). St. Petersburg Bilimler Akademisi 1780 yılında sesli harflerin doğası, farkları ve seslerin yapay olarak nasıl üretilebileceği konusunda ödüllü bir bilimsel yarışma düzenlemeye karar verir. Profesör Christian Gottlieb Kratzenstein bu konuda çalışmalar yürüttüğünü öne sürerek yarışmaya katıldı (Brackhane, 2015; Ohala, 2011). C. Kratzenstein İngiliz alfabesindeki beş ünlü harfinde ('a', 'e', 'i', 'o', 'u') fizyolojik farklarını açıkladı (Şekil 1) ve nefesli enstrümanlar gibi sesleri insan seslerine benzeten mekanik bir makine geliştirdi (Brackhane, 2015; Ohala, 2011; Sami Lemmetty, 1999).

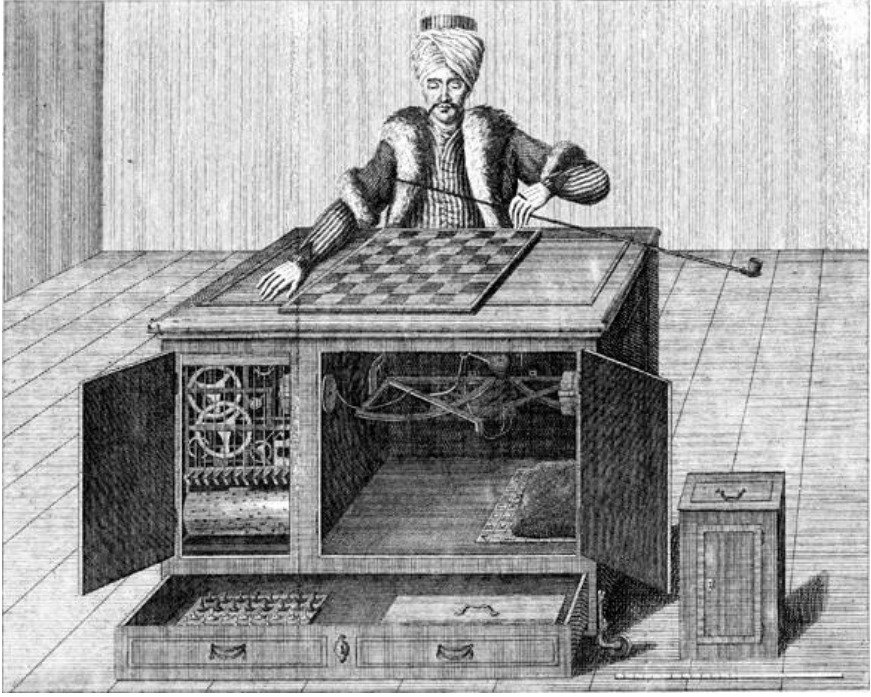


Şekil 1. Kratzenstein'in sesli harf rezonatörleri (Sami Lemmetty, 1999)

Kratzenstein'in buluşundan hemen ardından 1791 yılında Viyana'da Wolfgang von Kempelen adındaki bir mucit "Akustik-Mekanik Konuşma Makinesi" adındaki icadını tanıttı. Bu makine

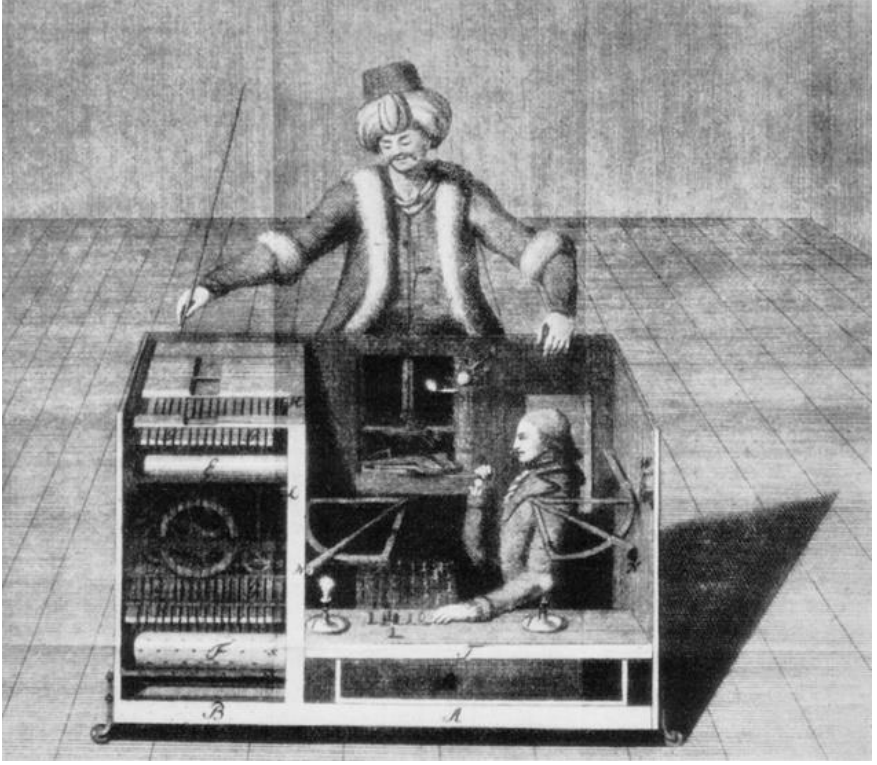
bazı tek sesleri ve birkaç harften oluşan kelimeleri üretebiliyordu (Ohala, 2011; Sami Lemmetty, 1999). Kempelen Kratzenstein'den farklı olarak dil ve dudak modellerini de makinesine ekleyerek makinesini insan anatomisine daha uygun hale getirmiştir (Wikipedia, n.d.-a). Ancak Kempelen'in 1783 yılında Londra'dan Avrupa'ya yolculuk yaptığı sırada sergilediği “Satranç Oynayan Türk” adındaki otomat nedeniyle kötü şöhretle anıldığı için TTS teknolojisinin temelini oluşturan bu önemli buluş döneminde fazla ilgi görmemiştir (Sami Lemmetty, 1999).

Kempelen'in “Satranç Oynayan Türk” adını verdiği otomatı 6 ayda geliştirerek 1770'te ilk kez İmparatoriçe Maria Theresa için sergilenmiştir (Brackhane, 2015). Üzerine satranç tahtası çizilen küçük, tekerlekli bir kabin bulunmaktadır. Kabinin yanında oturan bıyıklı, sarıklı ve pelerinli mekanik parçalardan oluşan Türk figürü oturmaktadır (Şekil 2). Gösterici gösteriye başlamadan önce makinenin ve figürün iç bölmelerindeki makara, kaldıraç gibi karmaşık mekanik malzemeleri göstererek seyircilerde akıllı cihaz izlenimi oluşturunuyordu (Brackhane, 2015; Mark J. Sussman, 1999; Wikipedia, 2023). Figür rakibiyle satranç oynarken satranç tahtasını inceliyor, ara sıra başını sallayarak taşların yerini değiştiriyor ve maç bitiminde seyircilerden gelen soruları tepsideki harfleri birleştirerek yanıtlatabiliyordu. Rakiplerinin çoğunu yenen bu otomat Napolyon Bonaparte, Benjamin Franklin gibi önemli isimlerle de satranç oynamıştır. Napolyon'unda dahil olduğu 52 adet oyunun detayları The Turk, Chess Automaton (Gerald Levitt) isimli kitapta bulunmaktadır (Moran, 2007; Wikipedia, 2023).



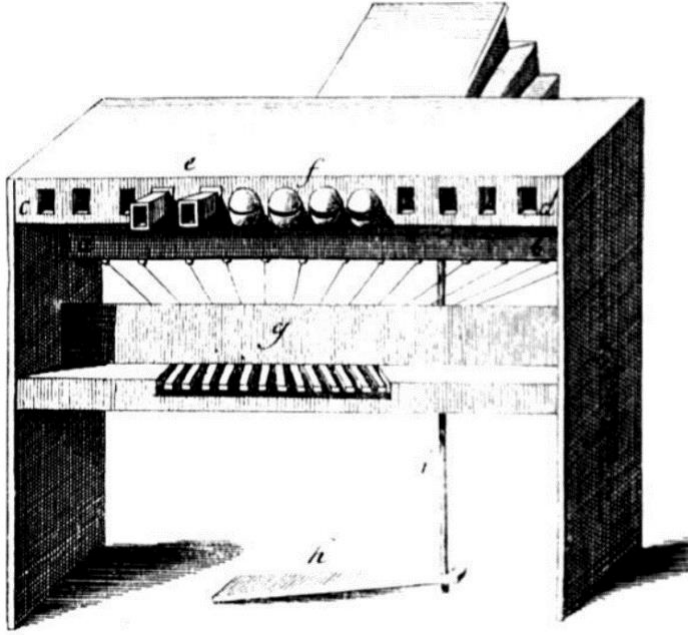
Şekil 2. “Satranç Oynayan Türk” Otomatının görüntüsü
(Wikipedia, 2023)

“Satranç Oynayan Türk” otomatı döneminde büyük ilgi uyandırarak pek çok yazıya konu olsa da aslında içerisine bir insanın saklandığı büyük bir aldatmacaydı (Şekil 3) (Moran, 2007). Mekanizmanın içerisindeki bacakları olmayan satranç ustası, mıknaatıslar ve mum yardımıyla oyunu takip edebilmiş ayrıca mekanik kollar sayesinde de figürü yönetmiştir (Mark J. Sussman, 1999; Moran, 2007; Wikipedia, 2023). Kempelen her ne kadar bu cihazın akıllı olduğu iddia etmese de çalışma prensibini hakkında hiçbir yorum yapmamasından dolayı kötü şöhretle anılmıştır (Brackhane, 2015). Bu sebeple “Akustik-Mekanik Konuşma Makinesi” adındaki gerçek ve önemli olan icadı yeterince ilgi görmemiştir.

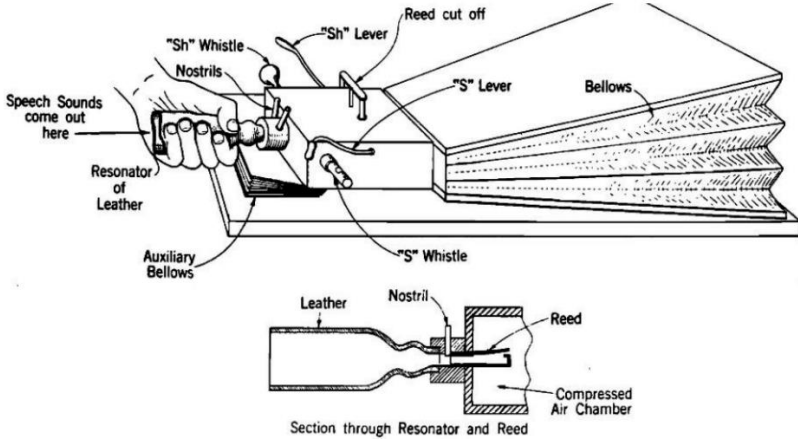


Şekil 3. “Satranç Oynayan Türk” Otomatının gizli bölmeli görüntüsü (Mark J. Sussman, 1999)

1837’de Charles Wheatstone adındaki bir bilim adamı, gelişen teknolojinin de katkılarıyla Kempelen’in konuşan makinesinin (Şekil 4) çoğu harfı çıkarabilen bir versiyonunu üretti (Şekil 5). Wheatstone'un geliştirdiği konuşma makinesinin sergilendiği bir etkinlik sonrasında genç Alexander Graham Bell, babasının da rehberliği ve teşvikiyle kendi konuşma cihazını oluşturmaya karar verdi. Bell'in bu deneyleri ve araştırmaları sonucunda 1876'de telefonun icadı gerçekleşti ve bu, dünya çapındaki iletişim yöntemlerini kökten değiştirdi (Sami Lemmetty, 1999; Takayuki Arai, 2013).



Şekil 4. Kempelen'in ses sentezi makinesi (Brackhane, 2015)

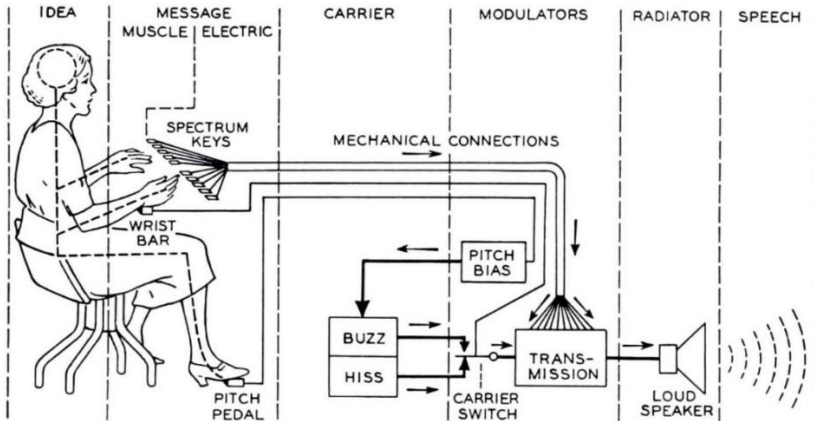


Şekil 5. Charles Wheatstone'ın konuşan makinesi (Takayuki Arai, 2013)

3.2 Elektronik Dönemi

1922'de, Stewart tarafından yapılan elektrikli bir cihaz, ilk defa konuşma sesleri üretmek için kullanıldı. Bu cihaz, konuşma sesinin özelliklerini taklit etmek için zil benzeri bir uyarıcı ve ses yolunu modellemek için iki rezonatörden oluşan devreleri içeriyordu. Ancak bu cihaz sadece ünlü sesler üretebiliyor ünsüz sesleri ve kelimeleri üretemiyordu. Aynı türden bir sentezleyici daha sonra Wagner tarafından da yapıldı. Wagner'in cihazı Stewart'ın cihazından farklı olarak dört elektrikli rezonatörden oluşuyordu (Sami Lemmetty, 1999).

1939 yılında TTS teknoloji için önemli bir gelişme yaşandı. Bu gelişme, New York Dünya Fuarı'nda Bell Laboratuvarı tarafından tanıtılan VODER adlı bir cihazla ilk elektronik konuşma sentezi yapılarak gerçekleşti. VODER, sesi analiz eden VOCODER adlı başka bir cihazdan esinlenerek yapılmıştı. Cihaz bir piyano gibi görünüyordu (Şekil 6) sesin frekansını ayarlamak için bir ayak pedalı ve harf, kelime ve kısa ifadeleri oluşturabilen tuşlara sahipti. Ancak, bu cihazı kullanabilmek için eğitilmiş bir uzman gerekiyordu (Akhmetzhanov, 2021). VODER cihazının sergisi, elektronik bir cihaz ile konuşma üretmenin mümkün olduğunu gösterdi ve konuşma sentezi konusundaki ilgiyi artırdı (Chettri & Shah, 2013).



Şekil 6. Voder Kullanım Şeması (Wikipedia, n.d.-b)

1953 yılında Walter Lawrence'ın geliştirdiği "Parametric Artificial Talker" (PAT) adını verdiği bir cihaz, sesi bir elektrik devresi kullanarak yapay bir şekilde oluşturmak için tasarlandı. Aynı dönemde Gunnar Fant, "OVE I" adını taşıyan kendi cihazını geliştirdi. Daha sonra Fant, "OVE II" adlı daha gelişmiş bir cihazı üretti. Bu cihazlar, sesin belirli özelliklerini doğrudan kontrol ederek, formant sentezleme olarak bilinen yapay konuşma yöntemini başlattılar (Sami Lemmetty, 1999; Story, 2019).

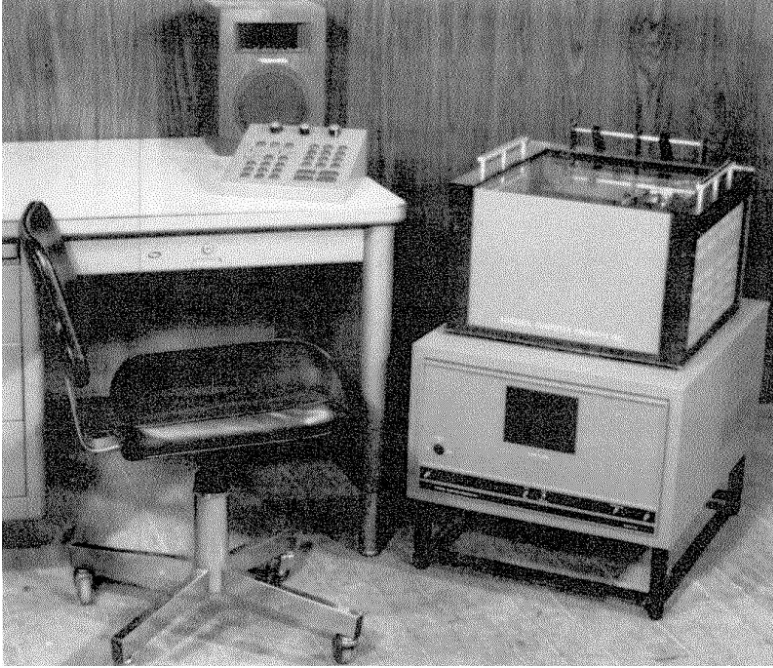
1958 yılında Massachusetts Teknoloji Enstitüsü'nde (M.I.T) George Rosen, DAVO adındaki cihaz ile ilk artikülasyon sentezlemeyi geliştirdi (Sami Lemmetty, 1999).

1968 yılında TTS sistemi İngilizce için tam olarak ilk kez Japonya Elektroteknik Laboratuvarı'nda Noriko Umeda ve arkadaşları tarafından geliştirildi. Bu sistem, bir artikülasyon modeline dayanıyordu ve karmaşık kurallara sahip bir söz dizimi analiz modülünü içeriyordu. Konuşma oldukça anlaşılır olsa da monotondur ve günümüz sistemlerinin kalitesinden oldukça uzaktı (Sami Lemmetty, 1999).

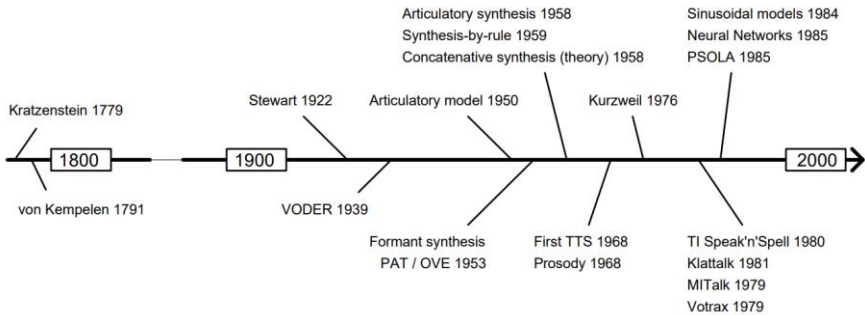
1976 yılında görme engelliler için oluşturulan fuarda Kurzweil'in okuma makinesi tanıtıldı. Makine yaygın kullanılan yazı tipleri ve puntoları fotokopi makinelerinde olduğu gibi optik tarayıcı ile tarayarak seslendiriyordu (Şekil 7). 1000'den fazla fonetik kural kullanan cihaz metinleri ortalama olarak dakikada standart konuşma hızı olan 150 kelime civarında seslendirebiliyordu (Kleiner & Kurzweil, 1977).

Kurzweil'in okuma makinesinden sonra, 1970'lerin sonlarından modern döneme kadar birçok sentez türü ve ticari TTS ürünü piyasaya çıkmıştır. Özellikle bilgisayarlarda yaşanan gelişmeler bu teknolojinin ilerlemesine büyük katkıda bulunmuştur. Bilgisayarların işlem kapasiteleri arttıkça, daha karmaşık ses sentezi algoritmaları geliştirilebilmiş ve metinler daha iyi bir şekilde seslendirilebilir hale gelmiştir (Story, 2019). Şekil 8'de bulunan şemada 2000'li yıllara kadar bazı önemli gelişmelerin ve ürünlerin tarihsel şeması gösterilmiştir. Geliştirilen ürünler özellikle

engellilerin metin tabanlı içeriklere erişimi kolaylaştırmıştır. Bu dönemdeki gelişmeler, gelecekteki TTS teknolojisinin daha karmaşık ve çok yönlü hale gelmesinin kapısını açarak modern TTS teknolojisinin temelini atmıştır.



Şekil 7. Kurzweil'in Okuma Makinesi (Kleiner & Kurzweil, 1977)



Şekil 8. TTS teknolojisinde önemli gelişmelerin tarihsel şeması.
(Sami Lemmetty, 1999)

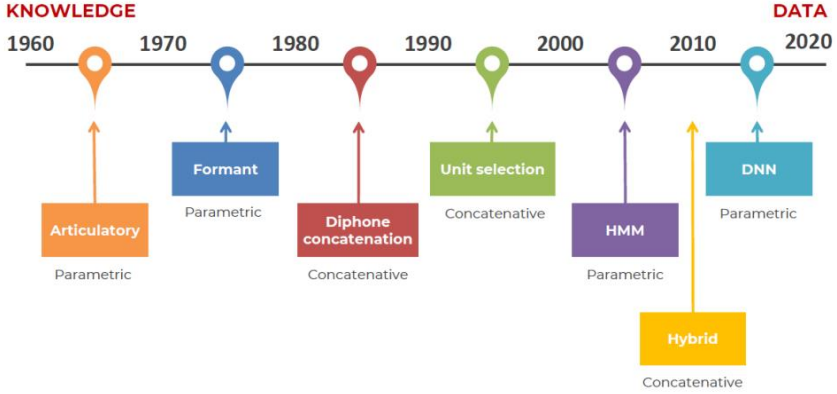
3.3 Modern Dönem

TTS teknolojisi, hızla gelişen bilgisayar gücü, doğal dil işleme, yapay zekâ ve makine öğrenme tekniklerinin bir araya gelmesiyle büyük bir ivme kazandı. Bu dönemde birçok önemli gelişme ve değişim gözlemlendi. Bu dönemin öne çıkan özelliklerinden biri, ses sentezi kalitesinin kayda değer bir şekilde artmış olmasıdır. Daha önce metinleri seslendiren TTS sistemleri genellikle mekanik ve yapay sesler üretirken, modern TTS teknolojisi sayesinde metinler daha doğal ve insan gibi seslendirilebilmektedir. İnsan konuşmasını taklit eden sesler, vurgu, tonlama ve duygu ifadelerini daha iyi yakalayabiliyor olması metin tabanlı içeriklerin sesli olarak sunulmasının daha kolay dinlenir ve anlaşılır olmasını sağladı. Şekil 9'da TTS teknolojisinin modern döneme kadar olan gelişimi gösterilmektedir.

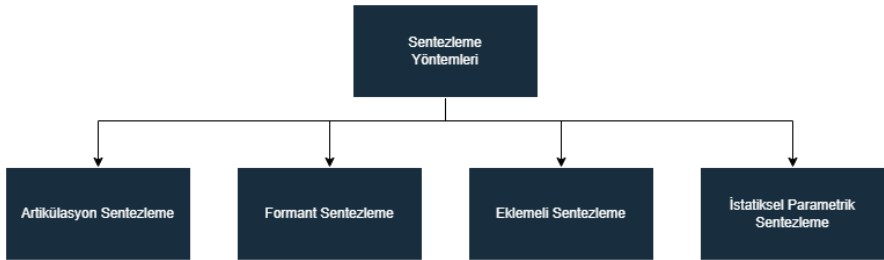
4. Sentezleme Yöntemleri

Yapay konuşma üretimi için birçok sentezleme yöntemi bulunmaktadır. Dünya üzerindeki bütün dillerde işe yarayacak evrensel bir sentezleme türü bulunmadığı için ve konuşma kalitesinin yükseltilmesi amaçlandığından farklı yöntemler ortaya çıkmıştır (Panda et al., 2020). Bu çalışmada temel sentezleme yöntemlerinin işleyişi, avantajları ve dezavantajları anlatılmaktadır. Sentezleme yöntemleri 4 temel sınıfa ayrılır (Şekil 10).

- Artikülasyon Sentezleme
- Formant Sentezleme
- Eklemeli Sentezleme
 - Birim Seçme Sentezleme
 - İkili Ses Sentezleme
 - Uygulamaya Özgü Sentezleme
 - Hece Tabanlı Sentezleme
- İstatiksel Parametrik Sentezleme
 - Hidden Markov Model Sentezleme
 - Derin Sinir Ağı Sentezleme



Şekil 9. TTS Modern Dönem Gelişmeleri (Manero Alvarez Advisors et al., 2022)

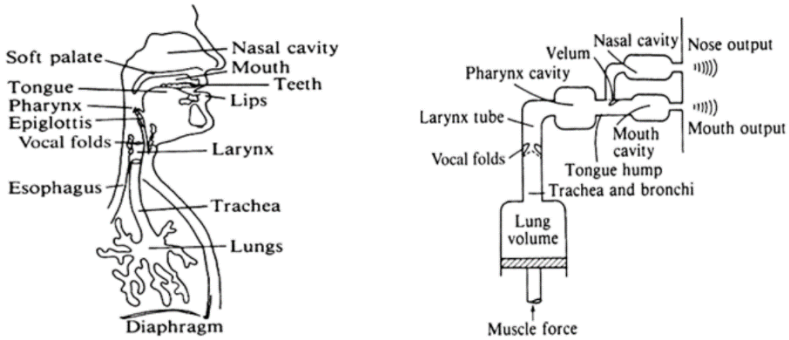


Şekil 10. TTS Temel Sentezleme Yöntemleri

4.1 Artikülasyon Sentezleme

Artikülasyon sentezleme yönteminde insanların doğal konuşma üretim sürecini modeller ve bu süreçte hava akışı dijital olarak simüle edilerek yapay konuşma üretmeyi amaçlamaktadır. Artikülasyon sentezleme ile üretilen seslerde anlaşılabilirlik oranı yüksektir fakat artikülerin hassas bir şekilde modellenmesi zordur bu sebeple konuşma kalitesi genellikle diğer sentez türlerine göre daha düşüktür (Guangpu Huang & Meng Joo Er, 2012; Kaur & Singh, 2023). Bu yöntemle ile üretilen yapay konuşmalar kalite olarak diğer sentez yöntemlerinden düşük kalitede olsa da konuşma üretim sürecini incelemek için araştırmacılar tarafından kullanılan en uygun

yöntemdir (Panda et al., 2015, 2020). Artikülasyon konuşma sentezi sürecini anlamak için, öncelikle insanların konuşurken kullandıkları organların (dil, çene ve dudaklar gibi) anlaşılması gerekmektedir (Görsel 11’de insan konuşma organlarının modellenmesi bulunmaktadır.) (Panda et al., 2015). Kural tabanlı bir sentezleme türü olan artikülasyon sentezlemede kontrol parametreleri dudak açıklığı, dudak çıkıntısı, dil ucu yüksekliği, dil ucu konumu, dil yüksekliği, dil konumu vb. parametreler kullanılır. Uyarılma parametrelerinde ise ses telleri açıklığı, tellerin gerilimi ve akciğer basıncı gibi parametreler kullanılır. Konuşurken solunum kasları bir enerji kaynağı olarak görev yapar ve akciğerler basınçlı hava depolar. Ses yolundaki kaslar, artikulatörleri hareket ettirir ses yolunun şeklini değiştirerek farklı sesleri üretebilmeyi mümkün hale getirir (Panda et al., 2015, 2020).



Şekil 11. Sol tarafta konuşma için gerekli organları gösterilmiştir. Sağ tarafta ise insan konuşma organları modellenmiştir (Panda et al., 2015).

Artikülasyon konuşma sentezi, otomatik yüz animasyonu ve konuşma bozukluklarının tedavisi gibi uygulamalarda başarılı sonuçlar verir aynı zamanda sesin fonetik özelliklerini, konuşmacının fizyolojik özelliklerini (cinsiyet, yaş, duygusal durum) ve konuşma tarzını tanımlamak için kullanılabilir (Guangpu Huang & Meng Joo Er, 2012).

4.2 Formant Sentezleme

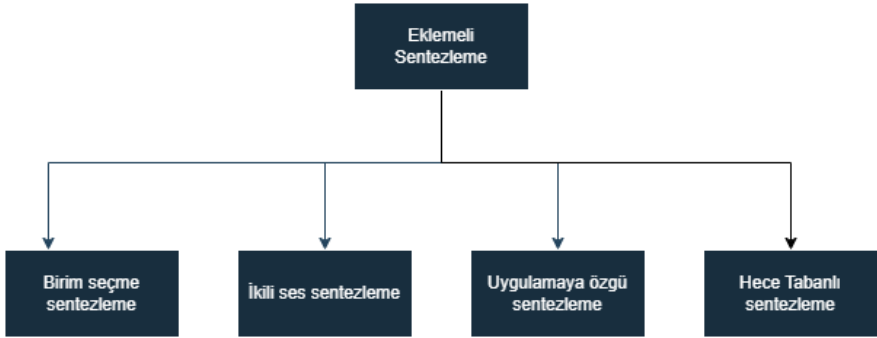
Formant sentezleme, sinir ağlarının ortaya çıkmasından önce konuşma sentezi için en iyi sentezleme yöntemi olarak kullanılıyordu. Sinir ağlarının otomatik olarak konuşmada desenleri ve yapıları bulma yeteneğine sahip olmadığı dönemde, formant sentezi kullanılarak karmaşık ve matematiksel yöntemler ile sağlam modeller oluşturuluyordu (Mchargue & Podgurski, 2023). Formant sentezi tekniği, kurallara dayalı bir sentezleme tekniği olup artikülasyon sentezi ile yakından ilişkilidir. Bu teknik, doğal konuşmanın formant yapısını mümkün olduğunca yakından taklit eden belirli kurallara dayalı yapay sinyaller üreterek konuşma oluşturur (Panda et al., 2015). Konuşma sentezlemek için temel frekans, ses genliği ve gürültü seviyesi gibi farklı parametrelere sahip bir akustik model kullanır (Manero Alvarez Advisors et al., 2022; Muhammed et al., 2019). Konuşmayı üretmek için bir konuşma veri tabanına bağlı olmadıkları için sesleri birleştiren tekniklere göre daha küçük programlardır. Bu nedenle, formant sentezi, bellek ve mikro işlemci gücünün sınırlı olduğu gömülü sistemler için uygun bir konuşma sentezi tekniği olabilir. Formant sentezinin başlıca dezavantajı, sistemin insan tarafından konuşulan doğal konuşmadan oldukça farklı olan yapay, robotik bir konuşma üretmesidir (Muhammed et al., 2019). Ayrıca, kaynak ve tüm filtre parametrelerinin zamanlamasını belirleyen kuralları tasarlamak basit kelimeler için bile oldukça zor olabilir (Panda et al., 2020).

4.3 Eklemeli Sentezleme

Birleştirici konuşma sentezi tekniği, veri tabanına kaydedilmiş konuşma örneklerini (kelimeler, heceler, yarım heceler, fonemler vb.) kullanarak girdi metin ifadelerine göre uygun birimleri birleştirerek konuşma üretmek için kullanılan sentezleme yöntemidir (Panda et al., 2020).

Artikülasyon ve Formant sentezi gibi kurallı sentezdeki zorlukları ortadan kaldırmak amacıyla geliştirilmiştir (Muhammed et al., 2019). Ses sentezi kalitesi, veri tabanındaki birimlerin

uzunluğundan etkilenir daha uzun birimler kullanarak sentezlenen konuşmanın doğallığı artar, ancak daha uzun birimler kullanıldığında birleştirme noktaları azalır ve bu da birleştirme noktalarında doğal olmayan bölümlerin oluşma olasılığını azaltır. Bununla birlikte, daha fazla belleğe ihtiyaç duyulur ve veri tabanındaki birim sayısı artar. Öte yandan, daha kısa birimler kullanıldığında bellek gereksinimi azalır, ancak örnek toplama ve etiketleme tekniklerinin karmaşıklığı artar (Panda et al., 2015). Eklemeli sentezlemenin ürettiği yapay konuşma genel olarak yüksek anlaşılabilirlik ve insan sesine yakın bir sonuç verir. Ancak veri tabanında yüksek yer kaplar ve vurgu, duygu, tonlama gibi konularda eksiklikler yaşar. Bu eksiklerin nedeniyle üretilen sesler doğal ve duygusal değildir (Tan et al., 2021). Eklemeli sentezleme başlığı altında birçok sentezleme yöntemi bulunmaktadır. Bu çalışmada eklemeli sentezlemenin yaygın olarak kullanılan 4 türü anlatılacaktır (Şekil 12).



Şekil 12. Temel Eklemeli Sentezleme Yöntemleri

4.3.1 Birim Seçme Sentezleme

Birim seçimi sentezi, büyük veri tabanlarını kullanır. Bu veri tabanlarını oluştururken, her kaydedilmiş konuşma örneği, kelimeler, heceler gibi küçük parçalara ayrılır. Genellikle bu ayrımı yaparken özel bir konuşma tanıma sistemi kullanılır ve ardından el ile düzeltmeler yapılır. Her bir konuşma örneğinin, temel frekanstan, süreye kadar çeşitli özelliklerle indekslendiği bir veri tabanı

oluşturulur. Sentez işlemi yapılırken oluşturulacak konuşma veri tabanındaki en uygun ifadelerden seçilir. Uygun ifadeleri seçmek için karar ağaçları kullanılır. Birim seçme sentezi, hedef maliyeti ölçmek için ses birimlerinin birbirine olan benzerliğini ve uzaklığını ölçerek en iyi eşleşmeyi bulmaya çalışır. Bu sentezleme yönteminde veri tabanındaki konuşmalara az miktarda müdahale edildiği için doğal konuşma oluşturulabilmektedir. Yüksek doğallık seviyelerine ulaşabilmek için kaydedilmiş seslerin büyük bir veri tabanı gereklidir ve bunlar onlarca saatlik konuşma ve terabaytlarca bellek gerektirebilir. Birçok metin okuma sistemleri için bu kadar büyük miktarda veri ve gereken hesaplama kaynakları uygulanabilir değildir. Başka bir dezavantajı da birim seçimi algoritmalarının, en iyi sentezin gerçekleşmesi için daha iyi bir seçenek olsa bile bazı durumlarda daha kötü alternatifleri seçmektedir (Panda et al., 2015; Sairanen, 2023).

4.3.2 İkili Ses Sentezleme

İkili Ses Sentezleme bir konuşmanın seslerinin geçişlerini bir araya getirerek ses üretme yöntemidir. Bu yöntem, bir dilin temel seslerini temsil eden parçaları kullanır. Bir sesin bir diğerine geçişi sırasındaki ses değişimlerini yakalamaktır. Farklı dillerde bu ses değişimlerine ilişkin belirli kurallar olduğundan, her dilin bir sınırlı sayıda ses geçiş parçası vardır. Yapay konuşma üretebilmek için, metin önce temel seslere dönüştürülür ve sonra bu sesler veri tabanındaki uygun ses geçişleri ile eşleştirilir. Sonra, ses geçişleri arasındaki geçişi düzeltmek ve vurgu, süre ve tonlamayı ayarlamak için dijital işlemler yapılır. İkili ses sentezlemenin avantajı, veri tabanının sınırlı olmasıdır, ancak sentezlenen ses monoton ve doğal olmayan sesler üretilmesine yol açabilmektedir (Sairanen, 2023).

4.3.3 Uygulamaya Özgü Sentezleme

Uygulamaya özgü sentezlemede ifadeleri oluşturmak için genellikle veri tabanından sınırlı kelime ve cümlelerin birimlerini seçerek çalışan bir ses sentezleme yöntemidir. Çoğunlukla belirli bir konu veya uygulama alanına yöneliktir, örneğin konuşan saatler,

ulařım tarifleri, hava raporları, tren sorgular gibi alanlarda kullanılır. Basit ve uygulanması kolay olması nedeniyle uzun süredir ticari kullanımdadır. Uygulamaya özgü sentezlemenin en büyük avantajı, doğallık seviyesidir. Cümle türlerinin çeşitlilięi kısıtlı olduęu için çok yüksek bir doğallık seviyesine sahip olabileceęi gibi kaydedilen orijinal konuşmanın vurgusu ve tonlamasına da çok yakındır. Ancak sistemin kullanımı sınırlıdır çünkü sadece veri tabanındaki kelimeleri ve ifadeleri kullanabilir. Yani, özelleřtirme veya daha genel konuşmalar için uygun deęildir. Bu nedenle, genel kullanım amaçları için uygun bir seçenek deęildir (Panda et al., 2015; Sairanen, 2023).

4.3.4 Hece Tabanlı Sentezleme

Hece tabanlı sentezleme yönteminde heceler, bir kelimenin fonetik yapı taşları olarak kabul edilir. Hecelere dayalı konuşma sentezi teknięi, herhangi bir dilde ortak bir hece birimi seti için kaydedilmiş konuşma örneklerini saklar. Genellikle bir hece oluşturulurken başlangıç ve bitişlerine ünsüz harfler, ünsüz harflerin arasına ise ünlü harf ekleyerek 3 harften oluşan heceler oluşturulur. Hecelere dayalı konuşma sentezi sistemi oluşturmak, sistemin oluşturulacaęı dildeki hece birimlerini iyi tanımlamak gerekmektedir. Her dildeki hece birimlerinin toplam sayısı, dilin özelliklerine baęlı deęişmektedir. Kelimeleri hecelerden oluřturan dillerde başarılı sonuçlar verebilir ancak veri tabanını tasarlamak zor bir görevdir (Panda et al., 2015, 2020).

4.4 İstatiksel Parametrik Sentezleme

İstatistiksel Parametrik Konuşma Sentezi (SPSS), Metin Konuşma Sentezi (TTS) sistemlerinin bir türüdür ve son zamanlarda oldukça popüler hale gelmiştir (Muhammed et al., 2019). Bu yöntem konuşma sentezi için gereken parametreleri çıkarmak için öğrenme verileri üzerinde istatistiksel analizler kullanır. Özellikle konuşma kayıtlarındaki ses özellikleri ve dięer parametreler üzerinde çalışarak, temel frekans, spektral özellikler gibi ana parametreleri belirler. Bu belirlenen parametreler daha sonra sentezleme anında

kullanılarak, konuşma dalgasını oluşturmak için vocoder'lar aracılığıyla işlenir. SPSS'nin temel yaklaşımı, eğitim veri tabanındaki geniş bir konuşma veri setini kullanarak öğrenme işlemi gerçekleştirmesidir. Bu süreçte, konuşma kayıtlarındaki çeşitli özelliklerin istatistiksel analizleri yapılır ve konuşma için temel parametreler bu analizler sonucunda belirlenir. Örneğin, konuşmacının temel frekansı, sesin yüksekliği veya alçaklığı gibi temel ses özellikleri bu analizlerle tespit edilir. SPSS'nin en büyük avantajlarından biri, öğrenme verileri üzerinde çalışırken dil bağımsız olmasıdır. Yani, farklı dillerde konuşma sentezi için kullanılabilir ve genellikle çok dilli konuşma sentezinde de etkilidir. Bununla birlikte, SPSS'nin başlıca zorluklarından biri, eğitim veri setinin kalitesi ve çeşitliliğidir. Veri setinin yeterli olmaması veya çeşitlilik eksikliği, sentezlenen konuşmanın kalitesini etkileyebilir. Ayrıca, sentezleme sürecinde gerçekçi ve doğal bir konuşma üretmek için daha gelişmiş algoritmalar ve veri setleri gerekebilir. Genel olarak, SPSS, konuşma sentezi teknolojilerinin ileri düzeydeki bir örneği olarak kabul edilir ve geniş bir kullanım alanına sahiptir. Ancak, sürekli olarak geliştirilmekte olan bir alandır ve daha fazla veri çeşitliliği ve daha sofistike analiz yöntemleriyle gelecekte daha doğal ve etkileyici konuşma sentezi sistemleri sağlamayı hedefler.

4.4.1 Hidden Markov Model Tabanlı Sentezleme

HMM tabanlı sentez, konuşma sentezi üretmek için geliştirilmiş bir tekniktir. Bu teknik, sembolik parametrelerle çalışarak doğal dil işleme biriminden gelen bilgiyi kullanır ve konuşma sentezi üretir. HMM adını verdiğimiz istatistiksel bir modelleme yöntemi kullanılır. Bu sentezleme türünün birim seçimi sentezine göre bazı avantajları vardır. İlk olarak, işitsel hatalardan bağımsız olma özelliği vardır. Sembolik parametreler üzerinden sentezleme işlemi yapıldığı için, işitsel hataların etkisi daha azdır. Bu, daha doğal ve anlaşılabilir bir konuşma sentezi elde edilmesini sağlar. Bir diğer önemli avantajı ise depolama alanının boyutudur. Birim seçimi sentezi, büyük bir ses veri tabanına dayanır ve bu

nedenle genellikle çok büyük bir depolama alanı gerektirir. Ancak HMM tabanlı sentez, sembolik parametrelerle çalıştığı için depolama alanı gereksinimi daha küçüktür. Bu, özellikle taşınabilir cihazlarda kullanımı için idealdir çünkü daha az depolama alanı kullanarak bu teknolojinin cihazlara entegre edilmesini mümkün kılar. Sonuç olarak, HMM tabanlı sentez, işitsel hatalardan daha az etkilenen ve daha küçük depolama alanı gerektiren bir sentezleme tekniğidir. Bu özellikleri, daha doğal konuşma sentezleri üretmek ve taşınabilir cihazlarda kullanılabilirlik sağlamak açısından önemli bir avantaj sunar (Kaur & Singh, 2023; Muhammed et al., 2019).

4.4.2 Derin Sinir Ağı Tabanlı Sentezleme

Derin Sinir Ağları makine öğrenmesi ve yapay zekâ alanında kullanılan bir tür sinir ağıdır. Bu teknoloji, biyolojik sinir ağlarını taklit eden yapay bir modelleme biçimine dayanır. DNN'ler, çok katmanlı yapıları sahip olduğundan "derin" olarak adlandırılır. Her katman, önceki katmanın çıktılarından öğrenilen özellikleri içerir, bu da sistemde karmaşıklığı artırır ve daha yüksek düzeyde öğrenme sağlar. Konuşma işleme alanında, DNN'ler genellikle ses ve konuşma verilerinin karmaşıklığını anlamak, özelliklerini öğrenmek ve bu verileri sentezleme veya tanıma amacıyla kullanmak için kullanılır. DNN'ler özellikle HMM tabanlı konuşma sentezi sistemlerinin kısıtlamalarını aşmak ve daha iyi ses kalitesi elde etmek amacıyla önerilmiştir. DNN tabanlı konuşma sentezi, genellikle daha önceki tekniklere kıyasla daha fazla esneklik sunar. Örneğin, HMM tabanlı sistemler genellikle belirli bir konuşma tarzında kısıtlanırken, DNN'ler çeşitli konuşma tarzları üretebilir. Bu, ses dönüştürme, duygu sentezi ve lehçe gibi uygulamalarda çok daha çeşitli ve ifade dolu konuşmaların elde edilmesine olanak tanır. Ayrıca, uzun-kısa dönemli bellekli tekrarlayan sinir ağları (LSTM-RNN'ler) gibi DNN türleri, özellikle konuşma işleme alanında önemli başarılar elde etmiştir. Bu tür ağlar, önceki bilgileri hatırlayabilme yetenekleri sayesinde ses sentezi ve tanıma süreçlerinde daha etkili ve doğru sonuçlar elde etme potansiyeline sahiptir. Dolayısıyla, DNN tabanlı konuşma sentezi, konuşma işleme

alanında gelişmiş ve esnek bir yaklaşım sunarak, önceki tekniklerin sınırlamalarını aşma ve daha kaliteli, çeşitli ve ifade dolu sesler elde etme potansiyeli taşımaktadır (Kaur & Singh, 2023).

5. TTS Teknolojisinin Temel İşleyişi

TTS teknolojisinin temel işleyişi oldukça karmaşık ve çoklu bileşenler oluşmaktadır. Genellikle metinlerin sese dönüştürülmesi süreci metin analizi, akustik modelleme ve dalga formu sentezleme ile çıktının oluşturulması aşamalarından oluşmaktadır (Tan et al., 2021). Üç aşamadan her biri, son kullanıcının duyabileceği ve anlayabileceği doğal sesli bir çıktı oluşturmak için önemlidir.

5. 1. Metin Analizi

Metin Analizi, TTS teknolojisinin temel işleyişinin ilk aşamasıdır. Bu aşama metni daha küçük ve işlenebilir parçalara böler, metindeki dil bilgisi yapısını anlamayı amaçlar ve özel karakterleri ve sembolleri tanır. Metin analizin aşamaları aşağıdaki gibidir.

- **Metin Normalizasyonu**

Metin normalizasyonunda birkaç farklı aşamadan geçerek metindeki özel karakterler ve ifadeler yazıya dönüştürülür. Örneğin artı “+”, eksi “-”, yüzde “%”, üstü “^” gibi seslendirilebilen ifadeler yazıya dönüştürülür. “2023” gibi rakamla yazılmış sayılar “iki bin yirmi üç” gibi yazıya dönüştürülür. Kısaltmaların kullanılan ifadelerin uzun halleri yazıya çevrilir örneğin “vb.” ifadesi “ve benzeri” olarak yazılır. Kesme işareti “” , “(”, “{” gibi seslendirilmeyen ses özelliklerini etkilemeyecek ifadeler metinden çıkartılır (Manero Alvarez Advisors et al., 2022).

- **Karakter Fonem Dönüşümü**

Bu adım, yazılı metinde yer alan karakterleri, konuşma dilindeki seslerin temel birimleri olan fonemlere dönüştürmeyi içerir. Fonemler, dilin ses yapısını temsil eden en küçük birimlerdir ve bir dildeki ses farklılıklarını ifade ederler.

- **Özel Kelimeleri Tanıma**

Metindeki özel isimler, yabancı kelimeler ve belirli terimler gibi özel tanıma görevleri de bu aşamada gerçekleştirilir. Bu, TTS sisteminin bu özel kelimeleri doğru bir şekilde telaffuz etmesini sağlar.

- **Ses Birimleri Tanıma**

Bazı TTS sistemleri, daha yüksek kalitede ses sentezi için metindeki ses birimlerini tanır. Özellikle dilin belirli seslerini doğru bir şekilde yakalamak için bu adım önemlidir.

- **Duygu ve Tonlama Analizi**

Bu aşama bazı TTS sistemlerinde kullanılır. Metinde ifade edilen duygusal durumları veya metindeki tonlama değişikliklerini tanımak amacıyla kullanılır. Örneğin, bir metinde neşeli, üzgün, endişeli veya sakin bir tonun ifade edilip edilmediği analiz edilir.

Metin analizi, yazılı metni anlamak ve parçalara ayırmak için doğal dil işleme (NLP) tekniklerini içerir. Sesli çıktının doğal ve akıcı bir şekilde okunmasını sağlar. Ayrıca, metindeki özel terimler veya isimler gibi öğelerin doğru bir şekilde seslendirilmesine olanak tanır. İşitme engelliler ve farklı dil konuşanlar gibi birçok kullanıcı için doğru seslendirme önemlidir. Metin analizi, metin tabanlı bilgilere daha geniş bir kitleye ulaşma fırsatı sunar ve TTS teknolojisinin bu önemli aşamasının başarılı bir şekilde uygulanması, son sesli çıktının kalitesini belirler.

5. 2. Duygu ve Tonlama Analizi

Duygu ve Tonlama Analizi, TTS teknolojisinin metin analizi aşamasının bir parçası olarak bazı sistemlerde kullanılan özel bir aşamadır. Bu aşama, metinde ifade edilen duygusal durumları veya metindeki tonlama değişikliklerini tanımak için tasarlanmıştır. Özellikle sesli çıktının metni daha canlı, anlamlı ve insan gibi seslendirmesi için kullanılır. Duygu ve Tonlama Analizindeki temel aşamalar aşağıdaki gibidir.

- **Duygu Analizi**

Duygu analizi, metinde ifade edilen duygusal durumları tanımak amacıyla kullanılır. Bir metindeki metin analizi aşamasının ardından, duygu analizi bu metni inceleyerek neşe, üzüntü, öfke, heyecan, endişe veya sakinlik gibi duygusal ifadeleri tespit eder. Örneğin, bir metinde geçen "mutlu" kelimesi veya olumlu cümle yapıları, içerikte bir neşe veya mutluluk ifadesi olduğunu işaret edebilir.

- **Tonlama Analizi**

Tonlama analizi, metindeki tonlama değişikliklerini tanımak için kullanılır. Metin, farklı tonlamalarla okunduğunda anlamı büyük ölçüde değiştirebilir. Bu aşama, metindeki vurgu, vurgu değişiklikleri ve tonlama farklılıklarını algılamak amacıyla kullanılır. Özellikle konuşmanın akıcılığı ve anlamını artırmak için metindeki bu tür değişiklikleri tanımlar.

- **Duygu ve Tonlama Etiketlemesi**

Duygu ve tonlama analizi sonuçları, metindeki ilgili bölümlere etiketler veya işaretleme ekler. Bu, ses sentezi aşamasında sesin nasıl modüle edileceğini belirlemek için kullanılır. Örneğin, mutlu bir ifade içeren bir cümle daha yüksek bir ses tonu ve pozitif bir vurgu ile seslendirilebilirken, üzgün bir ifade

- **Ses Modülasyonu**

Duygu ve tonlama analizi sonuçlarına dayanarak ses modülasyonu yapılır. Ses modülasyonu, metindeki duygusal ifadeleri veya tonlama değişikliklerini yansıtmak için sesin yüksekliği, hızı ve vurgusu gibi özelliklerin ayarlanmasıdır. Bu, sesli çıktının daha insan gibi ve anlamlı bir şekilde duyulmasını sağlar.

Duygu ve tonlama analizi, özellikle metin tabanlı içeriklerin sesli olarak okunmasının daha duygu dolu ve etkili olmasını sağlar. Örneğin, sesli kitapların veya metin tabanlı reklamların daha çekici ve etkileyici olmasına yardımcı olabilir. Bu aşama, insan sesini taklit etmek ve metni daha canlı bir şekilde sunmak için önemlidir. Ancak

duygu ve tonlama analizi, bazı metinler için daha karmaşık veya özelleştirilmiş olabilir ve her TTS sistemi tarafından kullanılmayabilir. Bu aşama, metin tabanlı içeriğin daha anlamlı ve duygusal bir şekilde seslendirilmesini isteyen uygulamalarda kullanılır.

5. 3. Ses Sentezi

Ses Sentezi, TTS teknolojisinin önemli bir aşamasını oluşturur ve metin analizi sonuçlarına dayanarak yazılı metinleri sesli olarak çıkarmak için kullanılır. Bu aşama, metinde tanımlanan metin birimlerini, kelimeleri, heceleri veya cümleleri seslendirmek için gerekli verileri kullanır. Ses sentezi sırasında aşağıdaki aşamalar aşağıdaki gibidir.

- **Metin Birimleri Seslendirme**

Metin analizi aşamasının sonuçlarına dayanarak, ses sentezi aşamasında metinde tanımlanan metin birimleri seslendirilir. Bu birimler, genellikle kelimeler, heceler veya cümleler olabilir. Özel dikkat, metindeki dil bilgisi yapısı ve vurgu değişikliklerine verilir, böylece sesli çıktı doğal ve akıcı olur.

- **Dil ve Aksan Uyumu**

Ses sentezi aşamasında, metin belirli bir dil ve aksanla uyumlu bir şekilde seslendirilir. Bu, metin içeriğinin hedef kitlesi veya kullanıcısı için daha anlaşılır ve erişilebilir olmasını sağlar. Özellikle çok dilli TTS sistemlerinde, metin hangi dilde ve hangi aksanla seslendirilmesi gerektiği konusunda dikkate alınır.

- **Doğal Ses Modülasyonu**

Ses sentezi, metindeki vurgu, tonlama ve duygusal ifadeleri yansıtmak için sesin yüksekliği, hızı ve vurgusu gibi özelliklerin ayarlanmasıyla gerçekleştirilir. Örneğin, bir mutlu ifade daha yüksek bir ses tonu ve pozitif bir vurgu ile seslendirilirken, üzgün bir ifade daha düşük bir ses tonu ve daha sakin bir vurgu ile seslendirilir.

- **Ses Akışı ve Akıcılık**

Ses sentezi aşamasında, metin birimleri arasındaki geçişler düzgün bir akıcılık sağlayacak şekilde gerçekleştirilir. Bu, sesli çıktının daha doğal ve kolay dinlenir olmasını sağlar.

- **Sesli Özellikler ve Hız**

Ses sentezi sırasında sesli çıktının özellikleri de belirlenir. Ses hızı, ses yüksekliği ve vurgu gibi özellikler, metnin anlamını ve ifadesini daha iyi yansıtmak için ayarlanır.

- **Ses Kalitesi ve Yükseklik**

TTS teknolojisinin ses kalitesi, kullanılan ses motoruna, ses veri tabanına ve algoritmalara bağlı olarak değişebilir. Ses sentezi aşamasında, yüksek kaliteli seslerin kullanılması ve doğal sesin yakalanması için özel önem verilir.

Ses sentezi aşaması, TTS teknolojisinin metinleri sesli olarak çıkarmak için kullanılan en karmaşık ve önemli aşamalardan biridir. Bu aşama, metni insan sesini taklit ederek sesli bir çıktıya dönüştürür. Başarılı bir ses sentezi, metin tabanlı içeriğin daha geniş bir kullanıcı kitlesi için erişilebilir hale gelmesini sağlar ve birçok uygulama alanında kullanılır, örneğin sesli kitaplar, sesli asistanlar, reklamcılık ve erişilebilirlik uygulamaları gibi. Ses sentezi aşamasının kalitesi, TTS teknolojisinin genel etkinliği ve kullanılabilirliği üzerinde büyük bir etkiye sahiptir.

5. 4. Ses Çıkışı Oluşturma

Ses sentezlemesi tamamlandığında, ses çıkışı oluşturma aşamasına geçilir. Bu aşamada, sentezlenmiş ses birimleri birleştirilir ve son kullanıcıya sunulacak sesli çıktı oluşturulur. Bu aşamada ses tonlaması, vurgu ve hız gibi özellikler de dikkate alınır. Ayrıca, sesli çıktının dosya formatı ve sıkıştırma gibi teknik detayları da burada belirlenir. Bu işleyiş TTS teknolojisinin temelini oluşturur. Her bir adım metin tabanlı verilerin sesli çıktıya dönüştürülmesinde önemlidir ve sonuç metni duyan veya kullanıcının metni sesli olarak dinlediği doğal bir sesli çıktıdır. TTS

teknolojisi, her geçen gün daha da gelişmekte ve birçok uygulama alanında kullanılmaktadır. Bu işleyişin anlaşılması, TTS teknolojisinin nasıl çalıştığını ve neden bu kadar önemli olduğunu anlamak için önemlidir.

6. TTS Teknolojisinin Uygulamaları

TTS (Text-to-Speech) teknolojisi günümüzde metin tabanlı verilerin sesli olarak sentezlendiği çok çeşitli uygulama alanlarında kullanılmaktadır. Bu bölümde, TTS teknolojisinin farklı sektörlerdeki bazı örnek uygulamalarını ve önemini inceleyeceğiz.

6. 1. Eğitim Materyalleri

Text to Speech (TTS) sistemleri okuma problemleri çeken öğrencilere destek sağlama ya da akıllı öğretim sistemleri için yardımcı teknoloji olarak birçok farklı amaçla eğitimde kullanılmıştır. Text to Speech ile verilen eğitimlerin öğrenmeyi kolaylaştırması ya da zorlaştırması hakkındaki bulgular tutarsız olsa da kolay erişebilirlik ve düşük maliyetli olması nedeniyle eğitim materyallerinde kullanılmaya devam edilmektedir (Dai et al., 2022). Bern Üniversitesi Tıp Fakültesi bazı eğitim içerikleri için seslendirme kullanmaktadır. Seslendirme maliyetini düşürmek amacıyla profesyonel seslendirmeleri, TTS ile üretilmiş sesler ile değiştirmeye karar verir. 100.000’den fazla kelimeyi içeren yirmi e-öğrenme modülü, TTS teknolojisi ile seslendirilir ve 20 saatlik ses üretir. Üretilen seslerin etkisi üzerine yapılan çalışmada tıbbi bilgi öğrenimi için TTS teknolojisinin etkili bir yöntem olması ve 48 bin dolar tasarruf edildiği gözlemlenir (Minder et al., 2012). TTS sistemleri yabancı dil öğreniminde de birçok fayda sağlamaktadır. Google Translate gibi çeviri yapan uygulamaların TTS sistemleri sayesinde kelimelerin telaffuz ve tonlama gibi özelliklerinin öğrenilmesine olanak tanır. Yapılan araştırmalarda öğrenme sonucu açısından TTS sesiyle öğrenen öğrencilerle insan sesiyle öğrenen öğrenciler arasında anlamlı bir fark bulunamadığı gözlemlenmiştir (Dai et al., 2022). Ayrıca TTS teknolojisi görme engelli öğrenciler içinde öğrenme fırsatı sağlamaktadır. Görme engelli öğrenciler

eđitim ieriklerinin sesli versiyonlarını birilerinin okumasına ihtiya duymadan ğrenebilmelerini sađlar. Metinleri seslendirmek iin hazırlanan birok web sitesi ve tarayıcı eklentisi bulunmaktadır. Ayrıca Microsoft Word gibi bazı uygulamaların kendi seslendirme sistemleri bulunmaktadır.

6. 2. Sesli Asistanlar

İnsanlar neredeyse bilgisayarların icat edildiđi günden itibaren bilgisayarlarla konuşmak istemişlerdir. Bu istek dođrultusunda sesli asistanlar geliştirilerek TTS teknolojisinin dikkat eken uygulama alanlarından birisi olmuştur (Hoy, 2018). Metinden konuşma üretme (Text To Speech) teknolojisi ile konuşmadan metin üretme (Speech to Text) teknolojisi birleşerek sesli asistanlar oluşturulmuştur (Siri Team, 2017). Sesli asistan uygulamaları Speech To Text teknolojisi ile sürekli olarak ortam sesini dinleyerek anahtar kelime veya kelimelerin ađırılmasını bekler. Anahtar kelime seslendirildiđinde (Örneđin : Hey Siri) aktif hale gelerek kullanıcının komutlarını dinler ve yerine getirir. Kullanıcı ile etkileşim kurmak ve komutlarına cevap vermek iin ise TTS teknolojisi kullanılır. Sesli asistanlara örnek olarak Siri, Google Asistan, Amazon Alexa ve Microsoft Cortana verilebilir. Apple'ın geliştirdiđi Siri sesli asistan uygulaması en uzun süredir var kullanılan sesli asistandır. 2010 yılında bađımsız bir uygulama olarak piyasaya ıkmış ve 2011'de iOS'a sistemine dahil edilmiştir. Microsoft Siri'den kısa bir süre sonra 2013'te Cortana'yı piyasaya sürmüştür. Amazon, 2014 yılında akıllı ev hoparlörü ile tanıtılan Alexa'yı piyasaya sürdü. Google 2016 yılında akıllı ev hoparlörü ile birlikte duyurduđu Google Assistant'ı tanıttı aynı zamanda Android tabanlı akıllı telefonlardaki Google uygulamasına entegre etti (Hoy, 2018).

6. 3. Sesli Kitaplar

Sesli kitaplar, kitapların seslendirmenler ya da TTS teknolojisi ile sesli olarak kaydedilmiş versiyonlarıdır. Son yıllarda büyük popülerlik kazanan sesli kitaplar 1933 yılından itibaren on binlerce sayıda üretilmiştir. Sesli kitaplar genellikle profesyonel aktörler

tarafından kaydedilir ve okuyucunun duyguları, diyaloglar ve anlatıdaki çevre ve karakter tanımlamalarını vurgulamak için okuyucu tarafından dramatik seslendirmeleri ve tonlamaları içerir. Modern TTS teknolojileri insan konuşmasına yakın sesler üretse de hikâye anlatımı gibi alanlarda önemli bir performans farkları bulunmaktadır (Pethe et al., 2023). Performans farklarına rağmen sesli versiyonları olmayan kitap ve e-kitapların TTS teknolojisi ile seslendirilmesi çocuklar, görme engelliler, disleksi ve yeni dil öğrenenler için içeriği erişilebilir kılar. TTS teknolojisini kullanarak e-kitapları seslendiren popüler TTS uygulamaları;

- Google Play Kitaplar
- Amazon Kindle
- Apple Books
- Libby
- Speechify

7. TTS Teknolojisinin Geleceği

Günümüzde kullanılan bazı TTS uygulamalarının gelecekte gelişen teknoloji ile birlikte daha etkin kullanılması beklenmektedir. Örneğin reklamcılık sektöründe basit reklamlar için TTS teknolojisi kullanılmaya başlanmıştır. TTS teknolojisi gelecekte ise reklamcılık alanında kişiselleştirilmiş içeriklerin oluşturulmasını kolaylaştırır. Reklamcılar potansiyel müşterilere özelleştirilmiş sesli reklamlar sunarak daha iyi bir kullanıcı deneyimi ve etkili bir pazarlama stratejisi oluşturabilirler. Yüksek kaliteli bir ses ile kullanıcının adını içeren bir sesli reklam müşterileri etkileyecektir. Ayrıca TTS teknolojisi reklamların çok dilli olmasına da olanak tanır. Her dil için ayrı ayrı dublaj yapmanın oluşturduğu maliyeti ortadan kaldırır. Şu anda gelişmekte olan ve IVR sistemlerinde kullanılarak müşteri hizmetleri ve çağrı merkezi uygulamalarının geliştirilmesine TTS sistemlerinin katkıları bulunmaktadır. Müşteriler, sesli yanıtlar ve talimatlar aracılığıyla etkileşime girebilirler. Günümüzde ilk örnekleri görülen sistemlerin ilerleyen süreçte insanlara gerek kalmayacak şekilde geliştirilmesi muhtemeldir. TTS teknolojisinin

gelecekte deđiřtireceđi alanlardan bir diđerisi ise oyun sektörüdür. Günümüzde kayıtlar kullanılan karakter seslendirmelerinin yerini ilerleyen süreçte TTS teknolojisinin alabileceđi öngörülmektedir. Özelleřtirilmiř karaktere verilen isimleri oyunda duymak daha fazla etkileřim ve katılım sađlama potansiyeli sađlar. Bu alanın geleceđi, daha fazla oyun geliřtiricisinin ve ierik üreticisinin TTS teknolojisinin potansiyelini keřfetmesi ve bu alanda yeniliki uygulamalar geliřtirmesi ile řekillenmeye devam edecek gibi görülmektedir.

TTS teknolojisinin geliřimi, kitapları sesli hale getirirken her karakter iin özelleřtirilmiř ses tonları, aksanlar ve duygu ifadeleri sunma potansiyeli tařıyor. Bu teknoloji, bir kitaptaki farklı karakterlere benzersiz sesler atayarak, dinleyicilere karakterler arasında net bir ayırım sađlayabilir. Örneđin, bir romanda geen her karaktere farklı bir ses tonu ve hatta özel bir aksan verilebilir. Böylece, dinleyiciler karakterlerin duygusal durumlarına göre ses tonlarının deđiřimini deneyimleyebilir, bir filmdeki gibi her karakter iin farklı kiřiler tarafından seslendirilmiř gibi bir etki oluřturulabilir. Bu, okuyucuların metnin derinliklerine daha fazla dalmasına ve karakterlerle daha yakından bađ kurmasına olanak tanırken, sesli kitap deneyimini daha zengin ve kiřiselleřtirilmiř hale getirebilir.

Deep Fake ve TTS, yapay zekâ teknolojileri kullanılarak insanların yüz, ses ve hareketlerini taklit eden veya deđiřtiren uygulamalardır. Bu uygulamalar, eđlence, sanat, eđitim, sađlık ve güvenlik gibi pek ok alanda faydalı olabilirler. Örneđin, Deep Fake ile bir filmdeki oyuncunun yüzünü bařka bir ünlüye veya tarihi bir kiřiye dönüřtürmek mümkündür. TTS ile ise, insan sesini taklit ederek metinleri sesli olarak okutmak veya farklı dillerde konuřmak mümkündür. Bu uygulamalar, gelecekte daha da geliřerek, insanların gereklik algısını deđiřtirebilir ve yeni sanal deneyimler sunabilirler. Ancak, Deep Fake ve TTS uygulamalarının kötüye kullanılması da büyük bir tehlike oluřurmaktadır. Bu uygulamalar, insanların kimliklerini almak, yalan haberler yaymak, siyasi veya ideolojik amalarla manipüle etmek, řantaj veya taciz yapmak gibi

suçlara yol açabilirler. Bu nedenle, bu uygulamaların etik ve yasal sınırlarını belirlemek ve denetlemek çok önemlidir. Ayrıca, insanların bu uygulamaları nasıl ayırt edebileceklerine dair bilinç ve becerilerini geliştirmek de gereklidir. Deep Fake ve TTS uygulamaları, yapay zekâ teknolojisinin hem faydalarını hem de risklerini gösteren örneklerdir. Bu uygulamaların gelecekte nasıl olacağı, insanların bu teknolojiyi nasıl kullandığına ve yönettiğine bağlıdır.

KAYNAKÇA

Akhmetzhanov, N. (2021). Implementation of a Multispeaker Text-to-Speech Synthesis Web Application.

Brackhane, F. (2015). Kempelen vs. Kratzenstein – Researchers on speech synthesis in times of change.

Chettri, B., & Shah, K. B. (2013). Nepali Text to Speech Synthesis System using ESNOLA Method of Concatenation. In International Journal of Computer Applications (Vol. 62, Issue 2).

Dai, L., Kritskaia, V., Van Der Velden, E., Jung, M. M., Postma, M., & Louwerse, M. M. (2022). Evaluating the usage of Text-To-Speech in K12 education. ACM International Conference Proceeding Series, 182–188. <https://doi.org/10.1145/3578837.3578864>

Guangpu Huang, & Meng Joo Er. (2012). An Adaptive Control Scheme for Articulatory Synthesis of Plosive-Vowel Sequences. IEEE.

Hoy, M. B. (2018). Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. Medical Reference Services Quarterly, 37(1), 81–88. <https://doi.org/10.1080/02763869.2018.1404391>

Kaur, N., & Singh, P. (2023). Conventional and contemporary approaches used in text to speech synthesis: a review. Artificial Intelligence Review, 56(7), 5837–5880. <https://doi.org/10.1007/s10462-022-10315-0>

Kleiner, A., & Kurzweil, R. C. (1977). A DESCRIPTION OF THE KURZWEIL READING MACHINE AND A STATUS REPORT ON ITS TESTING AND DISSEMINATION.

Manero Alvarez Advisors, A., Hernaez Rioja Eva Navas Cordon, I., & Azterketa eta Prozesamendua, H. (2022). Implementation and evaluation of a Spanish TTS based on FastPitch.

Mark J. Sussman. (1999). Performing the Intelligent Machine: Deception and Enchantment in the Life of the Automaton Chess Player. <https://muse.jhu.edu/pub/6/article/32960>

Mchargue, J., & Podgurski, A. (2023). EFFICIENT MULTISPEAKER SPEECH SYNTHESIS AND VOICE CLONING.

Minder, S., Notari, M., Schmitz, F., Hofer, R., & Woermann, U. (2012). Computer Generated Voice-Over in a Medical E-Learning Application: The Impact on Factual Learning Outcome. In *Journal of Universal Computer Science* (Vol. 18, Issue 3).

Moran, M. E. (2007). Evolution of robotic arms. *Journal of Robotic Surgery*, 1(2), 103–111. <https://doi.org/10.1007/s11701-006-0002-x>

Muhammed, D., Aliero, A. A., & Ibrahim, A. (2019). Taxonomy, Review and Research Challenges Of DNN-Based Text-To-Speech System for Hausa as Under-Resourced Language. *International Journal of Scientific & Engineering Research*, 10(7). <http://www.ijser.org>

Ohala, J. J. (2011). CHRISTIAN GOTTLIEB KRATZENSTEIN: PIONEER IN SPEECH SYNTHESIS.

Panda, S. P., Nayak, A. K., & Patnaik, S. (2015). Text to Speech Synthesis with an Indian Language Perspective. In *International Journal of Grid and Utility Computing, Inderscience* (Vol. 6, Issue 4).

Panda, S. P., Nayak, A. K., & Rai, S. C. (2020). A survey on speech synthesis techniques in Indian languages. *Multimedia Systems*, 26(4), 453–478. <https://doi.org/10.1007/s00530-020-00659-4>

Pethe, C., Yin, Y., & Skiena, S. (2023). Prosody Analysis of Audiobooks. <http://arxiv.org/abs/2310.06930>

Sairanen, V. (2023). Deep learning text-to-speech synthesis with Flowtron and WaveGlow. www.aalto.fi

Sami Lemmetty. (1999). Review of Speech Synthesis Technology.

Siri Team. (2017, August). Deep Learning for Siri's Voice: On-device Deep Mixture Density Networks for Hybrid Unit Selection Synthesis.

Story, B. H. (2019). History of speech synthesis. In *The Routledge Handbook of Phonetics* (pp. 9–33). Taylor and Francis. <https://doi.org/10.4324/9780429056253-2>

Takayuki Arai. (2013). MECHANICAL MODELS OF THE HUMAN VOCAL TRACT.

Tan, X., Qin, T., Soong, F., & Liu, T.-Y. (2021). A Survey on Neural Speech Synthesis. <http://arxiv.org/abs/2106.15561>

Wikipedia. (n.d.-a). Speech synthesis. Retrieved 29 October 2023, from https://en.wikipedia.org/wiki/Speech_synthesis

Wikipedia. (n.d.-b). Voder. Retrieved 26 November 2023, from <https://en.wikipedia.org/wiki/Voder>

Wikipedia. (2023). Mechanical Turk. https://en.wikipedia.org/wiki/Mechanical_Turk

BÖLÜM II

Kubernetes ile Küme ve Dağıtım Altyapısı Oluşturma, Dağıtım Güncellemelerini Yönetme ve Servislerin Düğüm Üzerinde Çalıştırılması: Deneysel Uygulamalar

Işıl KARABEY AKSAKALLI¹
Levent BAYINDIR²

Giriş

Son yıllarda konteyner orkestrasyon araçlarından popüler hale gelen Kubernetes, Google tarafından oluşturulmuş ve daha sonra Cloud Native Computing Foundation (CNCF) tarafından geliştirilmiştir ve piyasaya sürülmüştür (Shishkalov, 2022). Hafif ve taşınabilir bir uygulamadan oluşan Go dili tabanlı açık kaynak konteyner yönetim aracıdır (Saito, Lee, & Hsu, 2018). Kubernetes kümesi oluşturularak birçok ana bilgisayar üzerinde Docker

¹ Dr. Öğr. Üyesi, Erzurum Teknik Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü

² Dr. Öğr. Üyesi, Kocaeli Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü

konteyner uygulamaları dağıtılabilir, yönetilebilir ve ölçeklenebilir. Kubernetes, konteynerleri bir sunucu kümesi arasında otomatik olarak zamanlayacak şekilde planlamakta ve bu karmaşık işlemi geliştiricilerden ve operatörlerden soyutlamaktadır. Kubernetes kümesini yerel bir makinede oluşturmak ve test etmek için kubectl ve minikube kurulumu yapmak yeterli olmaktadır. Linux bir makinede kubectl kurmak için aşağıdaki adımlar izlenmektedir ("Install Tools," 2023):

```
$ curl -LO
https://storage.googleapis.com/kubernetes-release/release/
$(curl -s
https://storage.googleapis.com/kubernetes-release/release/stable.txt)
/bin/linux/amd64/kubectl
```

komutu çalıştırıldıktan sonra ikili çalıştırılabilir (binary executable) hale getirilmektedir. Sonrasında ise bu ikili /usr/local/bin/kubectl dizinine eklenmektedir.

```
$ chmod +x ./kubectl ,
$ sudo mv ./kubectl /usr/local/bin/kubectl
```

Minikube indirmek için gerekli komutlar aşağıdaki gibidir ("Kubernetes Playground," 2018):

```
$ curl -Lo minikube
https://storage.googleapis.com/minikube/releases
```

```
/v0.31.0/minikubelinux-amd64 && chmod +x minikube &&
sudo cp minikube /usr/local/bin/ && rm minikube
```

 şeklindedir. Bu komut ile minikube yerel makineye kurulduktan sonra her yerden erişilebilir şekilde bin klasörüne eklenmektedir. Minikube kurulduktan sonra versiyonunu öğrenmek için minikube version komutunu yazmak yeterlidir.

Bu çalışmada Kubernetes kümeleme aracının kurulumu için gerekli altyapılar kurulduktan sonra dağıtımı yapılacak servisler için minikube üzerinde kubernetes küme oluşturma uygulaması, dağıtım oluşturma, güncelleme, yapılan bir dağıtımı geri alma, bir dağıtımda

yapılan deęişiklikleri kontrol etme, daęıtımı ölçekleme işlemleri deneylerle açıklanacaktır. Ayrıca Kubernetes ile bölme oluşturmak için gerekli kubectl komutu, Vagrant kullanarak çok düęümlü Kubernetes küme oluşturma, aę başlatma, servis ölçekleme ve Vagrantfile kullanımı gibi ileri düzey konular Linux sanal makine üzerinde gösterilmektedir.

Kubernetes Kümesi Oluşturma: Minikube Kullanımı

Minikube başlatmak için \$minikube start komutu kullanılmaktadır. Minikube başlatıldıktan sonra yerel makinedeki Kubernetes istemci (client) kubectl bir mini-kube içeriğine sahip olacak ve otomatik olarak bu içerięi çalıştıracaktır. Böylece Kubernetes kümesinde bulunan düęümlerin bilgileri komut satırında \$kubectl get nodes komutu ile görülebilmektedir. Yerel minikube'e örnek bir Kubernetes daęıtımı \$kubectl run komutu ile Şekil 1'deki gibi yapılmaktadır. Bu daęıtımı harici bir aęa aktarmak için \$kubectl expose deployment hello-minikube --type=NodePort komutu kullanılmaktadır. Daha sonra bu daęıtımın bölmelerini listelemek için \$kubectl get pod komutu kullanılmaktadır. Örnek hello-minikube adlı servisin bilgilerine erişmek için curl \$(minikube service hello-minikube -url) komutu kullanılmaktadır. Oluşturulan servisi ve daęıtımı silmek için ise Şekil 2'de görüldüğü gibi kubectl delete deployment hello-minikube komutu kullanılmaktadır. Son olarak minikube stop komutu ile minikube sonlandırılmaktadır.

```

isil@isil-Veriton-M4640G ~ $ kubectl run hello-minikube --image=k8s.gcr.io/echoserver:1.10 --port=8080
kubectl run --generator=deployment/apps.v1 is DEPRECATED and will be removed in a future version. Use kubectl
run --generator=run-pod/v1 or kubectl create instead.
deployment.apps/hello-minikube created
isil@isil-Veriton-M4640G ~ $ kubectl expose deployment hello-minikube --type=NodePort
service/hello-minikube exposed
isil@isil-Veriton-M4640G ~ $ kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
hello-minikube-7c77b68cff-ls6xm    1/1     Running   0           13s
isil@isil-Veriton-M4640G ~ $ curl $(minikube service hello-minikube --url)

Hostname: hello-minikube-7c77b68cff-ls6xm
Pod Information:
  -no pod information available-
Server values:
  server_version=nginx: 1.13.3 - lua: 10008
Request Information:
  client address=172.17.0.1
  method=GET
  real path=/
  query=
  request_version=1.1
  request_scheme=http
  request_uri=http://192.168.99.102:8080/

```

Şekil 1. Yerel minikube ile örnek Kubernetes

```

isil@isil-Veriton-M4640G ~ $ clear
isil@isil-Veriton-M4640G ~ $ kubectl create -f https://k8s.io/examples/controllers/nginx-deployment.yaml
deployment.apps/nginx-deployment created
isil@isil-Veriton-M4640G ~ $ kubectl get deployments
NAME                DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment    3          3          3             0           8s
isil@isil-Veriton-M4640G ~ $ kubectl get rs
NAME                DESIRED   CURRENT   READY   AGE
nginx-deployment-86d59dd769    3          3          3       1m
isil@isil-Veriton-M4640G ~ $ kubectl get pods --show-labels
NAME                                READY   STATUS    RESTARTS   AGE   LABELS
nginx-deployment-86d59dd769-88j65    1/1     Running   0           1m    app=nginx,pod-template-hash=4281588325
nginx-deployment-86d59dd769-9k7g2    1/1     Running   0           1m    app=nginx,pod-template-hash=4281588325
nginx-deployment-86d59dd769-cvqsv    1/1     Running   0           1m    app=nginx,pod-template-hash=4281588325
isil@isil-Veriton-M4640G ~ $

```

Şekil 2. Yerel minikube ile servis bilgilerine erişme ve minikube sonlandırma

Minikube üzerinde Kubernetes Küme Uygulaması

İki kubectl komutu ile Minikube üzerinde Ghost mikro blog platformu çalıştırılmaktadır:

```
$ kubectl run ghost --image=ghost:0.9
```

```
$ kubectl expose deployments ghost --port=2368 --type=NodePort
```

Bu komutlardan sonra \$kubectl get pods komutu ile küme içerisinde oluşturulan ghost podunun durumu "running" olduktan

sonra, çalışıyor olduğundan emin olunur ve minikube service ghost komutu ile tarayıcı üzerinden otomatik olarak Ghost'a kubectl komutu ile aşağıdaki gibi erişilir:

```
$ kubectl get pods
```

```
NAME READY STATUS RESTARTS AGE
```

```
ghost-69f785b66c-s2wfg 1/1 Running 1 1d $ minikube service ghost
```

Dağıtım Oluşturma

Dağıtım ortamı oluşturma senaryosunda bir docker imajı kullanarak Tomcat App Sunucu dağıtılacaktır. Dağıtım için sırası ile yapılacak işlemler:

- Dağıtımı tanımla
- Servislerini oluşturma
- Sunucuyu Kubernetes kümesine dağıtma

Senaryoda en basit dağıtım yöntemi olarak tek bölme kullanılmaktadır. Tek bölmenin avantajı, herhangi bir servisin diğer bir servisten ayrımı olmaması, fazlalık (redundancy) olmaması ve tek bir bölmenin Kubernetes'e dağıtımının kolay olmasıdır. Servis olarak bölmeyi dış dünyaya aktarmak için kubectl expose komutu kullanılmaktadır. Kubernetes resmi sayfasında bulunan nginx-deployment.yaml dosyasında belirlenen dağıtımı oluşturmak için Şekil 3'te belirtilen adımlar takip edilerek ReplicaSet adı verilen kopyalar kubectl get rs komutu ile görülmektedir.

```
Server values:
  server_version=nginx: 1.13.3 - lua: 10008

Request Information:
  client_address=172.17.0.1
  method=GET
  real_path=/
  query=
  request_version=1.1
  request_scheme=http
  request_uri=http://192.168.99.102:8080/

Request Headers:
  accept=/*/*
  host=192.168.99.102:31407
  user-agent=curl/7.58.0

Request Body:
  -no body in request-

isil@isil-Veriton-M4640G ~ $ kubectl delete services hello-minikube
service "hello-minikube" deleted
isil@isil-Veriton-M4640G ~ $ kubectl delete deployment hello-minikube
deployment.extensions "hello-minikube" deleted
isil@isil-Veriton-M4640G ~ $ minikube stop
Stopping local Kubernetes cluster...
Machine stopped.
isil@isil-Veriton-M4640G ~ $
```

Şekil 3. nginx-deployment.yaml dosyasının oluşturulması

Dağıtım Güncelleme

nginx:1.7.9 imajı yerine nginx:1.9.1 gibi üst versiyon imajını kullanmak için nginx bölmeler güncellenmek istenirse;

\$ kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.9.1 --record deployment.apps/nginx-deployment image updated komutu ya da kubectl edit komutu ile ekrana yansıyan deployment.yaml dosyasında spec.template.spec.containers[0].image kısmı nginx:1.7.9 'dan nginx:1.9.1'a dönüştürülmektedir.

Rollout durumunu görmek için Şekil 4'deki gibi kubectl rollout status deployment.v1. komutu kullanılmaktadır. Rollout durumu bir uygulamanın yeni bir sürümünün dağıtımını başlatmak için kullanılmaktadır (Poulton, 2023). Rollout başarı ile

tamamlandıktan sonra yapılan dağıtımları `kubectl get deployments` komutu kullanılmaktadır. Dağıtım, bölmeler güncellenirken sadece belirli sayıda bölmelerin kapalı olabileceğinden emin olabilir.

```
isil@isil-Veriton-M4640G ~ $ kubectl rollout status deployment.v1.apps/nginx-deployment
deployment "nginx-deployment" successfully rolled out
isil@isil-Veriton-M4640G ~ $ kubectl get deployments
NAME                DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment    3         3         3            3           19m
isil@isil-Veriton-M4640G ~ $ kubectl get rs
NAME                DESIRED   CURRENT   READY   AGE
nginx-deployment-5d4fbdd945    0         0         0       12m
nginx-deployment-86d59dd769    0         0         0       20m
nginx-deployment-c4747d96c     3         3         3       13m
isil@isil-Veriton-M4640G ~ $ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
nginx-deployment-c4747d96c-2gzp4    1/1    Running   0          19m
nginx-deployment-c4747d96c-4759d    1/1    Running   0          19m
nginx-deployment-c4747d96c-cl4h8    1/1    Running   0          19m
isil@isil-Veriton-M4640G ~ $
```

Şekil 4. Rollout komutu

Varsayılan olarak istenen bölme sayısının en az %25'ini garanti etmektedir. (25% maksimum mevcut değildir) ("Deployments," 2023). Dağıtım ayrıca istenilen sayıda bölmenin üzerinde sadece belirli sayıda bölme oluşturulmasını sağlayabilir. Varsayılan olarak, istenen sayıda Pod'lerden % 25 daha fazla (%25 maksimum dalgalanma) olmasını sağlar ("Deployments," 2023). Örneğin Şekil 5'teki dağıtım detaylarına bakıldığında önce yeni bir bölme oluşturulur, daha sonra bazı eski bölmeler silinip yenileri oluşturulmaktadır.

```

islilgisil-Veriton-M4640G ~ $ kubectl describe deployments
Name:          nginx-deployment
Namespace:     default
CreationTimestamp: Tue, 18 Dec 2018 10:53:53 +0300
Labels:        app=nginx
Annotations:    deployment.kubernetes.io/revision: 8
                kubernetes.io/change-cause: kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.9.1 --record=
true
Selector:      app=nginx
Replicas:      3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
    nginx:
      Image:        nginx:1.9.1
      Port:         80/TCP
      Host Port:    0/TCP
      Environment:  <none>
      Mounts:       <none>
      Volumes:      <none>
Conditions:
  Type           Status  Reason
  ----           -
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets: <none>
NewReplicaSet:  nginx-deployment-c4747d96c (3/3 replicas created)
Events:
  Type           Reason          Age             From              Message
  ----           -
  Normal         ScalingReplicaSet   35m            deployment-controller   Scaled up replica set nginx-deployment-86d59dd769 to 3
  Normal         ScalingReplicaSet   28m            deployment-controller   Scaled down replica set nginx-deployment-86d59dd769 to 2
  Normal         ScalingReplicaSet   28m            deployment-controller   Scaled up replica set nginx-deployment-c4747d96c to 2
  Normal         ScalingReplicaSet   28m            deployment-controller   Scaled down replica set nginx-deployment-86d59dd769 to 1
  Normal         ScalingReplicaSet   24m (x3 over 28m)    deployment-controller   Scaled up replica set nginx-deployment-5d4fbd945 to 1
  Normal         ScalingReplicaSet   24m (x2 over 28m)    deployment-controller   Scaled down replica set nginx-deployment-c4747d96c to 2
  Normal         ScalingReplicaSet   24m (x2 over 28m)    deployment-controller   Scaled up replica set nginx-deployment-c4747d96c to 1
  Normal         ScalingReplicaSet   24m (x3 over 28m)    deployment-controller   Scaled up replica set nginx-deployment-c4747d96c to 3
  Normal         ScalingReplicaSet   24m (x8 over 28m)    deployment-controller   (combined from similar events): Scaled down replica set nginx-deployment-5d4fbd945 to 1
  Normal         ScalingReplicaSet   24m (x3 over 28m)    deployment-controller   Scaled down replica set nginx-deployment-5d4fbd945 to 0
islilgisil-Veriton-M4640G ~ $

```

Şekil 5. Yapılan dağıtım detaylarının listelenmesi

Yeterli sayıda yeni bölme üretmeden eski bölmeleri silmez ve aynı şekilde yeterli sayıda eski bölmeleri etkisiz hale getirmeden de yeni bölmeler oluşturmamaktadır. Bu da bölmelerin en az ikisinin en fazla da dördünün mevcut olmasını garantilemektedir ("Deployments," 2023).

Dağıtım Ölçekleme

kubectl scale komutu ile dağıtım ölçeklenebilmekte, autoscale komutu ile de minimum ve maksimum sayıda bölmeler girilerek CPU kullanımına göre otomatik ölçeklendirme işlemi yapılmaktadır (Ridha & Suhatman, 2022). Şekil 6'da görüldüğü gibi on kopya ile bir dağıtım çalıştırıldığında kubectl get deploy komutu ile güncel ve uygun olan on adet kopya görülecektir.

```

deployment.apps/nginx-deployment scaled
isil@isil-Veriton-M4640G ~ $ kubectl autoscale deployment.v1.apps/nginx-deployment --min=10 --max=15 --
cpu-percent=80
horizontalpodautoscaler.autoscaling/nginx-deployment autoscaled
isil@isil-Veriton-M4640G ~ $ kubectl get deploy
NAME          DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment  10        10        10           10          1h
isil@isil-Veriton-M4640G ~ $ kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:sometag
deployment.apps/nginx-deployment image updated
isil@isil-Veriton-M4640G ~ $ kubectl get rs
NAME          DESIRED   CURRENT   READY   AGE
nginx-deployment-595696685f  0         0         0       35m
nginx-deployment-5d4fbdd945  0         0         0       1h
nginx-deployment-867fdd8856  5         5         0       6s
nginx-deployment-86d59dd769  0         0         0       1h
nginx-deployment-c4747d96c   8         8         8       1h
isil@isil-Veriton-M4640G ~ $ kubectl get deploy
NAME          DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment  10        13        5           8          1h

```

Şekil 6. Dağıtımın ölçeklendirilmesi

Kümenin içerisinde yeni bir görüntü kubectl set komutu ile güncellendiğinde, güncellenen imaj ReplicaSet nginx-deployment-867fdd8856 ile birlikte yeni bir rollout başlatır. Fakat maxUnavailable gereksiniminden dolayı bloke edilmektedir. En az 10 en fazla 15 ReplicaSet oluşturan otomatik ölçeklemeyi kaldırmak için kubectl delete hpa nginx-deployment komutu kullanılmaktadır. Çalışan bir dağıtımı bekletmek için Şekil 7’de görüldüğü gibi "kubectl rollout pause deployment.v1.apps/nginxdeployment" komutu kullanılmaktadır. Dağıtımın çalışması durdurulduktan sonra imaj güncellenebilir ve kaldığı yerden devam ettirilebilir. Devam etmek için ise "kubectl rollout resume deployment.v1.apps/nginx-deployment" komutu kullanılmaktadır. Dağıtım sırasında kullanılacak kaynaklar da "kubectl set resources deployment.v1.apps/nginxdeployment-c=nginx --limits=cpu=200m,memory=512Mi" formatında belirlenebilmektedir.

```

isil@isil-Veriton-M4640G ~ $ kubectl get deploy
NAME          DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment 3          3          3             3           20m
isil@isil-Veriton-M4640G ~ $ kubectl get rs
NAME          DESIRED   CURRENT   READY   AGE
nginx-deployment-86d59dd769 3          3          3       20m
isil@isil-Veriton-M4640G ~ $ kubectl rollout pause deployment.v1.apps/nginx-deployment
deployment.apps/nginx-deployment paused
isil@isil-Veriton-M4640G ~ $ kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.9.1
deployment.apps/nginx-deployment image updated
isil@isil-Veriton-M4640G ~ $ kubectl rollout history deployment.v1.apps/nginx-deployment
deployment.apps/nginx-deployment
REVISION      CHANGE-CAUSE
1              <none>
isil@isil-Veriton-M4640G ~ $ kubectl get rs
NAME          DESIRED   CURRENT   READY   AGE
nginx-deployment-86d59dd769 3          3          3       21m
isil@isil-Veriton-M4640G ~ $ kubectl set resources deployment.v1.apps/nginx-deployment -c=nginx --limit
s=cpu=200m,memory=512Mi
deployment.apps/nginx-deployment resource requirements updated
isil@isil-Veriton-M4640G ~ $ kubectl rollout resume deployment.v1.apps/nginx-deployment
deployment.apps/nginx-deployment resumed
isil@isil-Veriton-M4640G ~ $ kubectl get rs -w
NAME          DESIRED   CURRENT   READY   AGE
nginx-deployment-775867b859 3          3          3       5s
nginx-deployment-86d59dd769 0          0          0       21m

```

Şekil 7. Çalışan dağıtımın bekletilip değiştirilmesi ve kaldığı yerden devam ettirilmesi

Birincil Komut Satırı Erişim Aracı: kubectl

kubectl neredeyse her Kubernetes'e erişim sağlamaktadır ve birincil komut satırı erişim aracıdır. Kubectl ile Kubernetes podları hakkında bilgi alma, parametre tanımlama, verilen bir dağıtımı bağlantı noktasına çıkarma, kaynak etiketlerinin tanımlanması, küme üzerinde belirlenen imajı çalıştırma gibi temel konteyner işlemlerini yerine getirmektedir (Gatev & Gatev, 2021). Aşağıdaki komut satırları temel kubectl komutlarını ifade etmektedir:

kubectl get pods: Tüm isim alanlarındaki bölmeleri listeler. Bölme ismi, bölmenin kaç adet örneğinin çalıştığı ve hazır olup olmadığı, hangilerinin yeniden çalıştığı ve yaşları gibi bilgileri listelemektedir.

kubectl describe pods [pod name]: Tüm bölmeler ya da parametre olarak belirtilmiş bölme için detaylı bilgi listelemektedir.

kubectl expose <type name> <identifier/name> [--port=external port] [target-port=container-port [--type=service-

type]: Verilen bir dağıtım, bölme veya başka bir kaynağı TCP ya da UDP olmak üzere bir bağlantı noktasına çıkarma (Şekil 8)

```
isil@isil-Veriton-M4640G ~ $ kubectl expose deployment nginx-deployment --type=NodePort
service/nginx-deployment exposed
isil@isil-Veriton-M4640G ~ $ kubectl port-forward nginx-deployment-775867b859-pxx2c 5000:6000
Forwarding from 127.0.0.1:5000 -> 6000
Forwarding from [::1]:5000 -> 6000
isil@isil-Veriton-M4640G ~ $ kubectl attach nginx-deployment-775867b859-pxx2c
Defaulting container name to nginx.
Use 'kubectl describe pod/ -n default' to see all of the containers in this pod.
If you don't see a command prompt, try pressing enter.
^C
isil@isil-Veriton-M4640G ~ $ kubectl exec -it nginx-deployment-775867b859-pxx2c bash
root@nginx-deployment-775867b859-pxx2c:/# whoami
root
root@nginx-deployment-775867b859-pxx2c:/#
```

Şekil 8. Kaynağı bağlantı noktasına çıkarma

kubectl port-forward <pod name> [LOCAL_PORT:]REMOTE_PORT]:

Bir ya da daha fazla yerel bağlantı noktasını bir bölmeye yönlendirir.

kubectl attach <pod name> -c <container>: Var olan bir konteyner içerisinde çalışan bir süreci ekler.

kubectl exec [-it] <pod name> [-c CONTAINER] -- COMMAND [args...]: Konteyner içerisinde komut çalıştırmayı sağlar.

kubectl label [--overwrite] <type> KEY_1=VAL_1...: Bir kaynak üzerinde etiketleri günceller.

kubectl run <name> --image=image: Küme üzerinde belirlenen imajı çalıştırma (Şekil 9)

```

isil@isil-Veriton-M4640G ~ $ kubectl run hazelcast1 --image=hazelcast --port=5701
kubectl run --generator=deployment/apps.v1 is DEPRECATED and will be removed in a future version. Use kubectl run --generator=run-pod/v1 or kubectl create instead.
deployment.apps/hazelcast1 created
isil@isil-Veriton-M4640G ~ $ kubectl describe pod
Name:          hazelcast-96dd5fc88-knlN4
Namespace:     default
Node:          minikube/10.0.2.15
Start Time:    Tue, 18 Dec 2018 16:24:20 +0300
Labels:        pod-template-hash=528819744
               run=hazelcast
Annotations:   <none>
Status:        Pending
IP:            172.17.0.4
Controlled By: ReplicaSet/hazelcast-96dd5fc88
Containers:
  hazelcast:
    Container ID:
    Image:        hazelcast
    Image ID:
    Port:         5701/TCP
    Host Port:    0/TCP
    State:        Waiting
      Reason:     ImagePullBackOff
    Ready:        False
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-5mvpj (ro)

```

Şekil 9. kubectl run ile küme üzerindeki imajın çalıştırılması

Vagrant Kullanarak Çok Düğümlü Kubernetes Küme Oluşturma

Çok düğümlü bir Kubernetes küme kurmak için yönetici (master) ve işçi (worker) düğümlerin özelliklerini içeren yml dosyalarına ve bu dosyaların konfigürasyonunu içeren Vagrantfile dosyasına ihtiyaç duyulmaktadır (Sayfan, 2018). Bu çalışmada yönetici, işçi ve Vagrantfile dosyalarının bulunduğu bir github klasörü ("Kubernetes Playground," 2018), git clone komutu ile ana makinenin bulunduğu çalışma dizinine kopyalanarak Şekil 10'da görüldüğü gibi Vagrantfile dosyasının bulunduğu dizine gidilip "vagrant up" komutu çalıştırılmaktadır. Bu komut sonrasında üç adet makine VirtualBox'da çalışır hale gelecektir. VirtualBox'da makinelerin çalıştığını görmek için komut satırına "virtualbox" yazılmaktadır.

```

isil@isil-Veriton-M4640G ~ $ cd gentfg-kubernetes/
isil@isil-Veriton-M4640G ~/gentfg-kubernetes $ cd installation/
isil@isil-Veriton-M4640G ~/gentfg-kubernetes/installation $ cd multinode-cluster
/
isil@isil-Veriton-M4640G ~/gentfg-kubernetes/installation/multinode-cluster $ cd
vagrant-ubuntu-k8s/
isil@isil-Veriton-M4640G ~/gentfg-kubernetes/installation/multinode-cluster/vagr
ant-ubuntu-k8s $ vagrant up
Bringing machine 'k8s1' up with 'virtualbox' provider...
Bringing machine 'k8s2' up with 'virtualbox' provider...
Bringing machine 'k8s3' up with 'virtualbox' provider...
==> k8s1: You assigned a static IP ending in ".1" to this machine.
==> k8s1: This is very often used by the router and can cause the
==> k8s1: network to not work properly. If the network doesn't work
==> k8s1: properly, try changing this IP.
==> k8s1: You assigned a static IP ending in ".1" to this machine.
==> k8s1: This is very often used by the router and can cause the
==> k8s1: network to not work properly. If the network doesn't work
==> k8s1: properly, try changing this IP.
==> k8s1: Checking if box 'ubuntu/xenial64' is up to date...
==> k8s1: A newer version of the box 'ubuntu/xenial64' for provider 'virtualbox'
is
==> k8s1: available! You currently have version '20181129.0.0'. The latest is ve
rsion

```

Şekil 10. vagrant up komutu ile yüklenen düğümler

Her makine, internete erişebileceği bir NAT-ed ağ arayüzüne ve 172.42.42.0/24 alt ağındaki özel bir ağ arayüzüne sahip olacaktır. Özel ağ küme içi iletişim için kullanılmaktadır. Küme büyüdükçe, küme yöneticisi (k8s1) bir kubeadm init gerçekleştirecek ve küme işçileri bir kubeadmin join gerçekleştireceklerdir. Bu küme statik bir Kubernetes küme belirteci kullanmaktadır. Vagrant up işlemi tamamlandıktan sonra vagrant ssh k81 komutu ile küme yöneticisi çalıştırılmakta ve kubectl -n kube-system get po -o wide komutu ile küme yöneticisi üzerinde çalışan bölmeler (pods) Şekil 11'deki gibi görülmektedir.

```

[ERROR Port-10250]: Port 10250 is in use
[ERROR Port-2379]: Port 2379 is in use
[ERROR Port-2380]: Port 2380 is in use
[ERROR DirAvailable--var-lib-etcd]: /var/lib/etcd is not empty
[preflight] If you know what you are doing, you can make a check non-fatal with '--ignore-preflight-errors=...'
root@k8s1:/home/vagrant# kubectl get pods -n kube-system -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP            NODE   NOMINATED NODE   READINESS GATES
coredns-86c58d9df4.jhcl  0/1     ContainerCreating  0      52m    <none>        k8s2   <none>            <none>
coredns-96c58d9df4.wsmkl  0/1     ContainerCreating  0      52m    <none>        k8s2   <none>            <none>
etcd-k8s1      1/1     Running    0          51m    172.42.42.1   k8s1   <none>            <none>
kube-apiserver-k8s1  1/1     Running    0          51m    172.42.42.1   k8s1   <none>            <none>
kube-controller-manager-k8s1  1/1     Running    0          51m    172.42.42.1   k8s1   <none>            <none>
kube-proxy-6tgqg  1/1     Running    0          52m    172.42.42.1   k8s1   <none>            <none>
kube-proxy-m74zl  1/1     Running    0          47m    172.42.42.3   k8s3   <none>            <none>
kube-proxy-qwd2  1/1     Running    0          49m    172.42.42.2   k8s2   <none>            <none>
kube-scheduler-k8s1  1/1     Running    0          51m    172.42.42.1   k8s1   <none>            <none>
root@k8s1:/home/vagrant# start-weave
serviceaccount/weave-net created
clusterrole.rbac.authorization.k8s.io/weave-net created
clusterrolebinding.rbac.authorization.k8s.io/weave-net created
role.rbac.authorization.k8s.io/weave-net created
rolebinding.rbac.authorization.k8s.io/weave-net created
daemonset.extensions/weave-net created
root@k8s1:/home/vagrant#

```

Şekil 11. Küme yöneticisi (k8s1) üzerinde çalışan bölmeler

Ağ Başlatma

Kümeleme ağının belirtilmesi otomatik değildir ve vagrant up komutunun tamamlanmasından sonra gerçekleştirilmelidir. Ağ başlatmak için bir komut dosyası çalıştırılarak küme yöneticisine (k8s1) / usr / local / bin / start-weave olarak yüklenmektedir. Ağ başlatıldıktan sonra Şekil 12’de görüldüğü gibi weave-net adında her bir düğüm için üç farklı bölme listelenmektedir.

Servis/Dağıtım Başlatma

Git deposunda kayıtlı olan service.yml ve deployment.yml dosyaları k8s1 yönetici düğüm üzerinde bulunmaktadır. “service” ve “deployment” dosyalarını çalıştırmak için "kubectl create -f /vagrant/service.yml -f /vagrant/deployment.yml" komutu çalıştırıldıktan sonra Şekil 12’de görüldüğü gibi "kubectl get service" komutu ile bu düğüm üzerinde çalışan servisler listelenmektedir. Aynı şekilde "kubectl get deployment" komutu ile de deployment.yml dosyasında belirtilen kopya sayısı kadar dağıtım bilgisi listelenmektedir.

```
root@k8s1:/home/vagrant# kubectl -n kube-system get po -o wide
NAME                                READY    STATUS      RESTARTS   AGE    IP              NODE    NOMINATED NODE    READINESS GATES
coredns-86c58d9df4-bnv6w           0/1     ContainerCreating    0         3m52s    <none>         k8s1    <none>             <none>
coredns-86c58d9df4-mtmzg           0/1     ContainerCreating    0         3m52s    <none>         k8s1    <none>             <none>
etcd-k8s1                           1/1     Running          0          61m     172.42.42.1    k8s1    <none>             <none>
kube-apiserver-k8s1                 1/1     Running          0          61m     172.42.42.1    k8s1    <none>             <none>
kube-controller-manager-k8s1        1/1     Running          0          61m     172.42.42.1    k8s1    <none>             <none>
kube-proxy-6tggg                    1/1     Running          0          62m     172.42.42.1    k8s1    <none>             <none>
kube-proxy-m74zl                    1/1     Running          0          57m     172.42.42.3    k8s3    <none>             <none>
kube-proxy-qmwd2                    1/1     Running          0          60m     172.42.42.2    k8s2    <none>             <none>
kube-scheduler-k8s1                 1/1     Running          0          61m     172.42.42.1    k8s1    <none>             <none>
weave-net-4b8h5                     1/2     Error            6          10m     172.42.42.3    k8s3    <none>             <none>
weave-net-bn46z                     1/2     Running          6          10m     172.42.42.2    k8s2    <none>             <none>
weave-net-sz8tt                     2/2     Running          0          10m     172.42.42.1    k8s1    <none>             <none>

root@k8s1:/home/vagrant# kubectl get service
NAME                TYPE        CLUSTER-IP      EXTERNAL-IP    PORT(S)    AGE
hello-service       ClusterIP    10.103.126.70    <none>         80/TCP     9m34s
kubernetes           ClusterIP    10.96.0.1        <none>         443/TCP     63m

root@k8s1:/home/vagrant# kubectl get deployment
NAME                READY    UP-TO-DATE    AVAILABLE    AGE
hello-deployment    0/3      3              0            9m46s

root@k8s1:/home/vagrant# kubectl scale deployment hello-deployment --replicas=1
deployment.extensions/hello-deployment scaled
root@k8s1:/home/vagrant# kubectl get deployment
NAME                READY    UP-TO-DATE    AVAILABLE    AGE
hello-deployment    0/1      1              0            10m

root@k8s1:/home/vagrant# kubectl scale deployment hello-deployment --replicas=5
deployment.extensions/hello-deployment scaled
root@k8s1:/home/vagrant# kubectl get deployment
NAME                READY    UP-TO-DATE    AVAILABLE    AGE
hello-deployment    0/5      5              0            10m

root@k8s1:/home/vagrant#
```

Şekil 12. kubectl scale komutu kullanımı

Servis Ölçekleme

Başlangıç hello-deployment dağıtımında 3 kopya bulunmaktadır. Bu kopya sayısını değiştirmek ve aşağı ölçeklemek için "kubectl scale deployment hello-deployment --replicas=1" komutu kullanılarak kopya sayısı bire indirilebilmektedir. Benzer şekilde "kubectl scale deployment hello-deployment --replicas=5" komutu ile kopya sayısı yukarı ölçeklenerek kopya sayısı 5'e çıkarılmaktadır.

Vagrantfile oluşturularak Kubernetes Küme Oluşturma

Üçüncü bölümde kullanıcı github klasöründe ("Kubernetes Playground," 2018) yer alan Vagrantfile kullanılarak Kubernetes küme oluşturulmuştur. Bu bölümde ise geliştirici Vagrantfile dosyasını kendisi oluşturup her bir düğüm için konfigürasyonlarını yapmaktadır. Öncelikle Vagrantfile dosyasına bir tanesi ana, diğer ikisi de işçi olmak üzere ubuntu/xenial imajı kullanılarak üç düğüm oluşturulmaktadır. Şekil 13'te geliştirici tarafından oluşturulan Vagrantfile dosyasının içeriği görülmektedir.

```
Vagrant.configure(2) do |config|

  (1..3).each do |i|
    config.vm.define "k8s-c3-n#{i}" do |s|
      s.ssh.forward_agent = true
      s.vm.box = "ubuntu/xenial64"
      s.vm.hostname = "k8s-c3-n#{i}"
      s.vm.network "private_network", ip: "192.168.11.10#{i}", netmask:
"255.255.255.0",
        auto_config: true,
        virtualbox__intnet: "k8s-net"
      s.vm.provider "virtualbox" do |v|
        v.name = "k8s-c3-n#{i}"
        v.memory = 1048
        v.gui = false
      end
    end
  end

  if Vagrant.has_plugin?("vagrant-cachier")
    config.cache.scope = :box
  end
end
```

Şekil 13. Oluşturulan Vagrantfile içeriği

Bu dosya oluşturulduktan sonra dosyanın bulunduğu dizine gelinerek "vagrant up" komutu çalıştırılmaktadır. Bu komutun çalışması tamamlandıktan sonra sanal makinede

- k8s-c3-n1 (ip: 192.168.11.101)
- k8s-c3-n2 (ip: 192.168.11.102)
- k8s-c3-n3 (ip: 192.168.11.103)

isimli üç düğüm oluşmaktadır. *vagrant status* komutu ile düğümlerin çalıştığı teyit edilmektedir. Düğümler oluşturulduktan sonra her üç düğümde de aşağıdaki konfigürasyonlar yapılmaktadır.

Google apt anahtar indirme ve ekleme için:

```
wget https://packages.cloud.google.com/apt/doc/apt-key.gpg
```

```
sudo apt-key add apt-key.gpg
```

```
sudo apt-add-repository "deb http://apt.kubernetes.io/  
kubernetes-xenial main"
```

```
sudo apt-get update
```

kubeadm indirmek için:

```
sudo apt-get install kubeadm -y
```

Docker indirmek ve servis sağlamak için:

```
sudo apt-get install docker.io -y
```

```
sudo systemctl enable docker
```

```
sudo systemctl start docker
```

k8s-c3-n1 yönetici düğüm üzerinde kubelet servis sağlamak için:

```
sudo systemctl enable kubelet
```

```
sudo systemctl start kubelet
```

komutları kullanılmaktadır. Tüm bu işlemler yapıldıktan sonra "sudo kubeadm init--apiserver-advertise-address=192.168.11.101 --pod-network-cidr=10.244.0.0/16" komutu ile küme

başlatılmaktadır. Buradaki apiserver-advertise-address, kullanıcının ana bilgisayarlarının birden fazla ağ kartına sahip olması durumunda api sunucunun spesifik bir IP adrese bağlanmasını sağlamaktadır. Seçilen bu ağ üzerinde tüm düğümlerin iletişim halinde olması göz önünde bulundurulmalıdır. Pod network CIDR ise bölmelerin kullanması istenilen ağı göstermektedir (Luksa, 2017). Bu çalışmada geliştirici tarafından 10.244.0.0/16 adresi belirlenmiştir. kubeadm init komutu sonrasında Şekil 14'teki çıktı ekrana gelmektedir.

```
Your Kubernetes master has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

You can now join any number of machines by running the following on each node
as root:

kubeadm join 192.168.11.101:6443 --token 2p7tw0.0gg6e5lpe0gmxn9z --discovery-t
oken-ca-cert-hash sha256:b90049445ec7e6e6e86c31e0bf7c4d82db0524c204a55988ca8f8f9
647002159

vagrant@k8s-c3-n1:~$ mkdir -p $HOME/.kube
vagrant@k8s-c3-n1:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
vagrant@k8s-c3-n1:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
vagrant@k8s-c3-n1:~$
```

Şekil 14. kubeadm init komutu ile küme başlatma

Bu işlemten sonra yönetici düğüm olarak atanması istenilen k8s-c3-n1 düğümü "vagrant ssh k8s-c3-n1" komutu ile çalıştırılarak aşağıdaki komut satırları girilmektedir ve düğüm yönetici (master) yapılmaktadır.

```
mkdir -p $HOME/.kube
```

```
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
```

Daha sonra yönetici düğüm üzerinde "sudo kubectl apply -f https://raw.githubusercontent.com/flannel/flannel/master/Documentation/kube-flannel.yml" komutu çalıştırılarak bölme ağı (pod network) eklenmektedir. Yönetici düğüm yapılandırıldıktan sonra her bir işçi düğüm üzerinde yönetici

düğümde yazılan kubeadm init komutundan elde edilen "kubeadm join" komutu çalıştırılmaktadır. Bu komut sonrasında kümeye düğümün eklendiği Şekil 15'te görülmektedir.

```
lidades against pinned roots, will use API Server "192.168.11.101:6443"
[discovery] Successfully established connection with API Server "192.168.11.101:6443"
[join] Reading configuration from the cluster...
[join] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -oyaml'
[kubelet] Downloading configuration for the kubelet from the "kubelet-config-1.13" ConfigMap in the kube-system namespace
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Activating the kubelet service
[tlsbootstrap] Waiting for the kubelet to perform the TLS Bootstrap...
[patchnode] Uploading the CRI Socket information "/var/run/dockershim.sock" to the Node API object "k8s-c3-n3" as an annotation

This node has joined the cluster:
* Certificate signing request was sent to apiservert and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the master to see this node join the cluster.

vagrant@k8s-c3-n3:~$
```

Şekil 15. kubeadm join komutu sonrasında işçi düğümlerin oluşması

İşçi düğümlerde belirtildiği gibi yönetici düğüm üzerinde yazılan "kubectl get nodes" komutu ile ana bilgisayar üzerinde çalışan ve konfigürasyonu yapılan üç düğüm de hazır(ready) konumunda listelenmektedir. Düğümlerin Ready konumuna gelmesi yaklaşık 30 saniye sürmektedir. Tüm bu işlemlerden sonra Kubernetes küme oluşturulmuş olup örnek bir uygulama, düğümlere dağıtılmaktadır. Şekil 16'da görüldüğü gibi "kubectl run" komutu ile "httpd" web sunucusu dağıtılmaktadır.


```

vagrant@k8s-c3-n1:~$ kubectl run --image=httpd:2.4 httpd-01 --port=80 --env="DOMAIN=cluster"
kubectl run --generator=deployment/apps.v1 is DEPRECATED and will be removed in a future version. Use kubectl run --generator=run-pod/v1 or kubectl create instead.
deployment.apps/httpd-01 created
vagrant@k8s-c3-n1:~$ kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
httpd-01-588d99f97d-25s6c          0/1     ContainerCreating   0           9s
vagrant@k8s-c3-n1:~$ kubectl expose deployment httpd-01 --port=80 --name=httpd-01-http
service/httpd-01-http exposed
vagrant@k8s-c3-n1:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
httpd-01-588d99f97d-25s6c          1/1     Running   0           32s

```

Şekil 16. httpd web sunucusunun dağıtımı

Bu bölme için istenilen isim verilebilir. Bu çalışmada bölme ismi http-01 verilmiştir. Bu bir web sunucusu olduğundan dolayı sunucuya erişilebilmesi için port numarasına expose komutu ile 80 atanmaktadır. httpd-01-http isimli servis oluşturularak Şekil 16'da görüldüğü gibi port atama işlemi yapılmaktadır. Sonrasında kubectl get pods komutu ile oluşturulan bölmenin ismi alınıp kubectl describe pod httpd-01-588d99f97d-25s6c komutu ile bölme hakkındaki detaylar listelenmekte ve imajın k8s-c3-n2 isimli düğümde çalıştığı görülmektedir.

```

vagrant@k8s-c3-n1:~$ kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
k8s-c3-n1           Ready     master   47m   v1.13.1
k8s-c3-n2           Ready     <none>   20m   v1.13.1
k8s-c3-n3           Ready     <none>   28m   v1.13.1
vagrant@k8s-c3-n1:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
httpd-01-588d99f97d-25s6c          1/1     Running   0           18m
vagrant@k8s-c3-n1:~$ kubectl get services
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
httpd-01-http       ClusterIP   10.106.60.110 <none>         80/TCP     17m
kubernetes           ClusterIP   10.96.0.1      <none>         443/TCP    47m
vagrant@k8s-c3-n1:~$ kubectl describe service httpd-01-http
Name:                httpd-01-http
Namespace:           default
Labels:               run=httpd-01
Annotations:          <none>
Selector:             run=httpd-01
Type:                 ClusterIP
IP:                   10.106.60.110
Port:                 <unset> 80/TCP
TargetPort:           80/TCP
Endpoints:            10.244.2.2:80
Session Affinity:     None
Events:               <none>
vagrant@k8s-c3-n1:~$

```

Şekil 17. Oluşturulan servis hakkında detaylı bilginin listelenmesi

Yönetici düğümüne geri dönülüp servisler listelendiğinde ve servis hakkında detaylı bilgi alındığında endPoint ve port numarası Şekil 17'deki gibi görülmektedir. Oluşturulan bu servisin işçi düğüm k8s-c3-n2'de çalıştığı bilindiğinden dolayı vagrant ssh komutu ile bu düğüm içerisinde wget -qO- http://10.244.2.2 komutu yazılarak yönetici düğümde görülen endpoint IP'si işçi düğümde çalıştırılmaktadır ve Şekil 18'de görüldüğü gibi uygulamanın çalıştığına dair html kodları listelenmektedir.

```
vagrant@k8s-c3-n1:~$ kubectl get nodes
NAME          STATUS    ROLES    AGE   VERSION
k8s-c3-n1     Ready     master   47m   v1.13.1
k8s-c3-n2     Ready     <none>   20m   v1.13.1
k8s-c3-n3     Ready     <none>   28m   v1.13.1
vagrant@k8s-c3-n1:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
httpd-01-588d99f97d-25s6c   1/1     Running   0           18m
vagrant@k8s-c3-n1:~$ kubectl get services
NAME          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
httpd-01-http ClusterIP    10.106.60.110   <none>        80/TCP     17m
kubernetes    ClusterIP    10.96.0.1       <none>        443/TCP    47m
vagrant@k8s-c3-n1:~$ kubectl describe service httpd-01-http
Name:         httpd-01-http
Namespace:    default
Labels:       run=httpd-01
Annotations:   <none>
Selector:     run=httpd-01
Type:         ClusterIP
IP:           10.106.60.110
Port:         <unset> 80/TCP
TargetPort:   80/TCP
Endpoints:    10.244.2.2:80
Session Affinity: None
Events:       <none>
vagrant@k8s-c3-n1:~$
```

Şekil 18. Servisin işçi düğüm üzerinde çalıştırılması

Sonuç

Bu çalışmada Kubernetes konteyner orkestrasyon aracının kurulumu ve çalıştırılması için gerekli altyapılar irdelenerek Kubernetes küme oluşturma, bölme oluşturma ve uygulamaların oluşturulan bölmelere dağıtımı bir Kubernetes küme üzerinde yapılabilecek dağıtım, bakım ve ölçekleme süreçleri detaylı bir şekilde incelenmiştir. Kubernetes aracında kullanılan kubectl,

kubeadm gibi temel komutlar ile birden fazla servisin küme yapısına sahip düğümler üzerinde nasıl çalıştırıldığı deneysel uygulamalarla gösterilmiştir. Bununla birlikte bir sanal makine aracı üzerinde Vagrant ile çok düğümlü küme oluşturma, spesifik bir konfigürasyon dosyası üzerinden servis, ağ ve dağıtım başlatma ve Vagrantfile oluşturma işlemleri Linux sanal makine üzerinde sunulmuştur.

KAYNAKÇA

Deployments. (2023). Retrieved from <https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>

Gatev, R., & Gatev, R. (2021). Getting up to speed with kubernetes. Introducing Distributed Application Runtime (Dapr) Simplifying Microservices Applications Development Through Proven and Reusable Patterns and Practices, 51-67.

Install Tools. (2023). Retrieved from <https://kubernetes.io/docs/tasks/tools/>

Kubernetes Playground. (2018). Retrieved from <https://github.com/ramakris/gentfg-kubernetes/tree/master/installation/multinode-cluster/vagrant-ubuntu-k8s>

Luksa, M. (2017). Kubernetes in action: Simon and Schuster.

Poulton, N. (2023). The kubernetes book: NIGEL POULTON LTD.

Ridha, M. A. F., & Suhatman, R. (2022). Perbandingan Kinerja Kubernetes Cluster dengan Virtualisasi KVM, Vagrant dan LXD. Jurnal Komputer Terapan, 8(1), 151-157.

Saito, H., Lee, H.-C. C., & Hsu, K.-J. C. (2018). Kubernetes Cookbook: Practical solutions to container orchestration: Packt Publishing Ltd.

Sayfan, G. (2018). Mastering Kubernetes: Master the art of container management by using the power of Kubernetes: Packt Publishing Ltd.

Shishkalov, I. (2022). Kubernetes Multiclustur Architectures. Saxon

BÖLÜM III

Gradyan Temelli Kenar Belirleme Yaklaşımları ile Görüntü Enterpolasyonu

Mahmut KILIÇASLAN¹

Giriş

Görüntüler dijital dünyada önemli bir rol oynamaktadır. Teknolojinin gelişmesi yüksek çözünürlüklü görüntülerin elde edilmesini beraberinde getirmiştir. Ancak ağ sistemlerinin yoğun kullanımı ve bant genişliği yetersizliği yüksek kalitedeki video, görüntü vb. içeriklerin iletiminde sorun haline gelmektedir. İlaveten düşük çözünürlüklü elde edilen görüntülerin iyileştirilmesi de diğer bir problemidir. Bu sebeple görüntü ölçeklendirme görüntü işleme alanında önemli bir çalışma alanıdır. Söz konusu problemlerin üstesinden gelmek adına görüntü enterpolasyon tekniklerine başvurulmaktadır. Görüntü enterpolasyonu, görüntü işlemenin önemli bir konusu olup havacılık, tıp, iletişim, meteoroloji, uzaktan

¹ Öğr. Gör. Dr., Ankara Üniversitesi

algılama ve askeri uygulamalarda yaygın olarak kullanılmaktadır (Jia & Huang, 2022; Karwowska & Wierzbicki, 2022). Görüntü enterpolasyonu düşük çözünürlüklü bir görüntüden yüksek çözünürlüklü elde etme süreci olarak tanımlanmaktadır. Başka bir ifade ile bilinen pikseller aracılığıyla, bilinmeyen piksellerin oluşturulmasıdır (İncetaş, 2023). Görüntü enterpolasyon teknikleri süper çözünürlüklü ve örnek bağımsız olarak 2 kategoriye ayrılmaktadır. Süper çözünürlük temelli yaklaşımlar düşük çözünürlüklü ve yüksek çözünürlüklü birçok örnek arasındaki ilişkileri öğrenmeye dayanmaktadır (Jin et al., 2021; Li et al., 2016; Ren et al., 2019; Sánchez-García et al., 2019; Wang et al., 2017). Örnek bağımsız yaklaşımlar ise bilinen pikseller üzerinde matematiksel işlemler ile enterpolasyon sürecini gerçekleştirmektedirler (Liu et al., 2020; Wei & Ma, 2013; Zhong et al., 2019). Örnek bağımsız yaklaşımlarda herhangi bir öğrenme aşaması olmadığından dolayı çok hızlıdır. En bilinen örneksiz yaklaşımlar en yakın, çift doğrusal ve çift kübik enterpolasyon yöntemleridir. Son zamanlarda yapılan enterpolasyon çalışmaları kenar tespitine dayalı teknikler olduğu görülmektedir. Genel olarak kenar tespitine dayalı çalışmalar kenar ve bölge piksellerine ayrı enterpolasyon teknikleri uygulayarak bilgi kaybını en aza indirmeyi amaçlamaktadır.

Literatürde kenar tespitine dayalı birçok enterpolasyon çalışması bulunmaktadır. 2013 yılında Wei ve Ma tarafından kontrast bilgisini görüntü enterpolasyon sürecine dahil eden enterpolasyon yöntemi önerilmiştir. Enterpolasyon altındaki görüntü göz önüne alındığında, kontrast kılavuzlu karar haritası oluşturulmaktadır. Söz konusu haritalar diyagonal piksellerin enterpolasyonu ve satır-sütun piksellerin enterpolasyonu için farklı açılarda işlem yapmaktadır. Geliştirilen teknik girdi görüntüsüne kenar algılama uygulandıktan sonra, algılanan her bir kenarın yakındaki kenar olmayan piksellerinin, bunların muhtemelen kenar pikselleri olarak yeniden sınıflandırılması için değerlendirilmesi esasına dayanmaktadır (Wei & Ma, 2013). Benzer şekilde kenar belirmeye dayalı diğer bir çalışmada PCI adı verilen yeni bir görüntü

enterpolasyon metodolojisi önerilmiştir. Düşük çözünürlüklü görüntüden bir tahmin modeline dayalı mimari geliştirilmiştir. Sonraki düzeltme aşamasında, tahmin edilen görüntünün kenar pikselleri belirlenmiş ve daha sonra görüntü kalitesinin daha da iyileştirilmesi için gerekli düzeltmeler yapılmıştır (Zhong et al., 2019). Gradyan temelli bir başka enterpolasyon çalışmasında “bölgesel gradyan tahminiyle görüntü enterpolasyonu” adı verilen doğrusal olmayan bir enterpolasyon yöntemi önerilmektedir. Söz konusu yaklaşım bir görüntünün kenar piksellerini ve kenar olmayan piksellerini daha etkili bir şekilde ayırt edebilmektedir. Bilinmeyen piksellerin gradyanını değerlendirirken görüntünün enterpolasyon etkisini arttırmak için bölgesel bir gradyan tahmin stratejisi kullanmaktadır (Jia & Huang, 2022). 2023 yılında öğrenme temelli olmayan sinir ağı tabanlı dikkat çeken bir yaklaşım bulunmaktadır. İncetaş bahsi geçen çalışmada kenar tespitine dayalı yeni bir enterpolasyon tekniği önermiştir. Geliştirilen yaklaşımda insan görsel sistemini başarılı bir şekilde simüle eden (İncetaş, 2022; İncetaş & Arslan, 2019; İNCETAŞ et al., 2022; İNCETAŞ & Murat) iğnecikli yapay sinir ağı modeli kenar belirleme için kullanılmıştır (İncetaş, 2023).

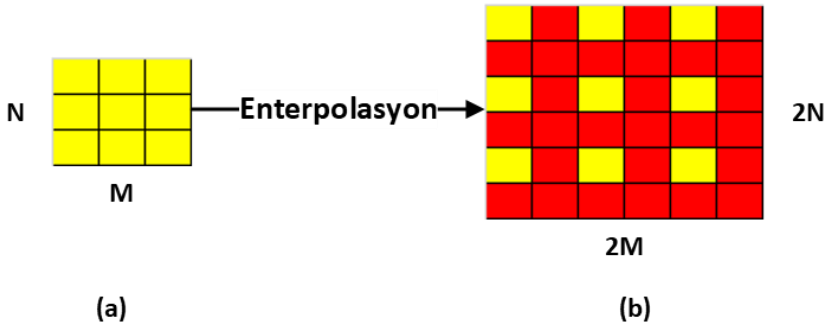
Bu çalışmada gradyan temelli kenar belirleme teknikleri ile görüntü enterpolasyonu gerçekleştirilmiştir. Çalışmada Sobel, Prewitt ve Roberts kenar operatörleri ile kenar pikseller tespit edilmiştir. Kenar belirleme işleminde söz konusu operatörlerin “nonthinging” paramateresi kullanılmıştır. Kenar belirleme işleminde yatay, dikey ve diyagonal eksenindeki tüm kenarlar işleme dahil edilmiştir. Ardından kenar olarak tanımlanan piksellere 1 boyutlu kübik enterpolasyon, geri kalan piksellere ise bikübik enterpolasyon uygulanmıştır. Deneyisel çalışmalar BSDS500 veri setinden seçilen 100 adet görüntü üzerinde gerçekleştirilmiştir. Son olarak değerlendirme kriteri olarak PSNR ve SSIM değerlerinden faydalanılarak, Sobel, Prewitt ve Roberts kenar belirleme operatörlerinin görüntü enterpolasyonunda ki başarıları karşılaştırılmıştır.

Görüntü Enterpolasyonu

Görüntü enterpolasyonu bilinen pikseller üzerinde gerçekleştirilen matematiksel işlemler aracılığıyla bilinmeyen pikselleri tahmin etme süreci olarak tanımlanmaktadır. $M \times N$ düşük çözünürlüklü görüntünün (G_D), enterpolasyon sonucunda $2M \times 2N$ çözünürlüklü bir görüntüye (G_Y) dönüştürülmesi amaçlanmaktadır. Bu durumda;

$$G_D(i, j) = G_Y(2i - 1, 2j - 1) \quad (1)$$

eşitliği yazılabilir. Bu denklemde G_D düşük çözünürlüklü görüntüyü temsil ederken, G_Y ise yüksek çözünürlüklü görüntüyü ifade etmektedir. Şekil 1 ise bir görüntü enterpolasyonunu simgelemektedir.

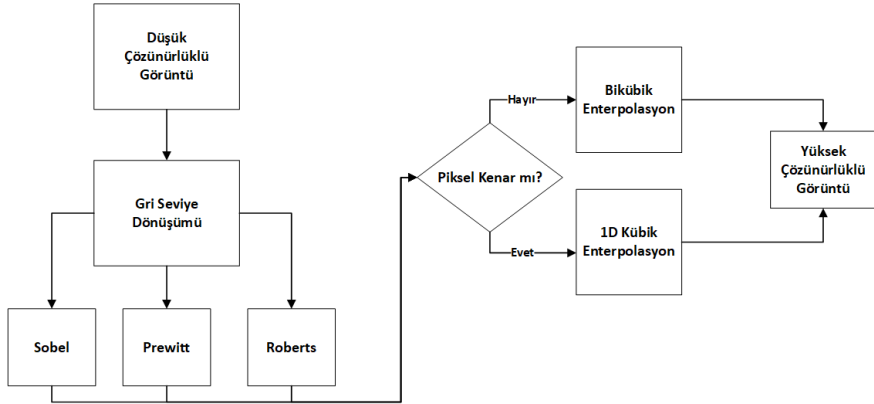


Şekil 1. Görüntü enterpolasyonu

Şekil 1a orijinal görüntüyü gösterirken, Şekil 1b enterpolasyon sonucu elde edilen çıktı görüntüyü ifade etmektedir. İlgili şekilde sarı renk olanlar bilinen piksel değerleri iken, kırmızı renktekiler ise enterpolasyon sonucunda tahmin edilmesi gereken pikselleri göstermektedir. Genel olarak kenar tabanlı enterpolasyon yaklaşımlarında, kenarlar ve yönleri tespit edildikten sonra ilk olarak diyagonal eksenindeki piksellerin değerleri hesaplanır. Ancak bu hesaplama düşük G_D ’de diyagonal yönlü bir kenar piksel bulunması bağlıdır. Bu durumda $G_D(2i, 2j)$ eksenindeki bilinmeyen pikseller, sarı renk ile temsil edilen bilinen piksel değerleri aracılığıyla

hesaplanmaktadır. Bahsedilen süreç enterpolasyon sürecinin ilk aşamasını oluşturmaktadır. Sonrasında ise yatay ve dikey yönlerdeki piksel değerleri işlemlere tabi tutulmaktadır. Eğer bir piksel $G_Y(2i, 2j+1)$ konumunda olması durumunda yatay yönlüde bir kenar bilgisi varsa, yatay yönlü komşu pikselleri kullanılarak hesaplama işlemi gerçekleştirilir. Benzer şekilde piksellerin $G_D(2i+1, 2j)$ konumunda olması dikey düzlemdeki komşu pikseller yardımıyla değerlerinin hesaplanacağı anlamına gelmektedir. Bir sonradaki aşamada ise kenar olmayan pikseller üzerinde ise genellikle biküçük enterpolasyon uygulanmaktadır. Böylece G_Y görüntüsündeki tüm piksel değerleri tespit edilmiş olur.

Bu çalışmada görüntü enterpolasyonu için kullanılan yöntem Şekil 2'de görselleştirilmiştir. Çalışmada kenar belirleme adımları olarak girdi görüntüye ayrı ayrı Sobel, Prewitt ve Roberts operatörleri uygulanmıştır. Kenar olarak tespit edilen piksellere 1D küçük enterpolasyon uygulanırken, kenar olmayan piksellere ise biküçük enterpolasyon işlemi yürütülmüştür.



Şekil 2. Uygulanan enterpolasyon yaklaşımı

Küçük görüntü enterpolasyonu, bir pikselin değerini belirlemek için çevresindeki piksellerin bilinen değerlerini kullanarak daha yüksek çözünürlükte veya daha düzgün bir görüntü oluşturmak için kullanılan bir enterpolasyon yöntemidir. Söz konusu teknik genellikle görüntü büyütme işlemlerinde kullanılmaktadır.

Kübik enterpolasyon, bir pikselin değerini hesaplamak için çevresindeki dört pikselin değerini kullanır. Bu dört piksel, x ve y koordinatlarının tamsayı kısımları arasında bulunur. Kübik enterpolasyon, piksel değerlerinin yanı sıra pikseller arasındaki uzaklıkları da dikkate almaktadır (Shi, J., & Reichenbach, S. E., 2006). Matematiksel olarak, kübik enterpolasyon şu şekilde ifade edilebilir:

$$f(x) = a(x - x_0)^3 + b(x - x_0)^2 + c(x - x_0) + d \quad (2)$$

burada $f(x)$ enterpolasyon sonucu elde edilen değeri temsil ederken, x enterpolasyon yapılacak noktanın x koordinatını gösterir. a, b, c, d katsayıları ise çevresindeki dört pikselin değerlerine ve konumlarına bağlı olarak belirlenmektedir. Kübik enterpolasyon, görüntüdeki pürüzlü kenarları yumuşatma ve daha doğal bir görüntü elde etme amacıyla kullanılmaktadır. Ancak, daha karmaşık matematiksel işlemleri gerektirdiği için hesaplama maliyeti biraz daha yüksektir. Bikübik enterpolasyon ise, bir pikselin değerini hesaplamak için çevresindeki 16 piksel değerini kullanmaktadır. Bu 16 piksel, x ve y koordinatlarının tamsayı kısımları arasında ve bu kısımların birer birim altında ve üstünde bulunur. Bikübik enterpolasyon, piksel değerlerinin yanı sıra pikseller arasındaki uzaklıkları da dikkate alır (Dengwen, 2010). Bikübik enterpolasyon şu şekilde ifade edilebilir:

$$f(x) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} \cdot (x - x_0)^i \cdot (y - y_0)^j \quad (3)$$

burada $f(x)$ enterpolasyon sonucu elde edilen değeri temsil eder ve x, enterpolasyon yapılacak noktanın x koordinatını, y ise y koordinatını gösterir. a_{ij} çevresindeki 16 pikselin değerlerine ve konumlarına bağlı olarak belirlenir. Bikübik enterpolasyon, daha karmaşık bir matematiksel model kullanarak daha düzgün ve doğal görüntüler elde etmeyi amaçlamaktadır. Ancak, bu daha karmaşık model, hesaplama maliyetini artırmaktadır.

Enterpolasyon çalışmalarında genellikle yukarıda bahsi geçen enterpolasyon tekniklerinin uygulanmasından ve kenar piksellerin belirlenmesinden önce, piksellerin yatay, dikey ya da diyagonal eksendeki bir kenara dâhil olup olmadığına bakılmaktadır. Enterpolasyon işleminde öncelikle diyagonal yöndeki kenar pikselleri için tahminler hesaplanmaktadır. Diyagonal eksen açısı tespitleri

$$\alpha(2i, 2j) \begin{cases} 45^\circ, F_{45}(i, j) > F_{135}((i, j)) \\ 135^\circ, F_{135}(i, j) > F_{45}((i, j)) \end{cases} \quad (4)$$

$F_{45}(i, j)$ ve $F_{135}(i, j)$ değerleri arasında fark yoksa Eşitlik 4'de olduğu gibi G_D görüntüsünün gri-seviye dönüşümünün çıktıları, $G_Y(2i, 2j)$ kenar olmayan piksel olarak kabul edilir ve herhangi bir atama yapılmaz. Fakat $F_{45}(i, j)$ daha büyükse $F_{135}(i, j)$ 'den daha büyükse, $\alpha(2i, 2j)$ 45 derece olarak atanır, aksi takdirde 135 derece olarak atanacaktır. Benzer işlemler bilinen yatay ve dikey komşulukların olduğu $G_Y(2i-1, 2j)$ ve $G_Y(2i, 2j-1)$ pikselleri için gerçekleştirilir. Yatay ve dikey yön tespitleri için;

$$\alpha(i, j-1) = \alpha(i-1, j) \begin{cases} 0^\circ, F_0(i, j) > F_{90}((i, j)) \\ 90^\circ, F_{90}(i, j) > F_0((i, j)) \end{cases} \quad (5)$$

şeklinde hesaplamalar yapılmaktadır. Eşitlik 5'te, gri seviyeli çıktılar olan $F_{90}(i, j)$ ve $F_0(i, j)$ değerleri birbirine eşitse, $G_D(2i-1, 2j)$ ve $G_D(2i, 2j-1)$ kenar olmayan pikseller olarak tanımlanır. Eğer $F_{90}(i, j)$, $F_0(i, j)$ 'den büyükse $\alpha(2i-1, 2j)$ ve $\alpha(2i, 2j-1)$ 90° , değilse 0° olarak atanır.

Gradyan Temelli Kenar Belirleme Teknikleri

Kenar belirleme görüntü işleme alanında önemli bir konudur. Literatür çalışmaları incelendiğinde birçok kenar belirleme yaklaşımı geliştirilmiştir. Bu başlık altında sıklıkla kullanılan kenar belirleme tekniklerinden gradyan tabanlı Sobel, Prewitt ve Roberts tekniklerinden bahsedilecektir.

Sobel kenar belirleme operatörü, bir görüntüdeki kenarları vurgulamak için kullanılan bir filtre veya operatördür. Genellikle dijital görüntü işleme ve görüntü analizi alanlarında kullanılır. Sobel operatörü, özellikle gradyan (türev) hesaplamak için kullanılır ve kenarların konumunu ve yoğunluğunu belirlemek için etkilidir. Sobel operatörü, görüntüdeki her pikselin çevresindeki pikselleri analiz eder ve bu piksellere ağırlıklar verir. Bu ağırlıklar, genellikle bir matris veya çekirdek kullanılarak belirlenir (Chaple et al., 2015; Kanopoulos et al., 1988). Sobel operatörünün x-yönündeki matrisleri Şekil 3'te gösterilmiştir.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Şekil 3. Sobel operatörü matrisleri

Yukarıda bahsi geçen matrisler giriş görüntüye ayrı ayrı uygulanabilmektedir. Böylece hem yatay hem de dikey yönde kenar piksel tespiti yapılabilmektedir. Diğer bir kenar belirleme yaklaşımı Prewitt operatörüdür. Görüntüdeki kenarları vurgulamak ve tespit etmek için kullanılan söz konusu filtre ya da operatör özellikle dijital görüntü işleme ve görüntü analizi uygulamalarında sıklıkla tercih edilen yaklaşımlardandır. Prewitt operatörü, görüntüdeki kenarları belirlemek için Sobel operatörüne benzer bir şekilde gradyan (türev) hesaplamak amacıyla tasarlanmıştır. Başka bir ifade ile Sobel operatörüne benzeyen ancak farklı bir çekirdek matrisi kullanan bir tekniktir (Chaple et al., 2015). Prewitt operatörünün kullandığı matrisler ise Şekil 4'te gösterilmiştir

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Şekil 4. Prewitt operatörü matrisleri

Bu matrisler, bir pikselin x ve y yönündeki türevlerini hesaplamak için kullanılır. Her bir pikselin türevi, çevresindeki piksellerin yoğunluklarına göre ağırlıklı bir toplam olarak hesaplanır. Bu türevler daha sonra gradyan vektörünü oluşturur. Prewitt operatörü, genellikle bir görüntüdeki dikey ve yatay kenarları vurgulamak için kullanılır. Yüksek gradyan değerlerine sahip pikseller, görüntüdeki kenarların bulunduğu yerleri temsil eder. Bu operatör, kenar tespiti, nesne algılama ve diğer birçok görüntü işleme görevinde kullanılır. Roberts kenar belirleme operatörü ise, bir görüntüdeki kenarları belirlemek için kullanılan basit bir gradyan filtresidir. Bu operatör, bir pikselin yanındaki piksellerle olan yoğunluk farklarını kullanarak kenarları tespit etmeye çalışır. Roberts operatörü, x ve y yönlerindeki türevleri hesaplamak için iki küçük matris veya çekirdek kullanır. Bu matrisler Şekil 5'te gösterilmektedir. İlgili matrisler, her bir pikselin x ve y yönündeki türevlerini yaklaşık olarak hesaplamak için kullanılır. Her bir türev, çevresindeki piksellerin yoğunluklarına göre ağırlıklı bir toplam olarak elde edilir. Roberts operatörü genellikle daha hafif ve daha basit bir kenar belirleme yöntemi olarak kullanılır. Ancak, daha büyük ve karmaşık görüntülerde diğer kenar belirleme operatörleri (örneğin, Sobel ve Prewitt) daha etkili olabilir. Roberts operatörü, özellikle kaynaklar sınırlı olduğunda veya hızlı bir çözüm gerektiğinde tercih edilebilir. 3 kenar belirleme algoritması görüntü işleme çalışmalarında sık tercih edilen renkli Lena görüntüsüne uygulanmıştır. Şekil 6'da yukarıda bahsi geçen kenar belirleme algoritmalarının uygulandığı Lena görüntüsünün çıktılarına yer verilmiştir. Şekil 6 da Sobel ve Prewitt operatörlerinin kenarları daha belirgin işaretlediği görülmektedir.

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Şekil 5. Roberts operatörü matrisleri



(a)



(b)



(c)



(d)

Şekil 6. Lena görüntüsü a)orjinal b)Sobel c)Prewitt d)Roberts

Sonuç olarak, hangi operatörün kullanılacağı görevin gereksinimlerine bağlı olarak değişmektedir. Sobel ve Prewitt, genellikle genel amaçlı kenar tespiti için daha uygundur, Roberts ise daha basit uygulamalarda tercih dlebilir. Sobel ve Prewitt bir pikselin çevresindeki piksellerin yoğunluk farklarını kullanarak kenarları tespit etmektedirler. Bu farklar, gradyan (türev) hesaplamak için kullanılmaktadır. Roberts operatörü de yoğunluk

farklarından yararlanılır, ancak daha küçük bir çekirdek kullanır. Yöntemlerin kenar belirleme hassasiyetleri açısından incelendiğinde, Sobel ve Prewitt genellikle daha büyük matrisler kullanarak daha geniş bir bölgeyi analiz ederler. Bu nedenle, daha geniş kenarları belirleme konusunda daha etkilidirler. Roberts operatörü, daha küçük bir çekirdek kullanmaktadır, bu nedenle daha küçük kenarlara duyarlıdır.

Deneyisel Bulgular

Deneyisel bulgular elde edilirken BSDS500 veritabanından seçilen 100 adet görüntü kullanılmıştır. Görüntülerin enterpolasyonu için 2x2 üst örneklenmesi amaçlanmıştır. Bu sebeple görüntülerin öncelikle $\frac{1}{2} \times \frac{1}{2}$ alt örnekleme elde edilmiştir. Alt örnekleme için doğrudan çıkarma yöntemi kullanılmıştır. Sonrasında tekrar 2 x 2 boyutuna entepole edilmiştir. Son olarak elde edilen görüntüler orijinal görüntülerle karşılaştırılmıştır. Karşılaştırmalar SSIM ve PSNR metrikleri ile gerçekleştirilmiştir. SSIM (Structural Similarity Index Measure) ve PSNR (Peak Signal-to-Noise Ratio) hesaplamaları;

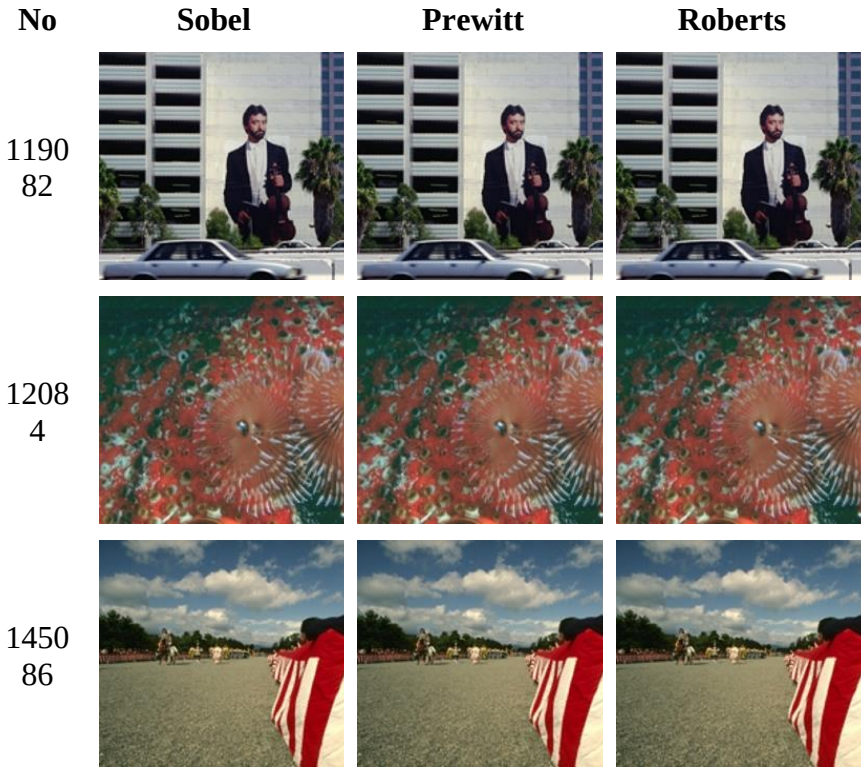
$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right),$$

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i, j) - O(i, j)]^2, \quad (6)$$

$$SSIM = \frac{(2\mu_I \mu_O + c_1)(2\sigma_{IO} + c_2)}{(\mu_I^2 \mu_O^2 + c_1)(\sigma_I^2 + \sigma_O^2 + c_2)}$$

şeklinde dir. Burada I ve O sırasıyla orijinal ve enterpolasyon görüntülerini temsil etmektedir. μ değerleri görüntülerin ortalamalarını, σ_I ve σ_O değeri görüntülerin varyanslarını, σ_{IO} ise kovaryanslarını temsil etmektedir. SSIM benzerlik metriği son yıllarda görüntü benzerliğinde kullanılan önemli yaklaşımlardandır. SSIM değeri 1'e yaklaştıkça görüntünün benzerliğinin arttığı anlamına gelmektedir. PSNR genellikle desibel ölçeği kullanan logaritmik bir ölçüm aracı olarak ifade edilir. PSNR için değerler,

bit derinliğinin 8 bit olması koşuluyla 30 ila 50 dB arasındadır ve daha yüksek olması başarının daha yüksek olduğu anlamına gelmektedir. PSNR değeri 60 dB veya daha yüksek olduğunda 12 bit görüntülerin işleme kalitesi yüksek olarak kabul edilir (Chervyakov, et al., 2020). Şekil 7, BSDS500 veri tabanında bulunan rastgele seçilmiş altı farklı görüntünün enterpolasyon çıktıları bulunmaktadır. Çıktı görüntüler incelendiğinde tüm yöntemlerin birbirine yakın sonuçlar ürettiği görülmektedir. Tablo 1’de ise aşağıdaki verilen altı farklı görüntünün SSIM ve PSNR değerleri bulunmaktadır.



2107
7



2530
27



3096

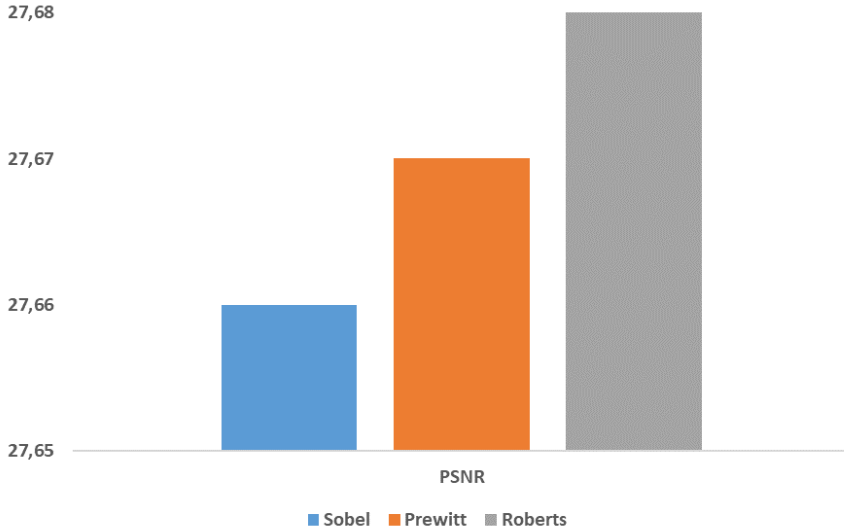


Şekil 7. Farklı kenar belirleme tekniklerinin kullanıldığı enterpolasyon yaklaşımlarının çıktı görüntüleri

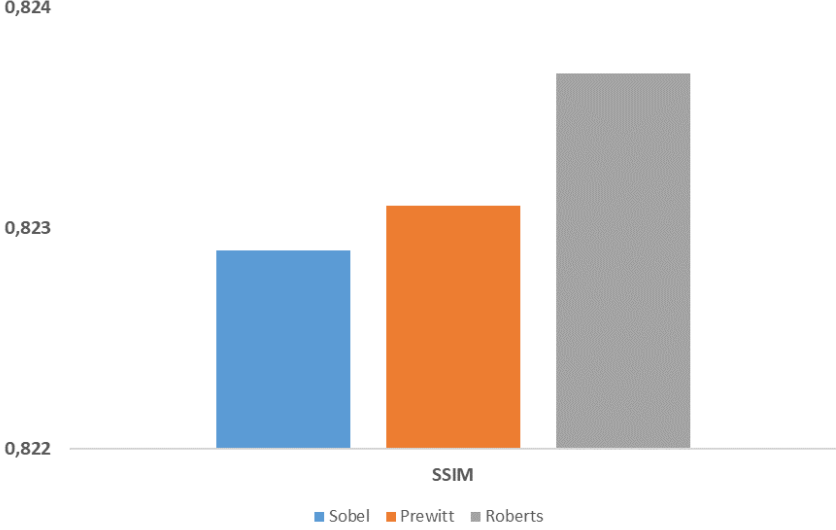
Tablo 1. Yöntemlerin performansları

	Sobel		Prewitt		Roberts	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
119082	24,51302	0,810605	24,51262	0,810628	24,54169	0,811396
12084	28,75097	0,856457	28,76624	0,856647	28,81504	0,85787
145086	25,36081	0,789924	25,36782	0,790054	25,34051	0,790221
21077	26,741	0,807664	26,73955	0,807672	26,74121	0,808147
253027	23,55881	0,814551	23,5697	0,814775	23,42066	0,813101
3096	37,4503	0,978338	37,47315	0,978366	37,54476	0,97848
Ortalama	27,72915	0,842923	27,73818	0,843024	27,73398	0,843203

Tablo 1 incelendiğinde kenar tespit yaklaşımlarının enterpolasyon sonuçları birbirine çok yakın değerler ürettiği görülmüştür. Söz konusu durumun temel sebebi yaklaşımların gradyan temelli teknikler olmasından kaynaklanmaktadır. Tablo 1, Şekil 7 de verilen görsel çıktıları da desteklemektedir. Şekil 8 ve Şekil 9 da 100 görüntünün ortalama PSNR ve SSIM değerlerine yer verilmiştir. Tablo 1, Şekil 8 ve Şekil 9'a bakıldığında Roberts operatörü ile daha başarılı enterpolasyon yapıldığı görülmektedir. Ancak performanslar arasında çok düşük bir fark olması, farklı görüntülerde ya da daha büyük veri tabanlarında başarılarının değişebileceğini göstermektedir.



Şekil 8. Yöntemlerin genel performansı: PSNR



Şekil 9. Yöntemlerin genel performansı: SSIM

Sonuç

Enterpolasyon görüntü işlemede kullanılan önemli bir tekniktir. Son yıllarda enterpolasyon performansını iyileştirmek adına görüntünün kenarlarının kullanıldığı birçok teknik geliştirilmiştir. Dolayısıyla kenar belirleme tekniklerinin enterpolasyon için önemli olduğu görülmektedir. Bu çalışmada gradyan temelli kenar belirleme tekniklerinden olan Sobel, Prewitt ve Roberts operatörleri kullanılarak görüntü enterpolasyonu gerçekleştirilmiştir. Çalışmada öncelikle Sobel, Prewitt ve Roberts kenar operatörleri ile kenar pikseller tespit edilmiştir. Kenar belirleme işleminde yatay, dikey ve diyagonal eksenlerdeki tüm kenarlar işleme dahil edilmiştir. Ardından kenar olarak tanımlanan piksellere 1 boyutlu kübik enterpolasyon, geri kalan piksellere ise bikübik enterpolasyon uygulanmıştır. Deneysel çalışmalar BSDS500 validation veri setindeki 100 adet görüntü üzerinde gerçekleştirilmiştir. Son olarak değerlendirme kriteri olarak PSNR ve SSIM değerlerinden faydalanılarak, Sobel, Prewitt ve Roberts

kenar belirleme operatörlerinin görüntü enterpolasyonunda ki başarıları karşılaştırılmıştır. Görüntüler öncelikle alt örneklenmiş ve sonrasında orijinal boyutuna getirilmiştir. Kullanılan kenar belirleme tekniklerinin benzer nitelikte olması enterpolasyon sonuçlarına yansımıştır. Sobel, Prewitt ve Roberts kenar belirleme teknikleri birbirine çok yakın sonuçlar üretmiştir. Gelecekte yapay zeka ve optimizasyon tabanlı kenar belirleme algoritmalarının enterpolasyon süreci üzerinde etkilerinin incelenmesi amaçlanmaktadır.

KAYNAKÇA

Chaple, G. N., Daruwala, R., & Gofane, M. S. (2015). Comparisons of Robert, Prewitt, Sobel operator based edge detection methods for real time uses on FPGA. 2015 International Conference on Technologies for Sustainable Development (ICTSD),

Chervyakov, Nikolay; Lyakhov, Pavel; Nagornov, Nikolay (2020). "Analysis of the Quantization Noise in Discrete Wavelet Transform Filters for 3D Medical Imaging". *Applied Sciences*. 10 (4): 1223.

Dengwen, Z. (2010, October). An edge-directed bicubic interpolation algorithm. In 2010 3rd international congress on image and signal processing (Vol. 3, pp. 1186-1189). IEEE.

İncetaş, M. O. (2022). Anisotropic Diffusion Filter Based on Spiking Neural Network Model. *Arabian Journal for Science and Engineering*, 47(8), 9849-9860.

İncetaş, M. O. (2023). Image Interpolation Based on Spiking Neural Network Model. *Applied Sciences*, 13(4), 2438.

İncetaş, M. O., & Arslan, R. U. (2019). Edge Detection Using Integrate and Fire Neuron. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 23(2), 611-616.

İNCETAŞ, M. O., KILIÇASLAN, M., & FARŞİ, T. R. (2022). Image retrieval with SNN-based multi-level thresholding. *Gümüşhane Üniversitesi Fen Bilimleri Dergisi*, 98-108.

İNCETAŞ, M. O., & Murat, U. İĞNECİKLİ SİNİR AĞINA DAYALI BENZERLİK DÖNÜŞÜMÜ.

Jia, Z., & Huang, Q. (2022). Image Interpolation with Regional Gradient Estimation. *Applied Sciences*, 12(15), 7359.

Jin, Y., Zhang, Y., Cen, Y., Li, Y., Mladenovic, V., & Voronin, V. (2021). Pedestrian detection with super-resolution reconstruction for low-quality image. *Pattern Recognition*, 115, 107846.

Kanopoulos, N., Vasanthavada, N., & Baker, R. L. (1988). Design of an image edge detection filter using the Sobel operator. *IEEE Journal of solid-state circuits*, 23(2), 358-367.

Karwowska, K., & Wierzbicki, D. (2022). Using super-resolution algorithms for small satellite imagery: A systematic review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15, 3292-3312.

Li, J., Wu, J., Deng, H., & Liu, J. (2016). A self-learning image super-resolution method via sparse representation and non-local similarity. *Neurocomputing*, 184, 196-206.

Liu, Y., Huang, Q., & Sato, K. (2020). Differential filtering algorithm for robot welding seam image enhancement. *Journal of Physics: Conference Series*,

Ren, C., He, X., Pu, Y., & Nguyen, T. Q. (2019). Enhanced non-local total variation model and multi-directional feature prediction prior for single image super resolution. *IEEE Transactions on Image Processing*, 28(8), 3778-3793.

Sánchez-García, E., Balaguer-Beser, Á., Almonacid-Caballer, J., & Pardo-Pascual, J. E. (2019). A new adaptive image interpolation method to define the shoreline at sub-pixel level. *Remote sensing*, 11(16), 1880.

Shi, J., & Reichenbach, S. E. (2006). Image interpolation by two-dimensional parametric cubic convolution. *IEEE Transactions on Image Processing*, 15(7), 1857-1870.

Wang, H., Gao, X., Zhang, K., & Li, J. (2017). Single Image Super-Resolution Using Gaussian Process Regression With Dictionary-Based Sampling and Student- t Likelihood. *IEEE Transactions on Image Processing*, 26(7), 3556-3568.

Wei, Z., & Ma, K.-K. (2013). Contrast-guided image interpolation. *IEEE Transactions on Image Processing*, 22(11), 4271-4285.

Zhong, B., Ma, K.-K., & Lu, Z. (2019). Predictor-corrector image interpolation. *Journal of Visual Communication and Image Representation*, 61, 50-60.

BÖLÜM IV

B-GAN: Biyomedikal Bilişim Uygulamalarında Yapay Zekanın Üretken Yüzü

İsa ATAŞ¹

Giriş

Üretken Çekişmeli Ağlar veya GAN'lar, yapay zekâ alanında üretken bir yüz, özellikle biyomedikal uygulamalarında dönüştürücü bir güç olarak öne çıkmaktadır. Popüler bir derin öğrenme modeli olan GAN'lar, (Goodfellow & ark. 2014) tarafından tanıtılmıştır. GAN çekişmeli bir ortamda derin sinir ağında eğitilir ve veri dağılım modelini öğrenir. Makine öğreniminde son 10 yılda geliştirilen en ilginç ve başarılı üretken model olarak yapay zekâ alanında en güncel araştırma yöntemlerinden biri haline gelmiştir. Önerildiği günden bu yana büyük ilgi görmektedir. GAN'ın mükemmel performansla sahip üretken bir model olarak kullanılması ile birlikte ilham verici çekişmeli öğrenme fikrinin bir dizi yeni araştırmaya yön

¹ Dr. Öğr. Üyesi, Dicle Üniversitesi - Diyarbakır Teknik Bilimler Meslek Yüksekokulu - Bilgisayar Teknolojileri Bölümü

vermesi ve çeşitli uygulamalarla sonuçlanması özellikle önemlidir (Lin & ark. 2018). GAN'lar temel olarak bir üretici ve bir ayırt edici ağ bileşeninden oluşur. Bu iki bileşen rekabet halinde çekişmeli olarak eğitilir. Çalışmaların gereksinimlerine bağlı olarak, güçlü bir üretici ya da hassas bir ayırt edici tasarımı hedeflenir. Bu bölümde GAN dünyasını, çalışma prensibini, gelişim tarihini ve biyomedikal alanda yaptıkları (Zhang & ark. 2019) (Hiasa & ark. 2018) (Zhu & ark. 2017) dikkate değer katkıları incelenerek GAN'ın tıbbi görüntülerin iyileştirilmesindeki kullanımı açıklanmış ayrıca biyomedikal görüntüler üzerinde üretken çekişmeli ağlar ile yapılacak farklı çalışmalara yol gösterecek bir kaynak olması hedeflenmiştir.

GAN'ları Anlamak

Derin öğrenme modelleri genel olarak diskriminant modeller ve üretken modeller olarak ikiye ayrılır. Diskriminant model, olasılık ve istatistik yönünde bilinmeyen veri ile bilinen veri arasındaki ilişkiyi modellemeye çalışır. Üretken model ise, özellikle bazı gözlemlenemeyen parametrelerin verilmesi koşuluyla rastgele gözlemler üretebilen bir modeli ifade eder (Feng & ark. 2018). Üretken modellerin gelişimi ilk zamanlarda modellemenin zorluğu nedeniyle gecikmiştir, fakat üretken model, makine öğrenimi tarihinde çok önemli bir role sahiptir. Derin öğrenme yöntemleri büyük miktarda veriyi (görüntü, konuşma, metin vb. gibi) işlerken, üretken modeller bu yüksek boyutlu verilerin dağılımını simüle etmemize yardımcı olabilir. Üretken modeller, veri üretme (Ataş, 2023a), veri büyütme (Xian & ark. 2017), altyazı oluşturma (Ataş, 2023b) gibi birçok uygulama için faydalı olmaktadır.

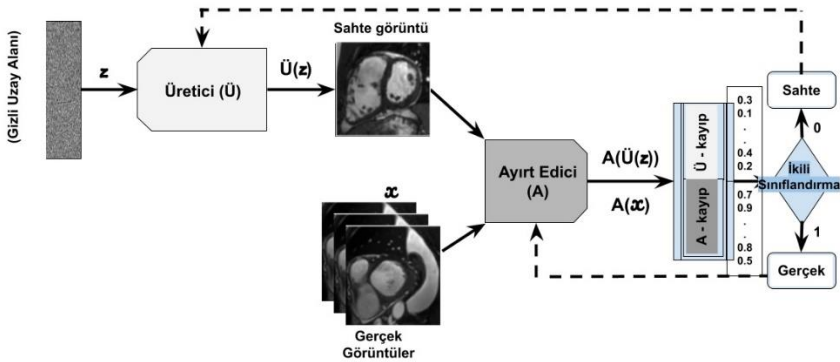
GAN Mimarisi ve İlkeleri

GAN, iki yapay sinir ağı arasında rekabet temelli bir öğrenme modelini ifade eder. Bu iki ağ, biri "Üretici" ve diğeri "Ayırt edici" olarak adlandırılır. Üretici ağ (Ü), rastgele gürültü veya başka bir giriş verisi kullanarak, gerçek benzeri veriler üretmeye çalışır. Ayırt edici ağ (A) ise gerçek verileri sahte verilerden ayırt etmeye çalışır.

Çekişmeli geçen bu eğitim döngüsünde, üreticinin giderek daha gerçekçi veriler üretmesi ve ayırt edicinin bu verileri ayırt etme yeteneğinin artmasıyla devam eder. (Goodfellow & ark. 2014). Üretici, Z 'yi (gizli bir alan verisi) X 'e (gerçek bir görüntü verisi) eşitlerken Ayırt edici, üretilen çıktının gerçek mi yoksa sentetik mi olduğunu tanımlar. Üretici ve Ayırt ediciyi tanımlayan amaç fonksiyonu $V(A, \hat{U})$, denklem (1)'de verilmiştir.

$$\min_{\hat{U}} \max_A V(A, \hat{U}) = E_{x \sim P_{data(x)}} [\log A(x)] + E_{z \sim P_{data(z)}} [\log (1 - A(\hat{U}(z)))] \quad (1)$$

Burada x , veri dağıtımından örneklenen gerçek verileri z , üreticinin dağılımından örneklenen rastgele bir gürültüyü temsil eder. $A(x)$ ve $\hat{U}(z)$ sırasıyla ayırt edici ve üretici ağ çıkışlarıdır. GAN mimarisi, Şekil 1'de gösterilmiştir. Bu model, üretici ve ayırt ediciyi aynı anda eğitir. Üretici, gizli uzay z alanında gerçeğe yakın veriler üretir. Ayırt edici, üretilen verilerin referans aldığı gerçek veriler ile karşılaştırır. Değerlendirme sonucunda benzerlik yönünden 0 ile 1 arasında kayıp değerler bulunur; 1'e yakın değerler gerçek verileri, 0'a yakın değerler ise sentetik verileri temsil eder. $\hat{U}$ - kayıp ve A - kayıp, sırasıyla üretici ve ayırt edicinin kaybıdır. (Cheng & ark. 2020).

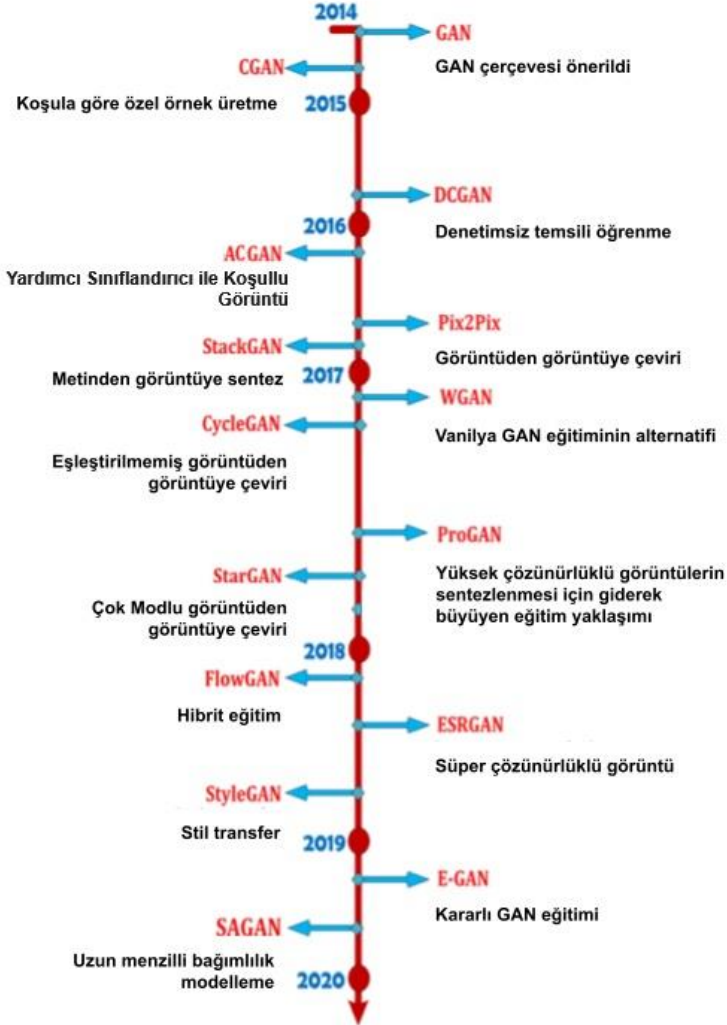


Şekil 1. Üretken Çekişmeli Ağ Mimarisi

GAN'ın Gelişim Evreleri

GAN'lar diğer üretken modellere göre daha gerçekçi görüntüler oluşturabilir. Bu avantajların yanı sıra GAN modeli oldukça özgür ve kontrol edilemez olduğundan yanlılık sorunu, kaybolan gradyan, eğitim zorluğu ve zayıf çeşitlilik gibi birçok sorun bulunmaktadır (Wang & ark. 2017). Daha iyi GAN'lar elde etmek için farklı optimizasyon yöntemleriyle birçok çaba harcanmıştır. Bu nedenle, 2014 yılından bu yana GAN ile ilgili teoriler ve makaleler sonsuz bir akış içinde ortaya çıkmış ve üretilen sonuçların performansını, kararlılığını ve kalitesini artırmak için vanilya GAN metodolojisinin çok sayıda çeşidi önerilmiştir (Gonog & ark. 2019).

Araştırmacılar çeşitli uygulamalar için CGAN (koşula göre özel örnek üreten), DCGAN (denetimsiz), ACGAN (Yardımcı sınıflandırıcılı), Pix2Pix (görüntüden görüntüye), StackGAN (metinden görüntüye), WGAN (GAN alternatifi), CycleGAN (Eşleştirilmemiş), ProGAN (yüksek çözünürlüklü), StarGAN (çok modlu), FlowGAN (hibrit), ESRGAN (süper çözünürlüklü), StyleGAN (stil dönüşümü), EGAN (kararlı), SAGAN (bağımlı) vb. gibi birçok GAN varyantı türetmişlerdir. GAN'ın gelişim zaman çizelgesi Şekil 2'de gösterilmiştir. Bu bölümde biyomedikal uygulamalarda yaygın olarak kullanılan GAN modellerinden birkaçı tanıtılacaktır.



Şekil 2. GAN model gelişiminin zaman çizelgesi (Pavan & Jayagopal, 2020)

Conditional GAN (CGAN): (Mirza & Osindero, 2014) tarafından önerilen geliştirilmiş bir GAN modelidir. Öğrenme sürecinde belirli koşulları (etiketleri veya sınıfları) kullanarak özel

veri üretme yeteneği sağlar. CGAN'larda, GAN'ların temel yapısını genişleterek etiket bilgisini algoritmanın içine entegre etmişlerdir. Görüntü Sentezi, Görüntü Düzenleme, Veri Artırma ve Çoklu Etiketleme gibi kullanım alanları mevcuttur. CGAN'lar, biyomedikal alanda, belirli hastalıklar özellikle patolojik durumlar için etkili olabilir. Örneğin, belirli bir patolojiye ait biyomedikal görüntülerin sentezinde veya spesifik bir hastalık durumunu simüle etmek amacıyla kullanılabilirler. Bu modeller, denetimli öğrenme görevlerinde daha fazla esneklik ve kontrol sağladıkları için popülerdir.

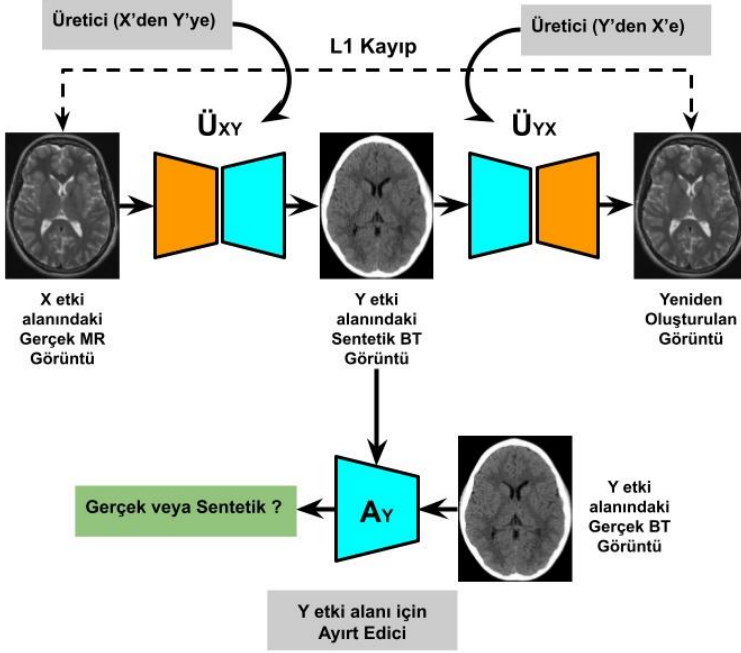
Pix2Pix GAN: (Isola & ark. 2017) tarafından önerilen geliştirilmiş bir GAN modelidir. Özellikle çeşitli görüntüleme uygulamalarında kullanılmak üzere tasarlanmıştır. Pix2Pix, bir görüntüleme türünden diğerine geçişi öğrenmek amacıyla CGAN mimarisini kullanır. Yani, belirli bir giriş görüntüsünden başka bir hedef görüntüsüne dönüşüm yapma yeteneğine odaklanır. Eğitim sürecinde algoritma, giriş görüntüsü ile hedef görüntü arasında doğrudan bir eşleşme yapmayı öğrenir. Veri artırma ve görüntü iyileştirme görevlerinde, düşük çözünürlüklü bir görüntüyü yüksek çözünürlüklü bir görüntüye dönüştürmede kullanılır. Giriş görüntüsünün belirli bölgelerini tanımlama ve ayrıştırma yeteneği, özellikle görüntü segmentasyonu uygulamalarında kullanışlıdır. Pix2Pix, biyomedikal görüntüleme gibi alanlarda başarıyla kullanılmıştır. Normal bir görüntüyü manyetik rezonans görüntüsüne dönüştürmek veya radyolojik görüntü iyileştirmesi için kullanmak mümkündür.

Wasserstein GAN (WGAN): (Arjovsky, Chintala & Bottou, 2017) tarafından önerilen geliştirilmiş bir GAN modelidir. WGAN'lar, GAN'ların eğitim sürecini kontrol ederek daha iyi sonuçlar elde etme amaçlı ortaya çıkmıştır. WGAN'lar, modelin ürettiği dağılım ile gerçek veri dağılımı arasındaki farkı ölçen Wasserstein mesafesini kullanır. Wasserstein mesafesi, eğitim sırasında daha açık bir optimizasyon düzenlenmesine giderek, GAN'ların daha hızlı ve daha kontrollü bir şekilde eğitilmesine yardımcı olabilir. Geleneksel GAN mimarisine benzer şekilde bir

retici ve bir ayırt edici aę ierir. Bu tr GAN'lar, biyomedikal veri setlerinde daha iyi eęitim performansı saęlayabilir.

StarGAN: (Choi & ark. 2018) tarafından nerilen geliřtirilmiř bir GAN modelidir. Birden ok etiketi veya stil zellięini ynetme yeteneęi ile ne ıkar. Bu model, tek bir aę kullanarak farklı etiketlere sahip verileri birleřtirmeyi ve dnřtrmeyi amalar. StarGAN, birok farklı uygulamada kullanılabilecek oklu etiket transferi yeteneklerine odaklanmıřtır. oklu etiketleri ynetme yeteneęine sahip olan bu GAN tipi, farklı kaynaklardan gelen verileri birleřtirmek ve eřitli biyomedikal veri setlerini entegre etmek iin kullanılabilir. Uydu grntlerini haritalamada, siyah beyaz fotoęrafları renklendirmede, bir sanat eserinin stilini deęiřtirmede veya farklı hastanelerden gelen grnt verilerini birleřtirerek genel bir model oluřurmada kullanılabilir. Geleneksel GAN'lar belirli bir grnt eviri grevi iin eęitilir, ancak StarGAN ayrı modellere ihtiya duymadan eřitli grevleri gerekleřtirmeyi ęrenebilir.

CycleGAN: İki farklı veri kmesi arasında dnřm yapmak iin (Zhu & ark. 2017) tarafından nerilen geliřtirilmiř bir GAN modelidir. Bu model, GAN mimarisini kullanarak bir dizi grntde her iki ynde dnřm gerekleřtirebilir ve iki farklı alan arasında eviri yapabilir. rneęin, bir CycleGAN modeli, manyetik rezonans (MR) grntlerini bilgisayarlı tomografi (BT) grntlerine evirebilir ve aynı zamanda BT grntlerini MR grntlerine evirmede de kullanılabilir. Bu dnřm iřlemi "ift ynl baęlantı" prensibi ile gerekleřtirilir. İki farklı kaynaktan gelen grntlerin ortak grsel ierięe sahip olması nemlidir. Normal grnt evirisi, iki alan arasında nemli bir fark olduęunda byk lde etkilenebilir. Bununla birlikte, bazı az grlen hastalıkların teřhisinde ok modaliteli tıbbi grnt sentezi iin kullanılabilir. Biyomedikal uygulamalarının tıbbi grnt analizlerinde referans olarak genelde CycleGAN modeli tercih edilmektedir (Wolterink & ark. 2017), (Huo & ark. 2017), (Tanner & ark. 2018), (Hongfei & ark. 2022). řekil 3'de Cycle-GAN modeli gsterilmiřtir.



Şekil 3. Cycle-GAN modeli (Zhu & ark. 2017)

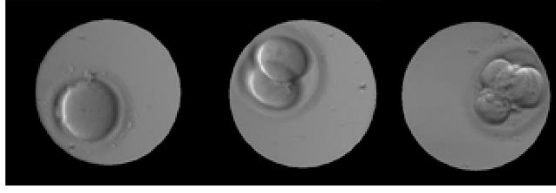
Biyomedikal Uygulamalarda GAN'lar

Biyomedikal uygulamalarda GAN'lar oldukça heyecan verici potansiyele sahiptir. Bu yapay zekâ modeli, veri üretme ve dönüştürme konusunda etkili bir şekilde kullanılabilir. Biyomedikal görüntüleme alanında GAN'lar, Tıbbi Görüntü İşleme, Tıp Bilişim ve Biyoinformatik dallarında etken bir rol üstlenebilir ayrıca daha fazla eğitim verisi oluşturmak veya var olan veriyi iyileştirmek için kullanılabilir (Lan & ark. 2020). Ancak, GAN'ların kullanımı etik ve güvenlik konularını da beraberinde getirir. Hassas tıbbi verilerin güvenliği ve doğruluğu büyük önem taşır. Bu nedenle, bu teknolojinin biyomedikal alanda kullanımı, dikkatli bir şekilde düzenlenmelidir (Zhang & ark. 2023).

GAN'ın Tıbbi Görüntü Analizinde Uygulanması

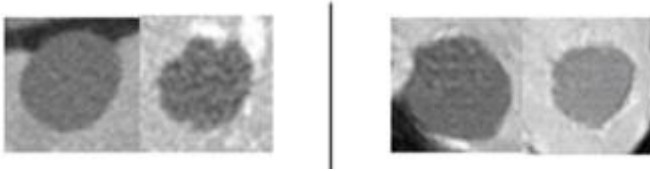
Yapay zekanın gelişmesi ile birlikte günümüzde bilim insanları tarafından tıbbi görüntülere talep çok artmıştır. Tıbbi görüntü analizinde işçilik, zaman ve maliyet gibi faktörler, araştırmacıların etiketli tıbbi görüntüler elde etmesini normal görüntülere göre daha zor hale getirmiştir. Son zamanlarda, derin öğrenme algoritmalarının gelişmesi ve etiketli görüntü veri kümelerinin büyümesiyle birlikte, evrişimli sinir ağları (ESA) tabanlı modeller birçok bilgisayarla görme görevinde büyük başarı elde etmiştir. ImageNet (Deng & ark. 2009), COCO (Lin & ark. 2014), gibi büyük görüntü veri kümeleri üzerinde önceden eğitilmiş birçok derin öğrenme modeli mevcuttur. Transfer öğrenimi (Shin & ark. 2016), denetimli eğitim için sınırlı etiketli veri kullanır. Transfer öğrenme görevlerinde, modelin ağırlıklarının çoğu sabit kalır ve yeni veri kümesinde yalnızca son birkaç katmanın ağırlıkları ince ayar yapılır. Bu şekilde, tıbbi görüntü analizinde önceden iyi eğitilmiş bir derin sinir ağı uygulanır. Transfer öğrenimi, günümüzde eğitim görüntülerinin eksikliğinden muzdarip olabilir (Wang & ark. 2019). Geleneksel veri büyütme yöntemleri yalnızca orijinalleriyle yakın bir dağılımı paylaşan veriler üretebilir. Veri seti çok küçükse, bu yöntemlerin veri artırma konusunda neredeyse hiç çabası yoktur. GAN tıbbi görüntü analizinde veri eksikliğine bir çözüm sunarak birçok çalışmada uygulanmıştır.

(Dirvanauskas & ark. 2019) geleneksel bir GAN modeli ile üç aşama (tek hücreli, iki hücreli ve dört hücreli) için insan embriyo hücresi görüntüleri oluşturmuşlardır. Sentezlenen tüm görüntüler, embriyo görüntü işleme görevleri için geliştirme, eğitim ve değerlendirmeyi kolaylaştırmak için kullanılabilir. Sentetik oluşturulan embriyonik hücre görüntüleri Şekil 4'de gösterilmiştir.



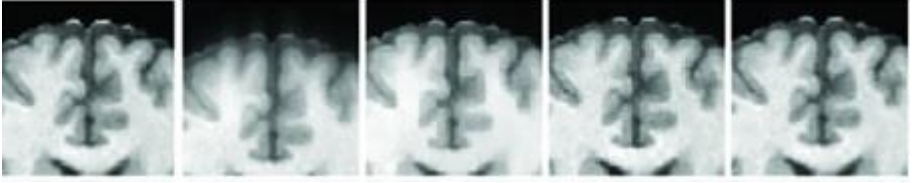
Şekil 4. GAN modeli ile oluşturulan embriyo hücre görüntüleri (Dirvanauskas & ark. 2019)

(Frid-Adar & ark. 2018) sentetik karaciğer lezyonları oluşturmak için GAN modellerinden DCGAN ve ACGAN kullanmışlardır. Sentetik görüntüler, 64×64 çözünürlüğe sahip karın BT görüntülerinde ilgilenilen bölgeleri (ROI) içeriyordu. Deney sonuçları, bu iki GAN modelinden elde edilen sentetik veri büyütmenin, klasik veri büyütmeyle kıyasla sınıflandırma doğruluğunu %77,5'ten %85,0'a çıkardığını göstermiştir. Üretilen sentetik lezyonlar Şekil 5'de gösterilmiştir



Şekil 5. Sol taraf: Gerçek lezyonlar; Sağ taraf: Sentetik lezyonlar (Frid-Adar & ark. 2018)

(Shitrit & Raviv, 2017) MR görüntü oluşturma sürecini hızlandırmak için GAN uyguladılar. Yazarlar, mevcut MR görüntülerinden MR görüntüleri oluşturmak yerine, eksik k-alanı örneklerini oluşturmak için GAN'dan yararlandılar. Yaklaşımları, zamana duyarlı veya çözünürlüğe duyarlı MR tarama görevleri için kullanılabilir. Sentetik üretilen beyin MR görüntüler Şekil 6'da gösterilmiştir.



Şekil 6. Sentetik Beyin MR Görüntüleri (Shitrit & Raviv, 2017)

(Mahapatra & Bozorgtabar, 2017) yüksek çözünürlüklü retinal görüntüler oluşturmak için yerel belirginlik haritalarını ve GAN'ı kullandılar. GAN'ın kayıp işlevine ek olarak, belirginlik haritalarındaki yüksek retinal görüntüleri ile düşük çözünürlüklü görüntüler arasındaki farktan kaynaklanan yerel belirginlik kaybını da eklediler. Üretilen yüksek çözünürlüklü sentetik retinal görüntüler Şekil 7'de gösterilmiştir.

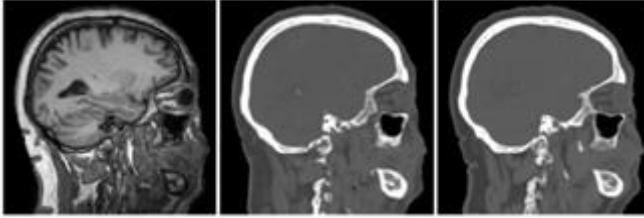


Şekil 7. Yüksek Çözünürlüklü Gerçek ve Sentetik Retinal Görüntüler

(Mahapatra & Bozorgtabar, 2017)

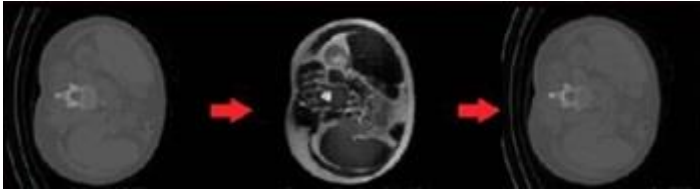
(Wolterink & ark. 2017) radyoterapi tedavi planlamasında beyin tümörlerinin MR ve BT görüntülerine Cycle-GAN uyguladılar. Sınırlı sayıda hastada MR ve BT görüntüleri eşlendiğinden, eşleştirilmemiş görüntülerden önce görüntüleri daha büyük bir çözünürlüğe doldurarak ve ardından benzer daha küçük çözünürlüğe kırparak eşleştirilmemiş görüntüler ürettiler. Yazarlar eşleştirilmemiş görüntülerle eğitilen Cycle-GAN'ın, eşleştirilmiş görüntülerle eğitilmiş tek bir GAN'dan daha iyi performans gösterdiğini gösterdiler. MR (sağ) ve BT (sol) görüntü dilimleri

kullanılarak Sentetik BT (orta) görüntü dönüşümü Şekil 8’de gösterilmiştir.



Şekil 8. MR ve BT etki alan etkileşimi (Wolterink & ark. 2017)

(Huo & ark. 2017), sentez ve segmentasyon arasındaki tamamlayıcı bilgilerden yararlanmak için Cycle-GAN ve segmentasyon ağını uçtan uca birleştirdiler. Ağlarının son kayıp fonksiyonu, Cycle-GAN kayıp fonksiyonlarından ve segmentasyon kayıp fonksiyonundan oluşuyordu. Deneylerinde, çoklu organ etiketlerine sahip MR görüntülerinin, karşılık gelen segmentli BT görüntüleri oluşturmak için kullanılabileceğini gösterdiler. Şekil 9’da BT (sol) görüntüsünden sentezlenen MR (orta) görüntüsünün, MR görüntüsünden yeniden yapılandırılan BT (sağ) görüntüsünün eğitim yolu ile dönüşümleri gösterilmiştir.

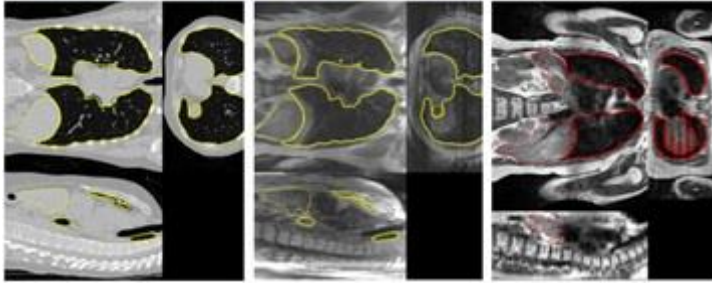


Şekil 9. MR ve BT etki alan etkileşimi (Huo & ark. 2017)

(Tanner & ark. 2018), Akciğer MR ve BT görüntülerinin deforme edilebilir görüntü kaydıyla (modalitelerin yapısal farklılıkları ve eksik yoğun zemin gerçeği) eşlenmemiş beyin verileri arasındaki yoğunluk ilişkisini öğrenmek için Cycle-GAN modelini kullanmışlardır.

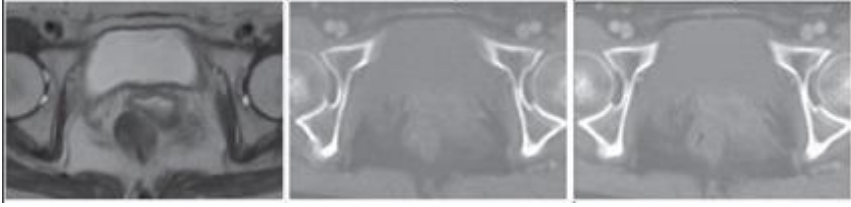
Bu çalışmada iki popülasyonu birbirinden ayıran geometrik

değişikliklerin (örneğin, nefes alma ve nefes verme) de kolayca sentezlendiğini göstermişlerdir. Bu, gerçek ve sentezlenen görüntü arasındaki uzamsal bağlantıların korunması için önemlidir (örneğin, plan, segmentasyon, yer işareti yayılımı). Bu sorunu hafifletmek için, alıcı alanlarının boyutunu azaltarak ayırt ediciye sağlanan uzamsal bilgiyi azaltmaya çalışmışlardır. Şekil 10’da soldan sağa doğru gerçek BT, sentezlenmiş MR ve gerçek MR görüntüleri gösterilmiştir.



Şekil 10. BT ve MR etki alan etkileşimi (Tanner & ark. 2018)

(Hongfei & ark. 2022), MR görüntülerinden daha yüksek kaliteli sentetik BT’yi sentezlemek için çok ayırıcı tabanlı CycleGAN modeli önermişlerdir. Modeli eğitmek için rahim ağzı kanseri simülasyonu aşamasında elde edilen MR ve BT görüntüleri seçildi. Üretici ağı olarak, DenseNet’i ana mimari olarak benimsediler. Test aşamasında modeli dörtlü çapraz doğrulama yöntemiyle doğruladılar. Tahmin aşamasında, BT görüntülerinin anatomi doğruluğunu değerlendirmek için Cycle-GAN tarafından sentezlenen BT görüntüleri ile karşılaştırdılar. Cycle-GAN modeli ile sentezlenen BT görsel sonuçlarından Şekil 11’de soldan sağa doğru gerçek MR, sentezlenmiş BT ve gerçek BT görüntüleri gösterilmiştir.



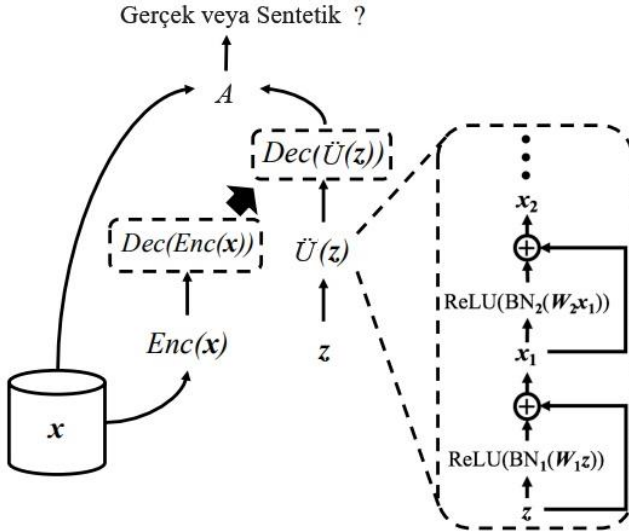
Şekil 11. MR ve BT etki alan etkileşimi (Hongfei & ark. 2022)

GAN'ın Tıp Bilişiminde Uygulanması

Sağlık bilişiminin gelişmesiyle birlikte, elektronik tıbbi kayıtlar (ETK), hastane bilgi sistemleri, Nesnelerin İnterneti (diğer adıyla IoT) tabanlı sağlık platformu, giyilebilir cihazlar ve diğer platformlar tıbbi verilerin artışına olanak sağlamıştır (Li & ark. 2019). Tıbbi verilerin nicelik ve niteliğindeki artış, tıpta bilimsel araştırma ve algoritmaların kullanımını da kolaylaştırmıştır. Ayrıca, hasta verilerinde kimlik gizleme işleminden sonra bazı kombinasyonlarla yeniden tanımlanma olanağı sağlanmıştır. Tıbbi bilişim uygulaması, parametreleri eğitmek için genellikle çok sayıda veri gerektirir. Tıbbi veri eksikliği, derin öğrenme algoritmalarının, özellikle de yapay zekanın tıp bilişimi alanında uygulanmasını ciddi şekilde kısıtlamaktadır. Bu sebeple tıp bilişiminin gelişimi, tıbbi görüntüler gibi alanların gerisinde kalmaktadır. GAN, görüntü oluşturmada önemli bir rol oynadığı gibi sürekli veri oluşturmada da iyi bir performans göstermiştir. Geleneksel ve tıbbi analize yönelik geliştirilen GAN modelleri ayrık veri üretiminde doğrudan kullanılamaz. Tıbbi verilerde, bir hastalığın teşhisi ve hastalığın şiddeti ayrı verilerdir. Dengesiz sınıflandırma nedeniyle az etiketli ve maliyetli tıbbi veriler mevcuttur. Bu nedenle, GAN'ın gerçek tıbbi verilere yönelik ayrık veri üretme modelleri geliştirilmiştir. GAN tabanlı tıbbi verilere yönelik çalışmalardan bazıları aşağıda açıklanmıştır.

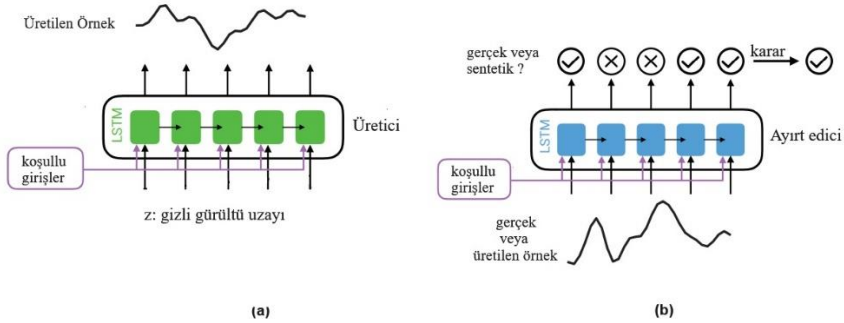
(Choi & ark. 2017), gerçeğe yakın sentetik hasta kayıtları oluşturmak için yeni bir yaklaşım olan Medikal Üretken Rekabet Ağı (medGAN) geliştirdiler. Mod çökmesini etkili bir şekilde önlemek ve toplu normalleştirme ve kısayol bağlantılarıyla öğrenme

verimliliğini artırmak için mini toplu ortalama almayı kullanmışlardır. medGAN, geleneksel GAN'daki sınırlamanın üstesinden gelmek için otomatik kodlayıcıdan yararlanarak yüksek boyutlu çok etiketli ayrık değişkenleri (teşhisler, ilaçlar ve prosedür kodları gibi ikili ve sayım değişkenleri) işleyebilir. Fizibilitiyi göstermek için medGAN'ın dağıtım istatistikleri, tahmine dayalı modelleme görevleri ve tıbbi uzman incelemesi dahil birçok deneyde gerçek verilerle karşılaştırılabilir. Yazarlar bu yöndeki analizlerinde ayrık değişkenler için etkileyici sentetik hasta kayıtları üretmişlerdir. Şekil 12'de medGAN mimarisi gösterilmiştir. (x) , ayrık veri setlerinden oluşan kaynak verilerdir. (z) , üretici tarafından oluşturulan rastgele örneklerdir; $\tilde{U}(z)$, kısayol bağlantılarına sahip ileri beslemeli bir ağıdır; Bir otomatik kodlayıcı (yani kodlayıcı (Enc) ve kod çözücü (Dec)) x kaynak verilerden öğrenilir; Ayrık veri çıkışı oluşturmak için ($\tilde{U}$) Üreticiden sonra aynı kod çözücü kullanılır. Ayırt edici (A), gerçek girdi olan x ile ayrık sentetik çıktı olan $Dec(\tilde{U}(z))$ 'yi ayırt etmeye çalışır.



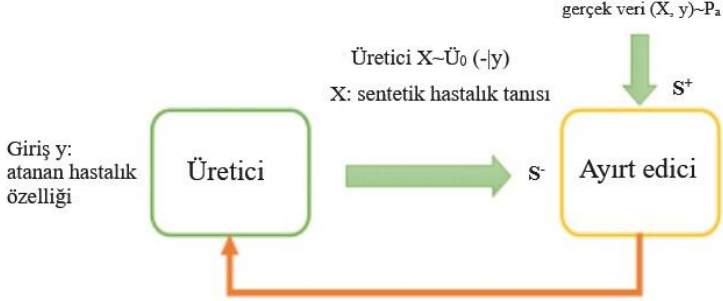
Şekil 12. medGAN mimarisi (Choi & ark. 2017)

(Esteban, Hyland & R  tsch, 2017), tıbbi veri uygulamalarına y  nelik ger  ek  i zaman serileri   retmek i  in Tekrarlayan GAN (RGAN) ve Tekrarlayan Ko  ullu GAN (RCGAN) modellerini   nerdiler. Uzun Kısa S  reli Bellek (di  er adıyla LSTM) hem ayırt edici hem de   retici i  in mimari olarak se  ilmi  tir. "Sentetik Veri   zerinde E  it, Ger  ek Veri   zerinde Test Et" adlı bir y  ntem kullanarak en sık kaydedilen d  rt de  i  kenli bir hastanın yakın gelecekte "kritik" hale gelip gelmeyece  ini tahmin ettiler. Ara  tırmacılar LSTM tabanlı RGAN ve RCGAN modellerinden, ger  ek verilerle kar  ıla  tırılabilir zamanlarda performans elde ettiler.   ekil 13'de LSTM tabanlı GAN mimarileri g  sterilmi  tir.



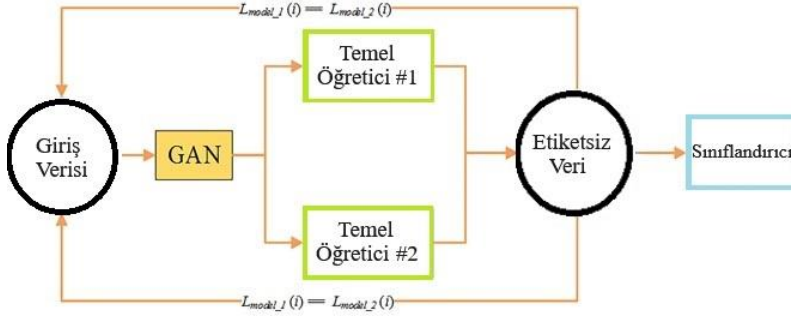
  ekil 13. (a)Tekrarlayan GAN (RGAN) ve (b)Tekrarlayan Ko  ullu GAN (RCGAN) mimarileri (Esteban, Hyland & R  tsch, 2017)

(Guan & ark. 2018), elektronik tıbbi kayıt (ETK) metinlerini peki  tirmek i  in medikal metin GAN di  er adıyla mtGAN kullanmı  lardır. mtGAN, belirlenmi   hastalık   zelliklerini girdi olarak alan ve gizlilik sorunlarının yanı sıra yetersiz ve dengesiz numune sorununu ele almak i  in kar  ılıklı gelen ETK metnini   reten ko  ullu bir GAN'dır. Ko  ullu kısıtlamayı empoze etmek i  in, hastalık   zellikleri her nesil adımımda ek bir girdi olarak modele beslenir. Bu   alı  ma ile hasta mahremiyetinin potansiyel sızıntısını   nlemek i  in yeni bir yol sa  lamı  lardır.   retici modelde RNN'nin geli  tirilmi   bir varyantı olan LSTM'yi kullanmı  lardır.   ekil 14'de mtGAN modeli g  sterilmi  tir.



Şekil 14. mtGAN mimarisi (Guan & ark. 2018)

(Yang & ark. 2018), IoT tabanlı sağlık hizmeti sisteminde klinik karar almayı desteklemek için GAN ile birlikte yarı denetimli bir yöntem sunmuşlardır. Yaklaşımlarında GAN, yalnızca etiketli veri sayısını artırmak için değil aynı zamanda yarı denetimli öğrenme performansını iyileştirmek amacıyla dengesiz etiketli sınıfları ek sentetik verilerle tamamlamayı benimsemişlerdir. GAN, etiketli seti girdi olarak alarak sentetik veriler üretmişlerdir. Etiketlenmemiş kümenin etiketleri iki öğretici tarafından tahmin edilmiş, genişletilmiş etiketli veri kümesi girdi alınarak, etiketli kümeyi oluşturmak için tekrardan GAN kullanılmıştır. Bir dizi kıyaslama ve gerçek dünyadaki tıbbi veri setleri üzerinde yapılan kapsamlı değerlendirmeler, önerilen tekniğin diğerlerinden daha iyi performans gösterdiğini ve pratik uygulamalar için potansiyel bir çözüm sunduğunu göstermektedir. Şekil 15’de IoT tabanlı sağlık hizmeti sisteminde yarı denetimli GAN yaklaşım modeli gösterilmiştir.



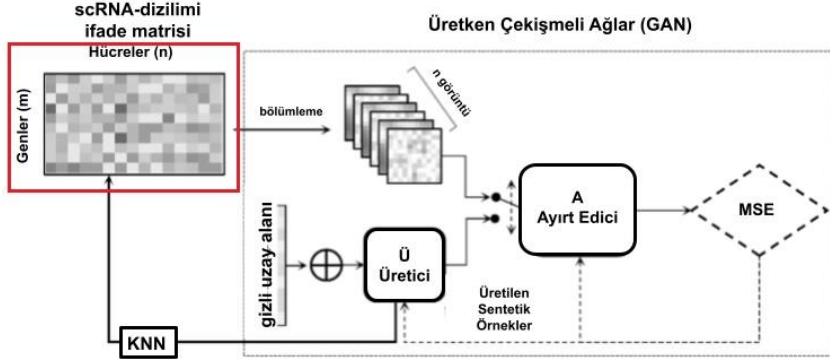
Şekil 15. IoT tabanlı sağlık hizmeti sisteminde yarı denetimli GAN yaklaşım modeli (Yang & ark. 2018)

GAN'ın Biyoinformatikte Uygulanması

Biyoinformatik, bilgi teknolojisi, biyoloji, uygulamalı matematik ve istatistik alanlarını birleştiren disiplinler arası bir bilim dalıdır (Shahnaz & ark. 2016). Bu bilim dalı, biyolojik verilerin toplanması, depolanması, analiz edilmesi, yorumlanması ve kullanılması için bilgisayar bilimleri ve istatistiksel yöntemlerin kullanılmasıyla ilgilidir. Biyoinformatik, genetik, genomik, proteomik ve diğer biyolojik disiplinlerde büyük veri setlerinin işlenmesi ve anlamlı bilgilerin çıkarılması için bilgisayar tabanlı teknikleri içerir (Mount, 2004). DNA, RNA, proteinler ve metabolitlerin moleküler dizileri ve diğer tüm genom verileri dahil olmak üzere büyük hacimli biyolojik verileri keşfetmek için geleneksel istatistikler, modern bilgisayar bilimi, makine öğrenimi ve diğer modelleme algoritmalarını uygular (Abdurakhmonov, 2016). Biyoinformatik araştırmacıları için en önemli ve zor konulardan biri, büyük veri kümelerinin erişilebilirliği ve kullanılabilirliğidir. Artan verim ve teknolojik gelişmeler DNA'ların yeni nesil dizileme teknikleri kullanılarak analiz etme stilini değiştirmiş olsa da dizileme maliyeti hala nispeten yüksektir. Ayrıca, araştırma amaçlı verilerin erişilebilirliği birçok yasal ve etik sorunu içerdiğinden, biyoinformatik veriler oldukça hassastır (Sharmilan &

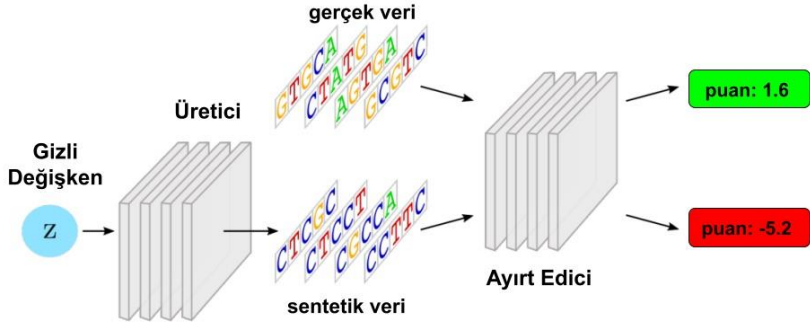
Chaminda, 2017). Mevcut biyolojik örneklerin eksikliği dengesiz veri kümelerine neden olabilir ve bu da aşırı uyum sorunlarına ve zayıf sınıflandırma performansına yol açabilir. Son zamanlarda, araştırmacılar bu sorunların üstesinden gelmek ve veri örnekleri oluşturmak için GAN modellerini kullanmışlardır. GAN tabanlı biyoinformatik uygulamalara yönelik çalışmalardan bazıları aşağıda açıklanmıştır.

(Xu & ark. 2020), GAN mimarisini, teknik gürültü, kesinti kaynağı gibi sınırlamalardan kaçınmak için gerçekçi hücreler kullanan tek hücreli RNA dizilimini (scRNA), dengeleyen scRNA dizilimi (scIGAN'lar) için önerdiler. scIGAN'lar, tek tek sırasıyla hücrelerin ifade profillerini görüntülere dönüştürür ve bunları GAN'a besler. Eğitilmiş üretken çekişmeli ağ, tanımlanmış türlerin gerçekçi hücrelerini temsil eden sentetik ifade profilleri üretir. Gözlemlenen hücreler yerine üretilen hücreler daha sonra gerçek hücrelerin düşüşlerini empoze etmek için kullanılır. scIGAN'lar, genetik bir yapıdan kaynaklanan düşük yapı ve aktiviteler ve/veya hücreden hücreye varyansa sahip çok az gen içeren küçük veri kümeleri için sağlamdır. scIGAN'lar, farklı scRNA-dizilimi protokollerinden gelen veri kümelerinde eşit derecede iyi çalışır ve 100.000'den fazla hücreye sahip veri kümelerine ölçeklenebilir. Yazarlar, scIGAN'ların yalnızca biyolojik veriler üzerine yapılan bilimsel çalışmalarda GAN'ların bir uygulaması olmadığını, aynı zamanda scRNA-dizilimi verileri için rakip bir atama yöntemini temsil ettiğini birçok yönden ikna edici kanıtlarla gösterdiler. Tek hücreli RNA-dizilimi ataması ve sistematik değerlendirmeler için üretken çekişmeli ağ modeli Şekil 16'da gösterilmiştir.



Şekil 16. scIGAN'ın üst düzey mimarisi (Xu & ark. 2020)

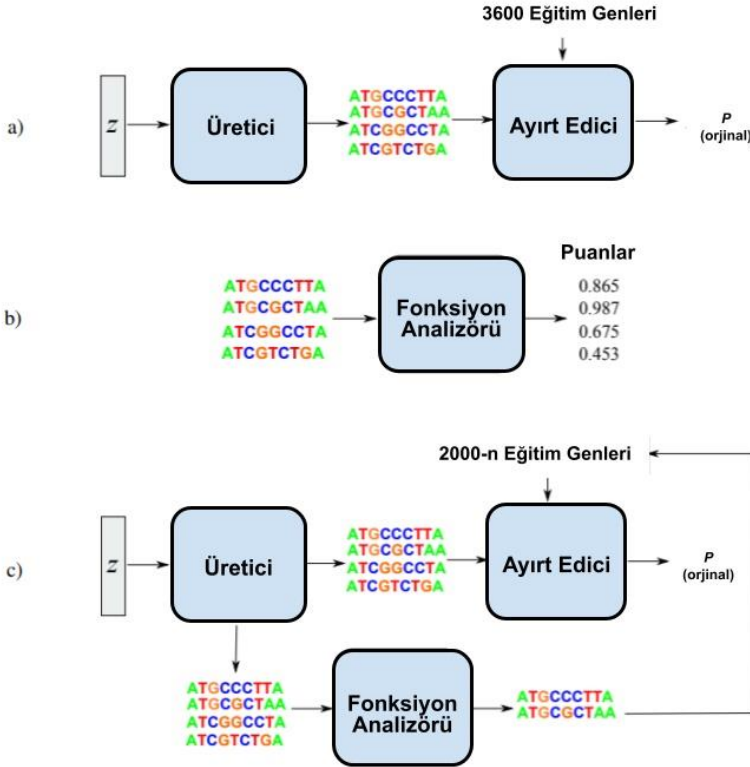
(Killoran & ark. 2017), karakterlerin ayrık dizilerini DNA'nın nükleotid olarak adlandırılan temel yapı birimlerinden (A, C, G, T) tek sıcak kodlamalarını sürekli bir temsile kodlayarak yeni DNA dizileri oluşturmak için WGAN tabanlı derin bir üretken çekişmeli ağ modelini sundular. Yazarlar, insan genomu hg38'in kromozom 1'ini kapsayan 4.6M 50 nükleotid uzunluğundaki dizilerden oluşan bir veri kümesine eğitilmiş bir üretici modeli eklediler. Maksimum aktivasyon versiyonunu genişleten ve DNA gibi ayrık diziler için uygun olan ortak bir yaklaşım önerdiler. Üretken çekişmeli ağ modelinin DNA dizilerinden önemli yapıları öğrenebileceğini ve istenen özelliklere sahip yeni DNA dizilerini keşfetmek ve tasarlamak için kullanılabileceğini buldular. Şekil 17'de yeni DNA dizileri oluşturmak için önerilen WGAN mimarisinin temel şeması gösterilmiştir. Üretici ve Ayırt edicinin her ikisi de derin sinir ağlarında parametre olarak derecelendirilmiştir. Açıklık sağlamak amacıyla kısa ve kesin veri dizileri gösterilmiştir. Uygulamada, gerçek veriler tek sıcak vektörler kullanılarak kodlanır ve üretici bu kodlama ile sürekli bir yaklaşım öğrenir.



Şekil 17. Sentetik DNA dizileri oluşturmak için önerilen WGAN mimarisi
(Killoran & ark. 2017)

(Gupta & Zou 2018), harici bir fonksiyon analizörü kullanarak istenilen özelliklerin sentetik gen dizilerini optimize etmek için geri beslemeli GAN diğer adıyla FBGAN adı verilen yeni bir geri bildirim döngü mimarisini önerdiler. Geri bildirim döngü modeli, GAN ve farklılaştırılabilir bir sinir ağı dahil olmak üzere iki bileşenden oluşur. GAN, yeni gen dizileri oluşturmak için kullanıldı. Analizör adlı diferansiyellenebilir sinir ağı bir gen dizisini, bağışıklık sisteminin korunmuş bileşeni olarak tanımlanan antimikrobiyal peptit (AMP) kodlama olasılığına dönüştürdü. En üst sıralarda yer alan n olumlu oluşturulmuş dizi, ayırt edici eğitim veri kümesinde bulunan en eski n genin yerini aldı. Yazarlar bu modelle, uzunluğu 50 amino aside kadar olan peptitleri kodlayan sentetik genler üreterek elde edilen peptitlerin ikincil alfa-sarmal yapısını optimize ettiler. Şekil 18’de gösterilen FBGAN mimarisinde; a) Gen dizilerini üretmek için kullanılan GAN modelinin genel eğitim mekanizması. Burada kullanılan eğitim seti, uzunluğu 50 amino asitin altındaki protein sekansı ve fonksiyonel bilgilerin serbestçe erişilebilen bir veri tabanı olarak bilinen UniProt kodlayan 3600 genden oluşur. b) Fonksiyon analizörün bir sıra alarak puan üreten genel formu. Analizör, derin sinir ağından laboratuvara kadar bu çerçeveye uyan herhangi bir model olabilir. c) FBGAN'daki yeni

geri bildirim döngüsü eğitim mekanizması. Her aşamada, üreticiden çeşitli tahminler örneklenir ve analizöre girilir. Analizör, b'de gösterildiği gibi her diziyeye bir puan verir ve en yüksek puanı alan diziler seçilir. Bu yüksek puanlama dizileri, ayırt edicide gerçek veriler olarak geri girilir. Analizörden seçilen n dizi, ayırt edicinin gerçek eğitim veri setindeki önceki n dizinin yerini alır. Bu şekilde yavaş yavaş ayırt edicinin gerçek verilerinin yerini, analizörden yüksek puanlar alan sentetik veriler alır.



Şekil 18. Geri Beslemeli GAN (FBGAN) mimarisi (Gupta & Zou 2018)

Sonuç

GAN tabanlı modellerin tıbbi görüntü analizinde veri eksikliği için iyi bir çözüm sunduğunu söyleyebiliriz. GAN mimarisine dayalı modeller, aynı modalitedeki görüntülerin çeşitliliğini ve miktarını artırmak için veri büyütme yöntemleri olarak daha fazla kullanılmaktadır. Bu temelde, Cycle-GAN tabanlı modeller, birden fazla modalite arasında çevirileri mümkün kılar. Cycle-GAN modellerinde segmentasyon ağlarına dahil olarak, çapraz modalite segmentasyonları uçtan uca öğrenilebilir. Bu, tıbbi görüntü analizinde derin öğrenme algoritmalarının uygulanmasını ve geliştirilmesini etkili bir şekilde teşvik edecektir. Bununla birlikte, GAN modellerinde hala sınırlamaları mevcuttur. Örneğin, kafadan alınan BT görüntüleri, karın BT görüntülerinden veya bacaklardan alınan MR görüntülerinden oluşturulamaz. Araştırmacıların, yakınsama veya model çökmesi gibi sorunlardan kaçınmak için veri akışlarını ve kayıp işlevlerini dikkatli bir şekilde tasarlamaları gerekir. Ek olarak, bazı tıbbi görüntü analizi görevleri, organın ayrıntılı 3B bilgisini gerektirir. GAN tabanlı modele 3B özellik öğrenimini veya 3B segmentasyonu dahil etmek zor olacaktır. ETK verilerine dayanarak GAN tabanlı modellerin, tıbbi bilişimde daha az etiket ve dengesiz sınıflandırma sorunlarını çözmek için gerçekçi sentezlenmiş ayrık veriler, sürekli veriler ve hatta zaman serisi verileri üretebilir olsa da yine de bazı sınırlamaları bulunmaktadır. Örneğin üretilen bu verilerin nasıl değerlendirilip raporlanacağı ve üretilen bu verilerin tıbbi sorunları çözmek için nasıl uygulanacağı tartışmalıdır ve bunu doğrulamak için gerçek veriler gerektirir. GAN modellerinin biyoinformatikte uygulanması heyecan verici olsa da halen erken bir gelişim aşamasındadır. Bu alanda yapılan çalışmaların çoğunda, veri kümeleri oluşturmak ve/veya artırmak için GAN veya geliştirilmiş modeller uygulanmaktadır. Sonuç olarak, Biyomedikal Bilişim Uygulamalarında Yapay Zekanın Üretken Yüzü (B-GAN) sayesinde, büyük veri setleri üzerinde hızlı ve hassas analizler gerçekleştirilebilmekte, bu da tanı ve tedavi süreçlerini önemli ölçüde iyileştirmektedir. Hastalıkların erken teşhisi, kişiselleştirilmiş tedavi planları ve sağlık hizmetlerinin daha

etkin bir şekilde yönetilmesi gibi konularda yapay zekâ uygulamalarının sağladığı avantajlar, biyomedikal bilişim alanında önemli bir dönüşümü beraberinde getirmiştir. Biyomedikal bilişimde kullanılan yapay zekâ, sağlık sektöründe sadece teknolojik bir yenilik olmanın ötesinde, kaliteli ve etkin hizmet sunumunu destekleyen bir araç haline gelmiştir. Yapay zekâ destekli biyomedikal bilişim uygulamalarının gelecekteki potansiyeli, hastaların yaşam kalitesini artırmak ve sağlık profesyonellerine daha etkili kararlar almalarında yardımcı olmak adına büyük bir umut vadetmektedir.

KAYNAKÇA

Abdurakhmonov, I. Y. (2016) Bioinformatics: basics, development, and future. *Intech Open*, 3–26. Doi: 10.5772/63817.

Arjovsky, M., Chintala, S., & Bottou, L. (2017) Wasserstein generative adversarial networks. *In International conference on machine learning*, 214–223, PMLR.

Ataş, İ. (2023a) Comparison of deep convolution and least squares GANs for diabetic retinopathy image synthesis. *Neural Computer & Application*. 35,19 14431–14448. Doi: 10.1007/s00521-023-08482-4.

Ataş, İ. (2023b) Image Caption Generator with CNN and LSTM. *9th International New York Conference on Evolving Trends in Interdisciplinary Research & Practices*, 9,2 61–69. ISBN: 978-1-955094-52-8.

Cheng, K. et al. (2020) An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset. *Springer, Berlin*, 13725–13752.

Choi, Y. et al. (2018) StarGAN: unified generative adversarial networks for multi-domain image-to-image translation. *In: Proceedings of the IEEE conference on computer vision and pattern recognition*, 8789–8797.

Choi, E. et al. (2017) Generating multi-label discrete patient records using Generative Adversarial Networks. *arXiv:1703.06490*.

Deng, J. et al. (2009) ImageNet: a large-scale hierarchical image database. *In: IEEE Conference on Computer Vision and Pattern Recognition. Miami, FL*. Doi: 10.1109/CVPR.2009.5206848.

Dirvanauskas, D. et al. (2019) HEMIGEN: human embryo image generator based on Generative Adversarial Networks. *Sensors* 19, 3578. Doi: 10.3390/s19163578.

Esteban, C. Hyland, S.L. & Rätsch, G. (2017) Real-valued (Medical) time series generation with recurrent conditional GANs. *arXiv:1706.02633*.

Feng, X. et al. (2018) Topic-to-essay generation with neural networks. *In: IJCAI, Stockholm*. 4078–84. Doi: 10.24963/ijcai.2018/567.

Frid-Adar, M. et al. (2018) GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*. 321, 321–31. Doi: 10.1016/j.neucom.2018.09.013.

Gonog, L. & Zhou, Y. (2019) A review: generative adversarial networks. *In: 2019 14th IEEE conference industrial electronics and applications (ICIEA)*, 505–510. IEEE.

Goodfellow, I. et al. (2014) Generative adversarial nets. *In Proceedings of the 27th International Conference on Neural Information Processing Systems Montréal., Canada*, 2672–2680. Red Hook, NY: Curran Associates.

Guan, J. et al. (2018) Generation of synthetic electronic medical record text. *arXiv:1812.02793v1*. 374–80. Doi: 10.1109/BIBM.2018.8621223.

Gupta, A. & Zou, J. (2018) Feedback GAN (FBGAN) for DNA: a novel feedback-loop architecture for optimizing protein functions. *arXiv:1804.01694*.

Hiasa, Y. et al. (2018) Cross-modality image synthesis from unpaired data using Cycle GAN: effects of gradient consistency loss and training data size. *In: Simulation and Synthesis in Medical Imaging - Third International Workshop, SASHIMI*, Springer.

Hongfei, S. et al. (2022) Synthesis of pseudo-CT images from pelvic MRI images based on MD-CycleGAN model for radiotherapy, *Phys. Med. Biology*, 67, 035006. Doi: 10.1088/1361-6560/ac4123.

Huo, Y. et al. (2017) Adversarial synthesis learning enables segmentation without target modality ground truth, *arXiv:1712.07695*. Doi: 10.1109/ISBI.2018.8363790.

Isola, P. et al. (2017) Image-to-image translation with conditional adversarial networks, *In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA*, 5967–5976.

Killoran, N. et al. (2017) Generating and designing DNA with deep generative models. *arXiv:1712.06148*.

Lan, L. et al. (2020) Generative Adversarial Networks and Its Applications in Biomedical Informatics. *Front. Public Health*, 8, 1–14. Doi: 10.3389/fpubh.2020.00164.

Lin, Y. et al. (2018) The new Frontier of AI research: generative adversarial networks. *Acta Autom Sin.* 44, 775–92. Doi: 10.16383/j.aas.2018.y000002.

Lin, T. Y. et al. (2014) Microsoft COCO: common objects in context. *arXiv:1405.0312*. Doi: 10.1007/978-3-319-10602-1_48.

Li, Q. et al. (2019) A framework for big data governance to advance RHINs: a case study of China. *IEEE Access.* 7:50330–8. Doi: 10.1109/ACCESS.2019.2910838.

Mahapatra, D. & Bozorgtabar, B. (2017) Retinal vasculature segmentation using local saliency maps and Generative adversarial networks for image super resolution. *arXiv:1710.04783*.

Mirza, M. & Osindero, S. (2014) Conditional generative adversarial nets. 2672–80. *arXiv:1411.1784*.

Mount, D. W. (2004) Bioinformatics: Sequence and Genome Analysis. *2nd edn*. New York, NY: Cold Spring Harbor Laboratory Press.

Pavan, Kumar, M.R. & Jayagopal, P. (2020) Generative adversarial networks: a survey on applications and challenges. *Int J Multimed Info Retr*, 10, 1–24. Doi: 10.1007/s13735-020-00196-w.

Shin, H. C. et al. (2016) Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans Med Imaging*. 35,1285–1298. Doi: 10.1109/TMI.2016.2528162.

Shitrit, O. & Raviv, T. R. (2017) Accelerated magnetic resonance imaging by adversarial neural network. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Doi: 10.1007/978-3-319-67558-9_4.

Shahnaz, M. et al. (2016) Bioinformatics: an overview for cancer research. *J Drug Deliv. Ther.*, 6, 69–72. Doi: 10.22270/jddt.v6i4.1290.

Sharmilan, S. & Chaminda, H. T. (2017) Generate bioinformatics data using Generative Adversarial Network: a review. In: *2nd International Conference on Information Technology Research. Colombo*.

Tanner, C. et al. (2018) Generative Adversarial Networks for MR-CT Deformable Image Registration, *arXiv:1807.07349*.

Wang, K. et al. (2017) Generative adversarial networks: introduction and outlook, *IEEE/CAA J Autom Sin.* 4, 588–98. Doi: 10.1109/JAS.2017.7510583.

Wang, S. H. et al. (2019) Alcoholism identification based on an AlexNet transfer learning model, *Front Psychiatry*. 10, 205. Doi: 10.3389/fpsy.2019.00205.

Wolterink, J. M. et al. (2017) Deep MR to CT synthesis using unpaired data, *arXiv:1708.01155*. Doi: 10.1007/978-3-319-68127-6_2.

Yang, Y. et al. (2019) GAN-based semi-supervised learning approach for clinical decision support in health-IoT platform. *IEEE Access*. 7:8048–57. Doi: 10.1109/ACCESS.2018.2888816.

Xian, W. et al. (2017) A survey of image synthesis and editing with generative adversarial networks, *Tsinghua Sci Technol.* 22, 660–74. Doi: 10.23919/TST.2017.8195348.

Xu, Y. et al. (2020) Single-cell RNA-seq imputation using generative adversarial networks. *bioRxiv*. Doi: 10.1101/2020.01.20.913384.

Zhang, Z. et al. (2019) Semantic prior guided face inpainting. *In: MM Asia '19. New York, NY*.

Zhang, F. et al. (2023) Medical applications of generative adversarial network: a visualization analysis. *Acta Radiol.* 64, 10, 2757-2767. Doi: 10.1177/02841851231189035.

Zhu, JY. et al. (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. *In: IEEE International Conference on Computer Vision. Venice*.

BÖLÜM V

Çelik Boru Kaynaklarında Oluşan Hataların Tespitinde Yeni Yöntemler

Murat MERAL¹
Kamil ORMAN²

Giriş

Doğalgaz ve petrol hatları başta olmak üzere otomotiv, havacılık, tersane, savunma sanayi gibi birçok sektörde çelik malzemelerin birbirlerine kaynatılması işlemlerinde kaynaklarda ki hataların tespiti Tahribatsız Muayene Testi (NDT: Non Destructive Testing) yöntemleri ile yapılmaktadır. Tahribatsız muayene yöntemleri ile kaynaklı imalatın herhangi bir hata içerip içermediği belirlenir. Bu yöntemler ile test edilen yüzey üzerinde hiçbir iz bırakılmadığından dolayı tamamlanmış kaynak üzerine uygulanarak

¹ Yüksek Lisans Öğrencisi, Erzincan Binali Yıldırım Üniversitesi, muratmeral@msn.com (Orcid: 0009-0003-1663-7591)

² Doç. Dr., Erzincan Binali Yıldırım Üniversitesi, Bilgisayar Mühendisliği Bölümü (Orcid: 0000-0002-7236-9988)

kaynakların güvenirliliği ve kalitesi hızlı ve uygun maliyetli bir şekilde tespit edilir.

Doğalgaz ve petrol boru hatlarında ki kaynak hatalarının tespiti için tahribatsız muayene yöntemleri arasında en çok kullanılanı ise Radyografik muayene (RT: Radiographic Testing) yöntemidir. Radyografik muayene yöntemiyle elde edilen negatif filmler NDT Uzmanları tarafından uluslararası standartlara göre değerlendirilerek oluşan Kaynak Hata türleri tespit edilir. Tespit edilen kaynak hataları ilgili standartlara göre değerlendirilir ve ilgili kaynağın kesilmesine (C/O: Cut Out), tamir edilmesine (R: Repair) ve ya ilgili standartlarda belirtilen değerlerin altında olması durumunda işlem yapılmayıp kabul (OK) edilmesine karar verilir. Ancak Değerlendirme sırasında gözden kaçabilecek durumlar olması olası bir durumdur. Bu sebeple Radyografik Filmlerin değerlendirilmelerinin yapay zeka destekli (Akar ve ark., 2023) görüntü işleme algoritmaları ile desteklenmesi daha doğru kararlar verilmesini destekleyebilir ve Doğalgaz / Petrol boru hatları başta olmak üzere kaynak işleminin uygulandığı tüm sektörlerde daha sağlıklı ve güvenli imalatların gerçekleştirilmesi yönünde sektöre katkıda bulunulabilir (Feng ve ark., 2016).

Literatürde kaynak hatalarının tespiti için görüntü işleme yöntemleri son yıllarda gelişen görüntüleme cihazları ve teknikleri sayesinde yaygın olarak kullanılmaktadır (Chen ve ark., 2021, Kafieh ve ark., 2011, Madani ve Azizi, 2015, Akar ve Aydın, 2017, Doğan ve Ark. 2022). Bu çalışmalarda kaynak hatalarını tespit etmek için farklı algoritmalar kullanılmıştır . Önerilen bir diğer çalışmada iki farklı algoritma geliştirilmiştir. İlk önce uzmanlardan, ders kitaplarından ve uluslararası standartlardan toplanan önceden tanımlanmış bir tanımlama ağacına dayanarak kusur türleri belirlenmiş ve sonra ise belirlenen kusurlar için uluslararası standartlara göre kabul kararı veren diğer bir karar destek algoritması kullanılmıştır (Shafeek ve ark., 2004). Boaretto ve Centeno tarafından kaynaklarda oluşan gözenekleri kusurlu ve kusur olarak sınıflandırma işlemi yapan çok katmanlı sınıflandırma algoritmaları kullanılarak tamirat gerektiren ve gerektirmeyen

kaynak hataları belirlenmiştir (Boaretto ve Centeno, 2017). Bir diğer kaynak kusuru tespiti yapılan çalışmada ise yapay bağışıklık sistemi ve derin öğrenme teknikleri kullanılarak kaynağın kusurlu yada kusursuz olarak sınıflandırılması yapılmıştır (Fioravanti ve ark., 2019).

Tahribatsız Muayene Yöntemleri

Tahribatsız muayenede özellikle hata bulunduktan sonra bu hatanın giderilmesi için gerekli işlemler standartlarda sunulmaktadır. Tahribatlı muayenin aksine, muayene parçası sınırlaması yoktur. Gerekğinde kaynak dikişlerinin %100 muayenesi mümkündür. Boru hatlarının kaynaklı birleştirilmesine radyografik muayene en yaygın olarak uygulanan yöntemdir.

Tahribatsız muayene yöntemleri (Shull, 2002);

- Görsel Muayene
- Radyografik muayene
- Ultrasonik muayene
- Manyetik partikül muayenesi
- Sıvı penetrant (boya) muayenesi
- Girdap akımları muayenesi
- Akustik emisyon muayenesi

Tahribatsız Muayene (NDT) yöntemiyle bulunan başlıca hatalar;(Luo ve ark., 2014)

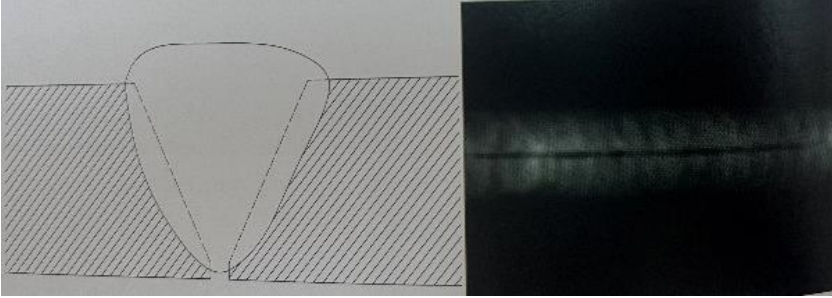
- Malzemede bulunan çatlaklar, boşluklar, gözenekler, hadde yönünde uzamış hatalar, laminasyonlar, cüruf veya tungsten gibi yabancı maddeler
- Kaynak hataları, döküm hataları, dövme hataları
- Et kalınlığı ölçümleri, kalınlık farklılıkları, iç korozyon, dış korozyon, aşınma

- Yüzey hataları, yüzey pürüzlülüğü, gözenek
- Malzeme farklılıkları, yoğunluk farklılıkları, boyut farklılıklarıdır.

Kaynak Hata Türleri

Yüksek-Düşük olmaksızın Yetersiz Penetrasyon (IP) : Kaynak kökünün tamamlanmamış dolgusudur. Bu durum Şekil-1 de şematik olarak gösterilmiştir. Aşağıdaki koşullardan herhangi birisi mevcut ise IP bir kusur olarak dikkate alınmalıdır (American Petroleum Institute,1995).

- Bir IP belirtisinin boyu 1 inç (25 mm)'i aşarsa;
- Birleşik IP belirtilerinin toplam boyları, 12 inç (300 mm) uzunluğundaki kaynak içerisinde 1 inç (25mm)'i aşarsa,
- Birleşik IP belirtilerinin toplam boyları 12 inç (300mm)'den daha kısa kaynak boylarında kaynak boyunun %8'ini geçerse; Yetersiz penetrasyondan bahsedilir.

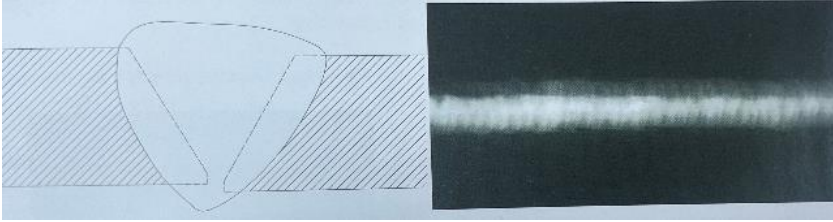


Şekil-1 Yüksek-Düşük olmaksızın Yetersiz Penetrasyon (IP) (Türkel, 2012)

Yüksek düşüğe bağlı yetersiz penetrasyon (IPD): Kökün bir kenarının, bitişikteki boru veya fittings'in yanlış hizalanmasından dolayı açıkta (veya bağımsız) kaldığı durum şeklinde tanımlanır. Bu

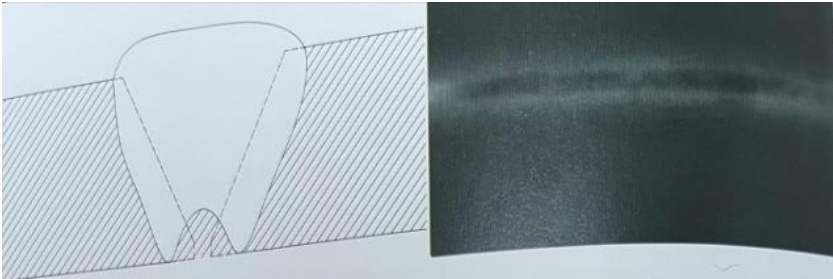
durum Şekil-2 de şematik olarak gösterilmiştir. Aşağıdaki koşullardan herhangi birisi mevcut ise IPD bir kusur olarak dikkate alınmalıdır. (American Petroleum Institute,1995)

- Bir IPD belirtisinin boyu 2 inç (50 mm)’i aşarsa;
- Birleşik IPD belirtilerinin toplam boyları, 12 inç (300 mm) uzunluğundaki kaynak içerisinde 3 inç (75 mm)’i aşarsa;



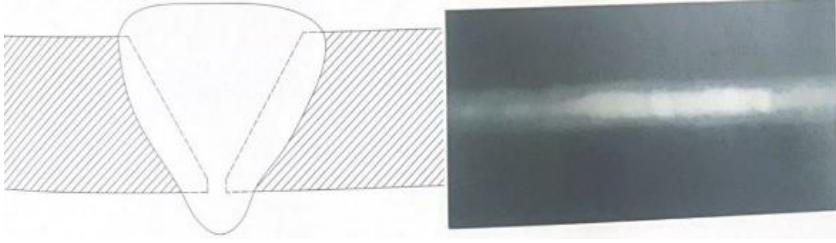
Şekil-2 Yüksek – Düşüğe bağlı Yetersiz Penetrasyon (Türkel, 2012)

Dahili İç Bükeylik: Düzgün bir şekilde ergimiş ve pahın her iki tarafına doğru boru cidar kalınlığı boyunca tam nüfuz etmiş fakat merkezi boru cidarının oldukça içerisinde olan pasodur. Konkavlık şiddeti boru cidar yüzeyinin aksenal uzantısı ile kaynak paso yüzeyinin en düşük noktası arasındaki dikey mesafedir.



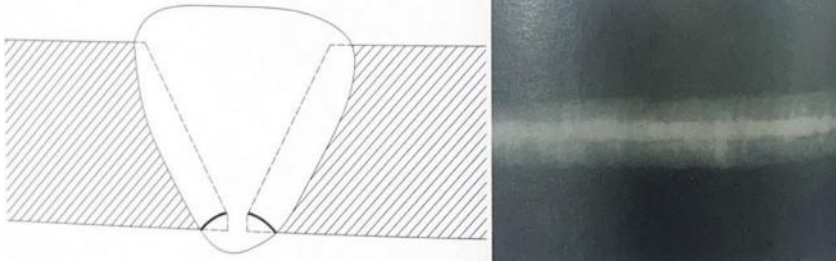
Şekil-3 Dahili iç bükeylik (Türkel, 2012)

Kaynak Taşması (BT): Kök pasonun bir kısmının, aşırı penetrasyon sebebiyle kaynak dolgusunun borunun içerisine doğru üflenmesidir.



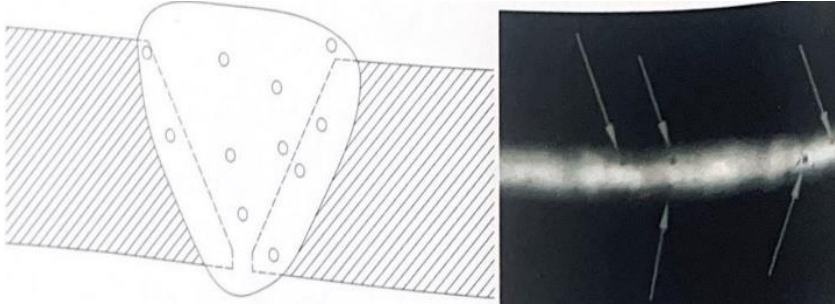
Şekil-4 Kaynak Taşması (Türkel, 2012)

Cüruflar: Cüruf enklüzyonu kaynak metali veya kaynak metali ile ana metal arasında sıkışık kalan metal olamayan katı şeklinde tanımlanır. Uzamış cüruf Enklüzyonları (ESIs) genellikle ergime bölgesinde bulunur. İzole edilmiş cüruf enklüzyonları (ISIs) düzensiz şekildedirler ve kaynak içerisinde herhangi bir yerde bulunabilirler.



Şekil-5 Cüruflar (Türkel, 2012)

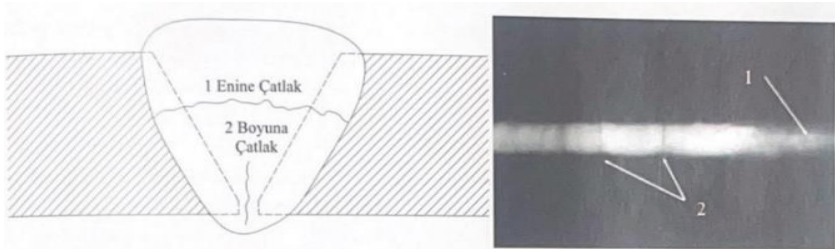
Gözeneklilik: Ergimiş haldeki kaynak bölgesinden yükselerek yüzeyden dışarı çıkma şansı bulamayıp, katılaşmakta olan kaynak metali içerisine hapsolmuş gaz olarak tanımlanır. Gözenekler genellikle küreseldir fakat bazen de uzamış veya düzensiz bir şekilde de olabilirler, örneğin boru (solucan deliği) gibi olan gözenekler.



Şekil-6 Gözenekler (Türkel, 2012)

Çatlaklar: Aşağıdaki durumlardan herhangi birisi söz konusu ise çatlaklar (C) bir kusur olarak değerlendirilir. (American Petroleum Institute, 1995)

- Herhangidir boyutta ve ya kaynağın herhangi bir yerindeki çatlak sıg krater çatlak veya yıldız çatlak değil ise
- Sıg krater çatlakı veya yıldız çatlak olup boyu 5/32 in.(4 mm)'i geçiyorsa



Şekil-7 Çatlaklar (Türkel, 2012)

Kaynak Hatalarının Değerlendirilmesi

Radyografik Muayene filmlerinde ki hatalar Endüstriyel sektörde uluslararası bir standart olan API-1104'e göre değerlendirilir (American Petroleum Institute, 1980). Bu standartta Radyografik filmlerin ne şekilde değerlendirileceği maddeler halinde aşağıdaki gibi belirlenmiştir.

- Bir gözenegin sapı 1.5 mm'yi geçmemelidir.

- Dağınık bir cürufun genişliği 3 mm den büyük olmamalı. En yakın cürufun da en az 50 mm uzakta olması gerekir.
- Kök paso şişkinliğinin (Taşkınlığının) 2 mm'yi aşmaması.
- İç bükey bir kök paso olması durumunda cidar kalınlığının % 10'u değerinde kabul edilebilir bir iç bükeylik olabilir.
- Kaynak dikişinde rastlanabilecek herhangi bir çatlak kabul edilemez ve bu dikişlerin tamir edilmesi gerekmektedir.
- Petrol borularının kaynaklarında yan cidara kaynamama ve yetersiz nüfuziyet kabul edilmez bir hatalardır.

Çelik boru kaynaklarında ki hatalar bir süreksizlik sonucu ortaya çıkar. Kaynak sırasında oluşan gaz boşlukları, yabancı madde kalıntıları, eksik ergime ve kaymalar süreksizlik oluşturur. Bu süreksizlikler Radyografik muayene filmlerinde ani renk değişimi olarak gözlemlenir. Bu sebeple çeşitli görüntü işleme yöntemleri kullanılarak bu süreksizlikler üzerine yoğunlaşarak çelik boru kaynak hataları tespit edilebilmektedir.

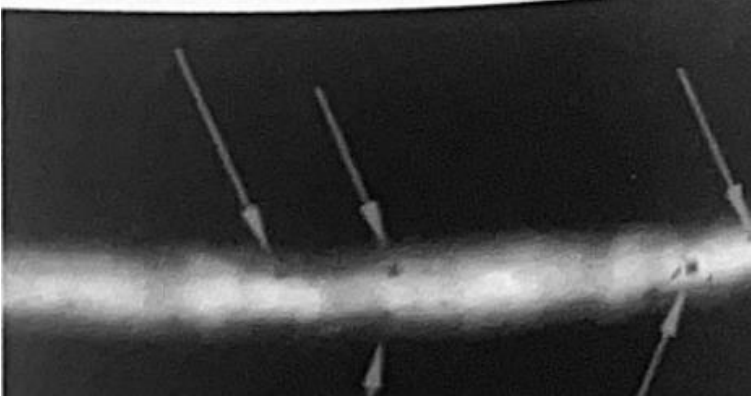
Görüntü İşleme Algoritmaları ile Gözenek Oluşumu Kaynak Hatasının Tespiti

API-1104'de belirtilen ilgili hata türlerinin boyutlarına göre ayırım yapabilmek için görüntü işleme yöntemlerindeki kenar bulma algoritmaları ile bulunan gözeneklerin alanı tespit edilir. Bu alan sınır değerlerden yüksek ise farklı bir renk ile işaretlenerek Radyografik Muayene filmlerini değerlendirecek olan NDT uzmanını uyarması amaçlanmaktadır.

Bu bölümde RT filmlerinden alınan görüntülerde kusurları belirlemede NDT Uzmanlarına yardımcı olmak için seçilen görüntü işleme algoritmaları ile gözden kaçabilecek gözenek kusurlarının tespiti amaçlanmış ve bir uygulamaya yer verilmiştir.

Renk Formati:

Saha uygulamalarında radyasyon cihazı ile kaynak üzerine konulan filme radyasyon gönderilerek çekim işlemi yapılır. Çekim işlemi yapıldıktan sonra gerekli banyo işlemleri yapılarak kaynak filmlerinin negatif görüntüleri elde edilir. Negatif görüntüler gri görünmesine rağmen bilgisayar ortamında RGB formatında algılanmaktadır. Radyografik muayene filmlerinin üzerinde işlem yapabilmemiz için öncelikle görüntülerimizi gri renk formatına sonrasında ise siyah-beyaz renk formatına çevirmemiz gerekmektedir.



Şekil-8 Gri Renk Formatı



Şekil 9 Siyah Beyaz Renk Formatı

Görüntü Kalitesini Düzenleme:

Literatürde görüntü işleme algoritmaları ile kusurların tespit edilmesine yönelik çalışmalarda, kaynak yapılan parçalardan elde edilen görüntü de bulunan gürültüleri ortadan kaldırmak için medyan filtre gibi filtreleme teknikleri uygulanmıştır. Histogram eşitleme işlemi ile görüntünün kontrastı artırılmıştır. Ayrıca gama düzeltme tabanlı bir aydınlatma algoritması ile eşit olmayan aydınlatma koşullarını dengelemek için kullanılmıştır. (Sözeri V.& Keskin O. 2023)

Kenar Bulma Algoritmaları:

Çeşitli işlemlerden geçirdiğimiz görüntüde canny, sobel, prewitt gibi kenar bulma algoritmaları kullanarak bulunan süreksizlikleri belirlenmiştir.

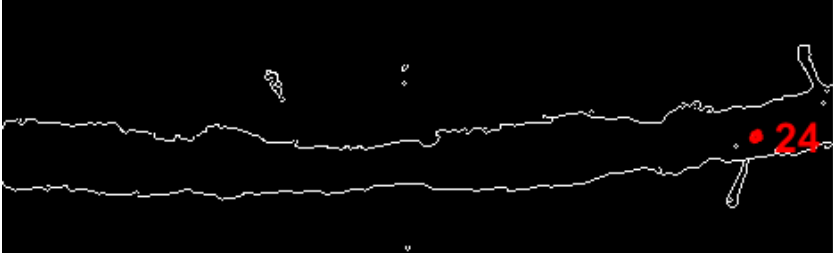


Şekil-10 Canny Kenar Bulma algoritması ile elde edilmiş görüntü

Karar Verme:

Kenar bulma algoritmaları ile kaynak üzerinde görünen hataların, Radyografik Muayene filmlerini değerlendirmede uluslararası bir standart olan API-1104'e (American Petroleum Institute) göre ilgili kaynağın kesilmesine (C/O), tamir edilmesine (R) veya belirtilen değerlerin altında olması durumunda işlem yapılmayıp kabul (OK) edilmesine karar verebilmek için hatanın büyüklüğünün hesaplanması gerekmektedir. Bunun içinde elemsize, boundary vs. alan hesaplama algoritmaları kullanılır. Bu algoritmalar ile hatanın alanını hesapladıktan sonra hatanın boyutu

istenilen sınırları aşması durumunda kırmızı (isteğe göre), kabul edilebilir sınırlar dahilinde olması durumunda ise yeşil (isteğe göre) işaretlenerek NDT uzmanının karar vermesine destek olunur.



Şekil-11 Hatanın Büyüklüğü

Sonuç ve Tartışma

Doğalgaz ve petrol hatları başta olmak üzere otomotiv, havacılık, tersane, savunma sanayi gibi birçok sektörde kaynaklı işlem yapılmakta ve bu kaynakların büyük çoğunluğu Radyografik muayene ile kontrol edilmektedir. NDT Uzmanları Radyografik filmleri kontrol ederken karanlık bir odada çok dikkatli incelemeleri gerekmektedir. Radyografik filmlerde görüntü kalitesi, film yoğunluğu ve filmleri incelemede kullanılan viewer'ın ışık şiddetine dikkat edilmesi gerekmektedir. Gözden kaçabilecek bir hatanın maliyeti çok yüksek sonuçlar doğurabileceği gibi insan hayatını etkileyecek telafisi mümkün olmayan sonuçlarda doğurabilir.

Bu çalışmada görüntü işleme yöntemleri kullanılarak kaynak hata türlerinden gözenek hatasının tespiti için bir örnek uygulama sunulmuştur. Bu bağlamda diğer kaynak hata türlerinin de tespitine yönelik çalışmalar yapılarak NDT Uzmanlarının karar vermesine yardımcı olacak bir program geliştirilmesi amaçlanmaktadır. Bu sayede kaynaklı işlem yapılan tüm sektörlerde hatalar en düşük seviyeye indirilmiş olacaktır.

Kaynaklar

Akar, F., & Aydın, Y. (2017). Comparison of Interest Point-Based Features in Object Recognition Applications. In *8th International Advanced Technologies Symposium (IATS'17)*, Elazığ, Türkiye 19(22), 3553-3556.

Akar, F., Katipoğlu, O. M., Yeşilyurt, S. N., & Taş, M. B. H. (2023). Evaluation of tree-based machine learning and deep learning techniques in temperature-based potential evapotranspiration prediction. *Polish J Environ Stud*, 32, 1009-1023.

API-American Petroleum Institute, (1980). API Standard 1104, (1980). Fifteenth Edition.

API-American Petroleum Institute, (1995). API Specification 5L, *Specification for linepipe*, s.3-7.

Boaretto, N., & Centeno, T. M. (2017). Automated detection of welding defects in pipelines from radiographic images DWDI. *Ndt & E International*, 86, 7-13.

Chen, P., Li, R., Fu, K., & Zhao, X. (2021, August). Research and Method for In-line Inspection Technology of Girth Weld in Long-Distance Oil and Gas Pipeline. In *Journal of Physics: Conference Series* Vol. 1986, No. 1, p. 012052. IOP Publishing.

Doğan, S., Akar, F., & Baran, A. (2022). “Geliştirilmiş Sobel Kenar Bulma Operatörünün Farklı Renk Uzaylarındaki Performansının Değerlendirilmesi.”, *Mühendislikte Güncel Araştırmalar*. Gece Kitaplığı, Ankara, ss.141-154

Dong, S., Sun, X., Xie, S., & Wang, M. (2019). Automatic defect identification technology of digital image of pipeline weld. *Natural Gas Industry B*, 6(4), 399-403.

Feng, Q., Li, R., Nie, B., Liu, S., Zhao, L., & Zhang, H. (2016). Literature review: Theory and application of in-line inspection technologies for oil and gas pipeline girth weld defection. *Sensors*, 17(1), 50.

Fioravanti, C. C. B., Centeno, T. M., & Da Silva, M. R. D. B. (2019). A deep artificial immune system to detect weld defects in DWDI radiographic images of petroleum pipes. *IEEE Access*, 7, 180947-180964.

Gürsel A. (2013). Petrol Borularının Kaynaklarında Radyografik Muayene Yönteminin Hata Tespit Kabiliyeti. *İleri Teknoloji Bilimleri Dergisi*. 2013; 2(1): 55-65

İgdaş (2011) “Sorularla Doğalgaz Çelik Boru Hattı Kaynağı ve Kontrolü” *İGDAŞ Yayınları*:26

Kafieh, R., Lotfi, T., & Amirfattahi, R. (2011). Automatic detection of defects on polyethylene pipe welding using thermal infrared imaging. *Infrared Physics & Technology*, 54(4), 317-325.

Luo, J., Tian, Z. and Yang, J., 2014. Fluorescent magnetic particle inspection device based on digital image processing. *Proceeding of the 11th World Congress on Intelligent Control and Automation*, IEEE, 5677-5681.

Madani, S., & Azizi, M. (2015). Detection of weld defects in radiography films using image processing. *Cumhuriyet Üniversitesi Fen Edebiyat Fakültesi Fen Bilimleri Dergisi*, 36(3), 2397-2404.

Shafeek, H. I., Gadelmawla, E. S., Abdel-Shafy, A. A., & Elewa, I. M. (2004). Automatic inspection of gas pipeline welding defects using an expert vision system. *NDT & E International*, 37(4), 301-307.

Shull, P.J., 2002. Nondestructive evaluation theory, Techniques and Applications. *CRC Press*

Sözeri, V., Keskin, O., Harmanşah, C., Kozak, İ., & Kibar, E. N. (2023). Görüntü İşleme Algoritmalarına Dayalı Olarak Manyetik Parçacık Test Görüntülerinde Kusur Tespitinin Araştırılması. *Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi*, 23(1), 115-122.”

Türkel, V. (2012). “Doğalgaz Dağıtımında Tasarım İmalat ve Yönetim” *İGDAŞ Yayınları*:27

BÖLÜM VI

Cisco Packet Tracer ile Ağ Tasarımı ve Simülasyonu

Muammer AKÇAY¹
Timorshah RAHİMİ²

Giriş

Ağ yöneticileri ve mühendisleri, sanal LAN teknolojisini veya VLAN'ı kullanarak fiziksel bir ağdan mantıksal bir ağ oluşturabilmektedir. Uyarlanabilir Güvenlik Cihazı (ASA) güvenlik duvarları, bir IP erişim kontrol listesi ve güvenlikle ilgili diğer araçlar kullanılmalıdır. Güvenlik açısından en önemli birim olan uçuş yönetim bölümü bu yöntemlerle korunmalıdır ve bu kişilerin veri çalması önlenmelidir. Ancak bu yöntemlerin yüksek düzeyde ağ güvenliği sunması ve bilgisayar korsanlarının sızmasını önlemesi gerekmektedir. Güvenlik yöntemleri geliştikçe bilgisayar korsanlarının yöntemleri de gelişecektir. Bilgisayar korsanlarının

¹ Dr. Öğretim Üyesi, Kütahya Dumlupınar Üniversitesi

² Yüksek lisans öğrencisi, Kütahya Dumlupınar Üniversitesi

ağa girip istediklerini elde etmelerini engellemek için de güvenlik önlemleri alınacaktır.

Bir kullanıcı diğer etkinliklerin yanı sıra telefonda konuşabilir, e-posta gönderip alabilir, sanal bir kurul veya sınıf toplantısına katılabilir, televizyon izleyebilir, radyo dinleyebilir, çevrimiçi bir şeye bakabilir, uygulama yazılımı, ödevler veya ders materyalleri gönderip alabilir. Tek bir günde. Başka bir milletten bir kişiyle çevrimiçi oyun oynayabilecektir. Günümüzde dünyanın neresinde olursa olsun, insanlar ve cihazlar ağlar aracılığıyla birbirine bağlanabilmektedir.

Üniversite için güvenli, genişletilebilir, kullanışlı ve yönetilebilir bir ağ tasarlamak bu araştırmanın amacıdır. Üniversitenin Mühendislik Fakültesi kurumsal ağının ölçeklenebilirliği, güvenliği ve yönetilebilirliği bu çalışmanın amacıdır. Bilgisayar ağları, Kişisel Alan Ağları (PAN'lar), Yerel Alan Ağları (LAN'lar), Metropolitan Alan Ağları (MAN'ler), Kampüs Alan Ağları (CAN'lar) ve Geniş Alan Ağları (WAN'lar) olmak üzere çeşitli şekil ve boyutlarda bulunmaktadır. LAN'larla kısıtlı bir coğrafi alan içinde daha yüksek veri aktarım hızları mümkündür. CAN, LAN ile karşılaştırıldığında daha geniş bir ağa sahiptir. Bir üniversite kampüsünde CAN genellikle çeşitli binaları, bilgisayar laboratuvarlarını, kütüphaneleri, araştırma laboratuvarlarını, kayıtları ve çeşitli akademik bölümleri birbirine bağlamak için uygulanmaktadır (Kurnaz et al, 2018).

Kampüs ağ hizmetini sağlamak için Nesnelerin İnterneti (IoT) cihazlarını kampüs ağındaki geleneksel ağ cihazlarıyla birleştiren bir Akıllı Kampüs Ağı Tasarımı önermektedir. Her akıllı cihazın farklı bir uygulama için Her Şeyin İnterneti sunucusuna kaydedilmesi ve yetkili kullanıcı tarafından kontrol edilmesi gerekmektedir. Akıllı Kampüs Ağı yapısı (SCND) tekniği, çeşitli kampüs ağ operasyonlarını kolaylaştırmak için bir IoT cihazını bir

ağ cihazıyla birleştirerek sunmaktadır. Bu cihazları birden çok ağ katmanında kümelemek için kullanılan hiyerarşik bir tasarım olduğundan Hiyerarşik Ağ Tasarımını içermektedir (Shemsi, 2017).

Bir proje üzerinde çalışan bir öğrenci, çeşitli laboratuvarlarda ve sınıflarda belirli bir proje dokümanına erişmek isteyecektir. Bölüm başkanı ise belirli sayıda oda ve sınıfa uğrayabilir. Üniversite binalarında öğretim üyeleri ile bölüm öğrencileri arasında bilgi paylaşımına yardımcı olacak bir ağ sistemi önerilmektedir (Sharma et al, 2019).

Bu çalışma, Packet Tracer'a genel bir bakış sağlanacak ve kullanıcılar daha fazla deneyim kazanacaktır (Javıd, 2014).

Ağ Yapılandırma

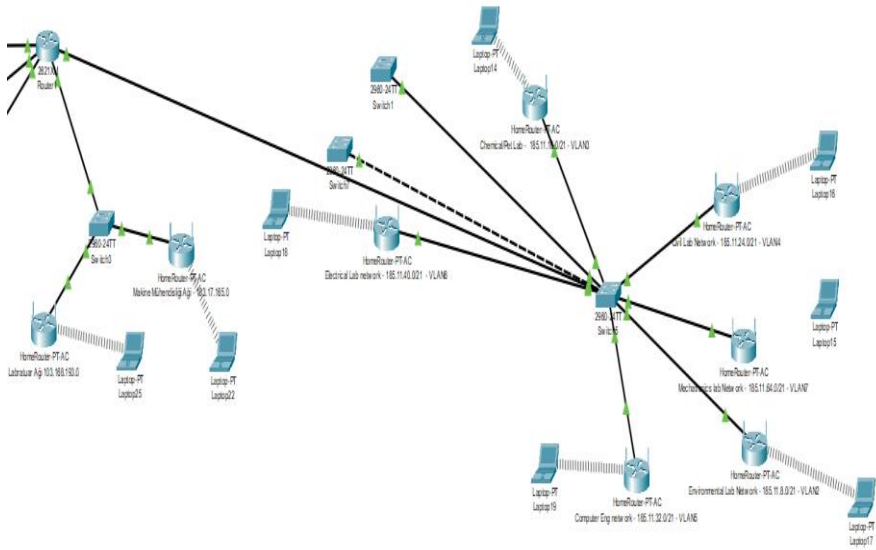
Cisco Adaptive Security Appliance (ASA), Cisco Systems tarafından geliştirilen bir dizi güvenlik cihazıdır. ASA ağ güvenlik duvarı, izinsiz giriş önleme sistemi özelliklerine tek bir platformda birleştirmektedir. Güçlü ve güvenliği ağ güvenliği sağlar ve ağ altyapısının çeşitli tehditlere karşı korunmasına yardımcı olmaktadır.

Güvenlik duvarı özellikleri: ASA, bir ağ güvenlik duvarı görevi görerek yöneticilerin gelen ve giden trafik için erişim kontrol politikalarını tanımlamasına imkân verir. Ağdaki Paketleri kaynak/hedef IP adresine, port numaralarına ve protokollere göre filtrelemektedir.

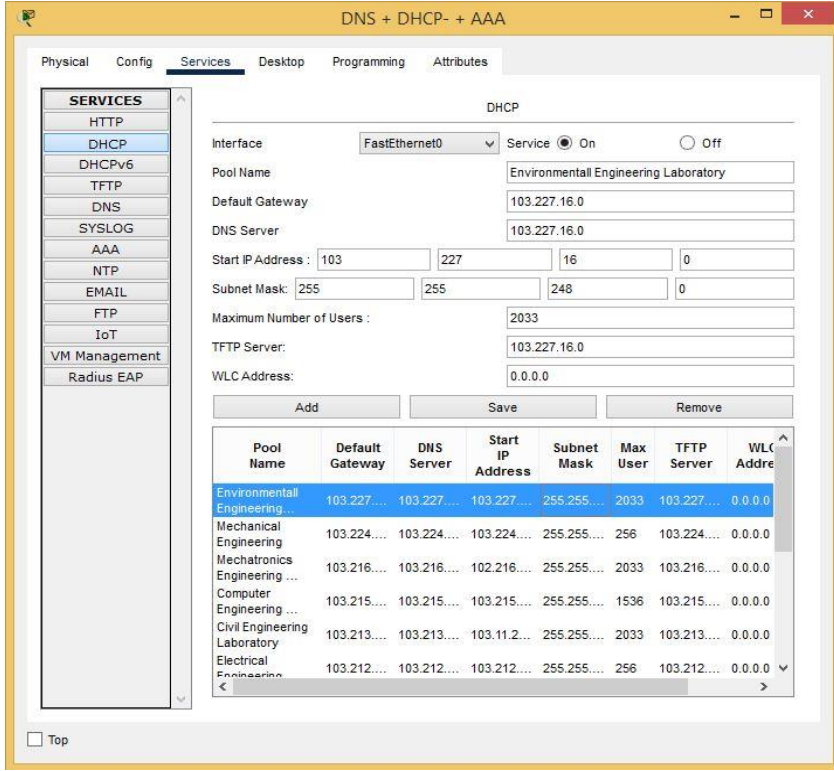
İzinsiz Giriş Önleme Sistemi (IPS): ASA, ağ trafiğini bilinen güvenlik açıkları ve kötü amaçlı etkinlikler açısından denetleyebilen bütünleşmiş bir IPS işlevselliği içermektedir. Kötü amaçlı yazılım, ağ saldırıları ve uygulama düzeyindeki saldırılar dâhil olmak üzere çeşitli tehdit türlerini algılayabilmektedir.

NAT (Ağ Adresi Çevirisi): ASA, özel IP adreslerinin giden trafik için genel IP adreslerine çevrilmesine olanak tanıyan NAT

işlevini desteklemektedir. Bu, IPv4 adres alanının korunmasına yardımcı olur ve dâhili ağ adreslerini gizleyerek bir düzeyde güvenlik sağlamaktadır. ASA ile Anahtar (Switch) arasındaki Cisco ASA, trunk bağlantısı ve alt ara yüzler aşağıdaki komutlar kullanılarak yapılandırılmıştır. İki ağ cihazı arasında bir trunk bağlantısı kurarken, yönlendirici ve anahtar komutu yalnızca bir kez kullanılmalı, güvenlik duvarı ve anahtar ise karşıt ara yüzlerin her birinde komutu iki kez kullanılmalıdır (Cisco, 2022).



Şekil 1. Örnek bir benzetim (Rahimi, 2023)



Şekil 2. Örnek Sunucu yapılandırma (Rahimi, 2023)

Sonuçlar

Bu çalışmada Cisco Packet Tracer kullanarak bir fakültenin ağının güvenliğinin sağlanması, erişim kontrolü, kimlik doğrulama, şifreleme ve ağ bölümlendirme gibi çeşitli güvenlik önlemlerinin uygulanması gösterilmiştir. Kaynak/hedef IP adreslerine, bağlantı noktalarına ve protokollere göre trafiği filtrelemek için yönlendiriciler ve anahtarlar üzerinde erişim kontrol listeleri uygulanmıştır. Ağ erişimini kontrol etmek için IEEE 802.1X veya RADIUS (Uzaktan Kimlik Doğrulama Çevirmeli Kullanıcı Hizmeti) gibi kimlik doğrulama mekanizmalarını uygulanmıştır. Farklı kullanıcı gruplarını veya bölümleri izole etmek için ayrı VLAN'lara (Sanal Yerel Alan Ağları) bölünmüştür. Bu, hassas

verilere yetkisiz erişimi önler ve olası güvenlik ihlallerinin etkisini sınırlar. Gelen ve giden trafiği izlemek ve kontrol etmek için ağ çevresine bir güvenlik duvarı yerleştirilmiştir.

Ancak gerçek dünyadaki ağ güvenliği uygulamaları ek konular ve teknolojiler içerebilecektir.

KAYNAKÇA

Kurnaz, S., Abdulkareem, M. M., ve Yaseen, A. S. (2018). Implementation Of Cisco Packet Tracer In Advance Computer Network, Faculty of Engineering Altinbas University, Aurum journal of Engineering system and architecture, vol.2 (issue. 1).

Shemsi, I. (2017, November). Boosting Campus Network Design Using Cisco Packet Tracer. International Journal of Innovative Science and Research Technology. Volume 2, Issue 11, ISSN No:-2456 –2165.

Sharma, M., Popreja, C., ve Arora, A. (2019, July). Design and Implementation of University Network. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-2S6.

Javıd, S. R. (2014). Role of Packet Tracer in learning Computer Networks International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 5,

<https://www.researchgate.net/publication/264233874>.

Cisco (2022). <https://www.cisco.com/c/en/us/products/security/adaptive-security-appliance-asa-software/index.html>, erişim tarihi: 11.02.2022.

Rahimi, T. (2023). Mühendislik Fakültesi için Güvenli Bir Ağ Tasarımı ve Simülasyonu. Yüksek Lisans Tezi, Kütahya Dumlupınar Üniversitesi, Lisans Üstü Eğitim Enstitüsü, Kütahya.

BÖLÜM VII

Kameraların Kısıtlı Alan Derinliğini Artırmak ve İyileştirilmiş Gece Görüşü Sağlamak için Görüntü Birleştirme Yöntemlerinin Karşılaştırılması

Fatma Çakıroğlu¹
Rifat Kurban^{2*}
Ali Durmuş³
Ercan Karaköse⁴

Giriş

Görüntü füzyonu iki ya da daha fazla kaynak görüntünün füzyon yöntemleri ile tek bir görüntüde birleştirilmesidir. Kaynak görüntüler farklı sensörlerden, farklı bakış açılarından, farklı

¹ Yüksek Mühendis, Kayseri Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, Kayseri

^{2*} Dr. Öğr. Üyesi, Abdullah Gül Üniversitesi, Bilgisayar Mühendisliği Bölümü, Kayseri, rifat.kurban@agu.edu.tr

³ Doç. Dr., Kayseri Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, Kayseri

⁴ Prof. Dr., Kayseri Üniversitesi, Mühendislik Temel Bilimleri Bölümü, Kayseri

odaklardan, çoklu pozlamadan yakalanan çeşitli görüntü çiftlerini içerir. Kaynak görüntülerdeki yararlı bilgileri kullanarak daha fazla bilgi içeren ve görüntü kalitesi artırılmış yeni bir görüntü elde edilir. Birleşik görüntü kaynak görüntülerin içermediği herhangi bir yapay olgu içermez. Birleştirilmiş görüntü, farklı görüntülerden yakalanan tamamlayıcı bilgileri sağlama kapasitesine sahiptir. Görüntü füzyonu aslında görüntü iyileştirmedir. Görüntü birleştirme farklı uygulama alanlarına hizmet etmektedir. Görüntü birleştirme bilgisayarlı görme, uzaktan algılama, mikroskobik ve tıbbi görüntüleme, görsel sensör ağları, optik mikroskopi, nesne tanıma, görüntü iyileştirme ve güvenlik alanında yaygın olarak kullanılmaktadır [1-8]. Genel olarak görüntü birleştirme yöntemleri üç ana kategoriye ayrılır. Bunlar uzamsal alanda çalışan yöntemler, dönüşüm alanı yöntemleri ve derin öğrenmeye dayalı yöntemlerdir [9, 10]. Uzamsal alan ve dönüşüm alanı yöntemlerinin avantajını kullanarak geliştirilen hibrit yöntemler de mevcuttur. Uzamsal alan yöntemleri kendi içinde piksel, blok ve bölge tabanlı yöntemler olmak üzere üç sınıfa ayrılır. Uzamsal alan ya da piksel tabanlı yöntemlerin görüntüleri, kaynak görüntülerin gri piksel düzeyi ya da renkli piksel değerlerini kullanarak birleştirir. Piksel tabanlı görüntü birleştirme yöntemleri için en iyi bilinen ve en basit yöntem aritmetik ortalama alma kabul edilebilir. Aritmetik ortalama alma yönteminde aynı noktada bulunan piksellerin aritmetik ortalaması alınarak birleşik görüntü oluşturulur [11, 12]. Bu yöntem hem çok odaklı hem de çok sensörlü görüntüleri birleştirmek için kullanılabilir. Bu yöntemlerin en büyük dezavantajı görüntü kontrastını azaltması ve gürültüye duyarlı olmasıdır. Farkların Gauss'u (GDFusion), Gauss filtresini kullanarak tüm kaynak görüntülerin her pikseli mesafeyle orantılı olacak şekilde komşularından ağırlıklandırılmasına dayanan basit ve etkili bir piksel tabanlı görüntü birleştirme Kurban tarafından literatüre sunulmuştur. Önerilen bu yöntem çok odaklı, çok modlu, çok sensörlü ve çok pozlamalı görüntülerde başarılı sonuçlar elde etmiştir [13]. Blok tabanlı yöntemlerde ise kaynak görüntüler $n \times m$ boyutunda bloklara ayrılır ve her blok için karşılaştırılma yapılarak

net blokların seçilmesi ile birleşik görüntü oluşturulur. Blok tabanlı yöntemde bloklar üst üste binmez yada aralarında boşluk kalmaz [14]. Blok tabanlı yöntemlerde önemli bir faktör olan blok boyutu $n \times n$ boyutunda bloklara ayrılacağı gibi optimize edilerek de bulunabilir. Blok boyutunun, yalnızca benzer netlik değerine sahip nesneler içerecek kadar küçük olması istenirken, netlik değerinin doğru olarak ölçülebilmesine imkan sağlayacak kadar da büyük olması beklenmektedir. Blok boyutu optimizasyonu için literatürde sık kullanılan algoritmalar şunlardır; Diferansiyel Gelişim Algoritması [11], Genetik Algoritma [15], Parçacık Sürü Optimizasyonu [16], Yapay Arı Kolonisi [17], Denizanası Arama Algoritması [18]. Blok boyutu optimizasyonuna dayalı farklı bir çalışma Kurban ve arkadaşları tarafından yapılmıştır. Bu çalışmada sürü tabanlı optimizasyon algoritmaları olan Parçacık Sürü Optimizasyonu, Denizanası Arama Algoritması ve Yapay Arı Kolonisi algoritmaları ile fizik tabanlı algoritmalar olan Arşimet Optimizasyon Algoritması, Atomik Yörünge Arama Algoritması, Denge Optimize Edici algoritmaları kullanılarak performans kıyaslaması yapılmıştır [19]. Bölge tabanlı görüntü birleştirmede Dürtü Bağımlı Sinir Ağı (PCNN) yöntemi, görüntüleri bölütlemeye kullanılmıştır. PCNN'ye dayalı görüntü birleştirme yöntemi Wang ve arkadaşları tarafından literatüre sunulmuştur. Bu yöntemde çift kanallı PCNN kullanılır. Kaynak görüntüler için odak ölçümleri yapılır. Odak ölçümü sonuçlarına göre ağırlıklı katsayılar otomatik olarak ayarlanır. Otomatik ayarlama kuralı belirli dönüşüme bağlıdır. Giriş görüntüleri çift kanallı PCNN'de birleştirilmiştir [20]. Çok odaklı, çok modlu ve çok sensörlü görüntü birleştirme yöntemlerinin bir diğeri ise dönüşüm alanı yöntemleridir. Dönüşüm tabanlı yöntemlerde genel mantık kaynak görüntülerin her birine öncelikle bir dönüşüm işlemi uygulanır. Ayrıştırma işleminden sonra birleşik dönüşüm katsayıları elde edilir ve ters dönüşüm işlemine tabi tutulur. Bu işlemler sonucunda birleşik görüntü elde edilir. Dönüşüm tabanında sıklıkla kullanılan yöntemler Temel Bileşenler Analizi (PCA) [21-23], Ayrık Kosinüs Dönüşümü (DCT) [24, 25], Ayrık Dalgacık Dönüşümü (DWT) [26-28], Laplace Piramidi (LP)

[29, 30], Durağan Dalgacık Dönüşümü (SWT) [23, 31]'dir. Dönüşüm alanı yöntemlerinin de kendi alanında avantaj veya dezavantajları mevcuttur. Uzamsal alanda ve dönüşüm alanında mevcut olan avantaj ve dezavantajları değerlendirip avantajları kullanarak yeni hibrit yöntemler geliştirilmiştir. Bu sayede daha kalite görüntüler elde edilmeye çalışılmıştır. Tyagi ve arkadaşları SWT ve PCA yöntemlerini kullanarak diğer dönüşüm tabanlı yöntemlerden daha iyi sonuçlar veren bir hibrit yöntem önermişlerdir [23]. Jindal ve ark. tarafından PCA, DWT, SWT yöntemlerini kullanan hibrit yöntem önerilmiştir [32]. Önerilen bu yöntem, görsel ve sayısal sonuçlar bakımından daha iyi sonuçlar elde etmiştir. Görüntü birleştirme de kullanılan ve son dönemde önemli çalışmalar yapılan bir diğer yöntem ise derin öğrenmeye dayalı yöntemlerdir. Derin öğrenmeye dayalı yöntemler de kendi içinde regresyon ve sınıflandırma tabanlı olarak iki gruba ayrılır. Görüntü birleştirme de en sık kullanılan model Evrişimli Sinir Ağı (CNN) modelidir. CNN modeli hem çok odaklı hem de çok sensörlü kıızılötesi ve görünür görüntü birleştirme için uygun bir modeldir [33, 34]. Çok odaklı görüntü füzyonu için kaynak görüntülerdeki odaklanmış ve odaklanmamış pikselleri komşuluk bilgilerinden tanıyabilen piksel bazında evrişimli bir sinir ağı (p-CNN) modeli [35], Darbe-Birleştirilmiş Sinir Ağı (PCNN) [20, 36], CNN tarafından çıkarılan yerel bilgileri entegre ederek odak bölgesi tanımlamasının doğruluğunu artıran Derin Evrişimli Sinir Ağı (DCNN) modeli [37] CNN' e bağlı geliştirilen yeni yöntemlerdir. Dönüşüm alanlı görüntü birleştirme algoritmalarından esinlenerek, birden fazla girdi görüntüsünden göze çarpan görüntü özelliklerini çıkarmak için iki evrişimli katman kullanarak görüntü birleştirmeye yarayan ve evrişimli sinir ağı modelini temel alan bir diğer yöntem IFCNN yöntemidir [38]. Tek bir modeli farklı görevler için sırayla eğitirken önceki füzyon yeteneklerinin kaybını önleyerek, çoklu füzyon görevlerine uygulanabilen birleşik model olan U2Fusion yöntemi de derin öğrenmeye dayalı yöntemdir ve çok modlu, çoklu pozlama ve çok odaklı görüntüler dahil olmak üzere farklı füzyon problemlerini çözme yeteneğine sahiptir [39]. Derin öğrenmeye

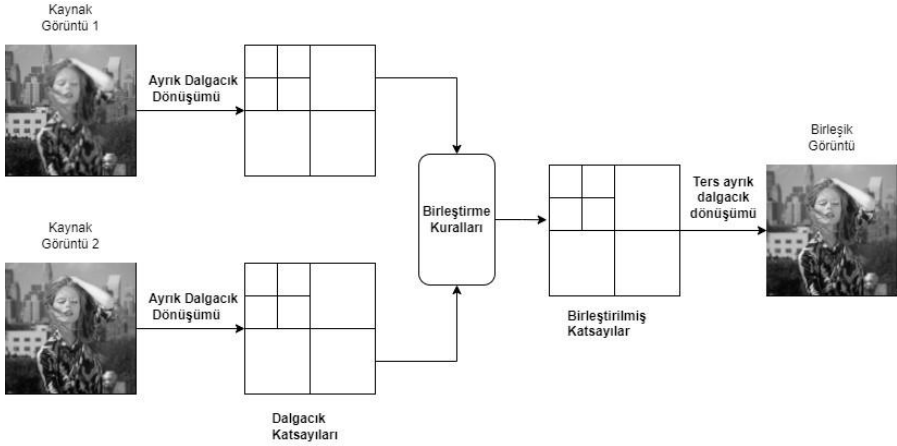
dayalı yöntemlerin kapsamlı bir incelemesi Wang ve arkadaşları tarafından literatüre sunulmuştur [40]. Bu bölümde literatürde kullanılan uzamsal tabanlı, dönüşüm tabanlı, hibrit ve derin öğrenmeye dayalı çok odaklı, çok sensörlü, çok modlu görüntü birleştirmeye uygun yöntemler hakkında genel bilgi verilmiştir.

2. Çok Odaklı ve Çok Sensörlü Görüntü Birleştirme Yöntemleri

Dijital görüntüleme sistemlerinde herhangi bir sahnenin görüntüsü alınırken kameranın sınırlı alan derinliği probleminden dolayı sahnenin tamamı net görüntülenemez [42]. Odak noktasının dışında kalan kısım bulanık görüntülenir. Bu problemin çözülebilmesi için aynı sahnenin farklı odaklarda alınmış iki ya da daha fazla görüntüsü birleştirilerek tamamı net olan yeni bir birleşik görüntü oluşturulur. Birleşik görüntü oluşturulurken temel amaç kaynak görüntülerdeki odak bilgisinin olabildiğince yüksek seviyede birleşik görüntüye aktarılmasıdır. Çok odaklı görüntülerin birleştirilmesi için bu çalışmada kullanılan farklı alanlarda çalışan yöntemler bu bölümde tanıtılmıştır.

2.1. Ayırık Dalgacık Dönüşümü (DWT)

Görüntü birleştirme çok yaygın olarak kullanılan dönüşüm tabanlı yöntemlerden biri Ayırık Dalgacık Dönüşümüdür [42]. DWT yönteminde kaynak görüntüye özyinelemeli olarak alçak ve yüksek geçişli filtre uygulanır. Devamında alt örnekleme işlemi yapılır. Bu açıdan DWT işlemi frekans kanalında sinyal ayrışması olarak düşünülebilir. Bu yöntemlere kaynak görüntülere dalgacık dönüşümü uygulanır. Kaynak görüntülerin her satırına alçak ve yüksek geçiren filtre uygulanır. Görüntü öz yinelemeli olarak frekans bantlarına ayrıştırılır. Dalgacık katsayıları oluşturulduktan sonra belirli kural dahilinde belirgin katsayılar seçilerek birleşik katsayılar oluşturulur. Bunun devamında birleşik katsayılarla ters dönüşüm işlemi uygulanarak birleşik görüntü elde edilmiş olur. DWT yöntemine ait blok diyagram Şekil 1.'de gösterilmiştir.



Şekil 1. *Ayrık Dalgacık Dönüşümü Tabanlı Çok Odaklı Görüntü Birleştirme Blok Diyagramı*

DWT yöntemi literatürde iyi bilinen başarılı bir yöntem olmasına karşın faz bilgisi eksikliği, zayıf yönlülük, alt örnekleme yapılmasına bağlı olarak oluşan kaynak görüntüler arasında değişimler ve kaymalar meydana gelmesi gibi dezavantajlara da sahiptir [43].

2.2. Denizanası Arama Algoritması (Jellyfish Search Algorithm-JSA)

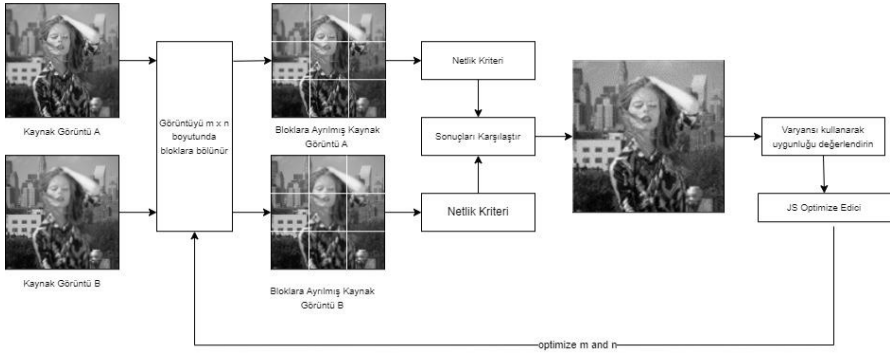
Denizanasının okyanustaki davranışından esinlenerek geliştirilmiş bir meta-sezgisel algoritmadır [44]. Problem çözümünde oldukça başarılıdır. Denizanasının okyanustaki arama davranışı, okyanus akıntısı, sürüsü içerisindeki aktif veya pasif hareketlerini, bir zaman kontrol mekanizmasını ve denizanası çiçeklerine yakınsamalarını içermektedir. Denizanası farklı boyut, şekil ve renklere sahiptir. Su içerisinde farklı sıcaklıklarda ve farklı derinliklerde yaşarlar. Bazı denizanasının boyu bir santimetreden daha küçük iken bazıları fazlasıyla büyüktür. Okyanus akıntısı onlara ne getirirse onu yerler çünkü beslenme şekilleri de farklılık göstermektedir. JSA temelde üç kurala dayanır.

1. Denizanası sürünün içinde hareket eder veya okyanus akıntısını takip eder ve bu hareket türleri arasındaki geçişi bir "zaman kontrol mekanizması" ile yönetir.

2. Denizanası okyanusta aktif ya da pasif hareketlerle yiyecek arar. Mevcut gıda miktarının daha fazla olduğu yerlere daha fazla ilgilidirler.

3. Bulunan yiyecek miktarı, ilgili amaç fonksiyonu ve konum belirlenir.

JSA ile çok modlu görüntü birleştirme yöntemine ait blok diyagramı Şekil 2.'de gösterilmiştir.



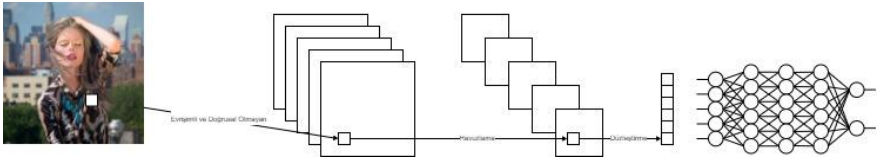
Şekil 2. Denizanası Arama Algoritması ile Çok Odaklı Görüntü Birleştirme Blok Diyagramı

JSA literatürde görüntü birleştirmede kullanılan başarılı bir yöntemdir. Çıtıl ve arkadaşları blok boyutu optimizasyonu için JSA yöntemini kullanarak başarılı sonuçlar elde etmiştir [18].

Bu yöntemde kaynak görüntüler öncelikle bloklar birbiri ile üst üste gelmeyecek veya aralarında boşluk kalmayacak şekilde bloklara ayrılır. Bu görüntüler netlik kriterine göre değerlendirilir ve sonuçlar kıyaslanır. Net olan bölge seçilerek birleşik görüntüye taşınır. JSA metodunun blok boyutunu optimize ederek birleşik görüntünün oluşmasını sağlar.

2.3. Evrişimli Sinir Ağı Modeli (CNN)

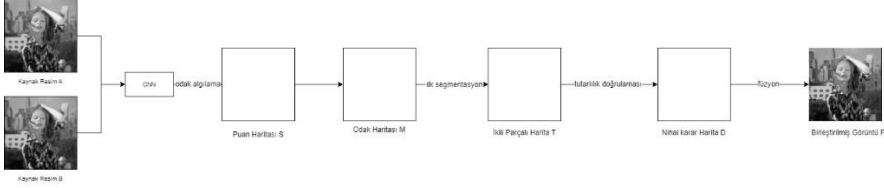
CNN, görüntü işlemede yaygın olarak kullanılan, eğitilebilir çok aşamalı ileri beslemeli bir yapay sinir ağıdır ve her aşama, özellikler için bir soyutlama düzeyine karşılık gelen belirli sayıda özellik haritası içerir [45]. CNN, bilgisayarlı görü alanında başarılı sonuçlarından dolayı en yaygın kullanılan derin öğrenme mimarisidir. CNN mimarisi AlexNet [46], ResNet [47], GoogleNet [48] gibi farklı modeller içerir. Klasik bir CNN modeli evrişim katmanı (CONV), indirgeme (pooling) ve tam bağlantı (fully connected) katmanlarına sahiptir [49]. Evrişim ya da konvolüsyon katmanı, derin öğrenme katmanlarının yapı taşıdır. Bu katman grid görüntüsünün özelliklerini algılar ve bir özellik çıkarımı sağlar. Girdi görüntüsündeki yüksek ve alçak seviyeli özellikleri çıkarmak için görüntüye bazı filtreler uygulanır. Görüntüye uygulanan filtreler sayesinde görüntü üzerindeki özellikler daha belirgin hale getirilerek görüntü bulanıklaştırma, görüntü keskinleştirme ve kenar tespiti yapılabilir. Filtreleme sonucu elde edilen çıktı matrisine özellik haritası (feature map) ya da aktivasyon haritası denir. Konvolüsyon katmanından sonra indirgeme (pooling) katmanı kendisine gelen matristeki öne çıkan özellikleri daha belirgin hale getirir. Tam bağlantı katmanında önceki bütün katmanlara bağlıdır. Tam bağlantı katmanının girdi matrisi kendinden önceki giriş matrisine düzleştirme işlemi uygulanarak elde edilir. CNN modelinin genel yapısı bu şekildedir. CNN mimarisinin genel yapısı Şekil 3.' de gösterilmiştir.



Şekil 3. CNN Mimarisinin Genel Yapısı

CNN modeli ile çok odaklı görüntü birleştirmede ilk adımda kaynak görüntülere ait puanlama haritası oluşturulur. İki kaynak görüntüde birbiriyle örtüşen blokların ortalaması alınarak odak

haritası elde edilir. Son adımda ise piksel bazında ağırlıklı ortalamayı kullanan nihai karar haritasıyla birleştirilmiş görüntü elde edilir. CNN modeli ile çok odaklı görüntü birleştirme blok diyagramı Şekil 4.'te gösterilmiştir [33].

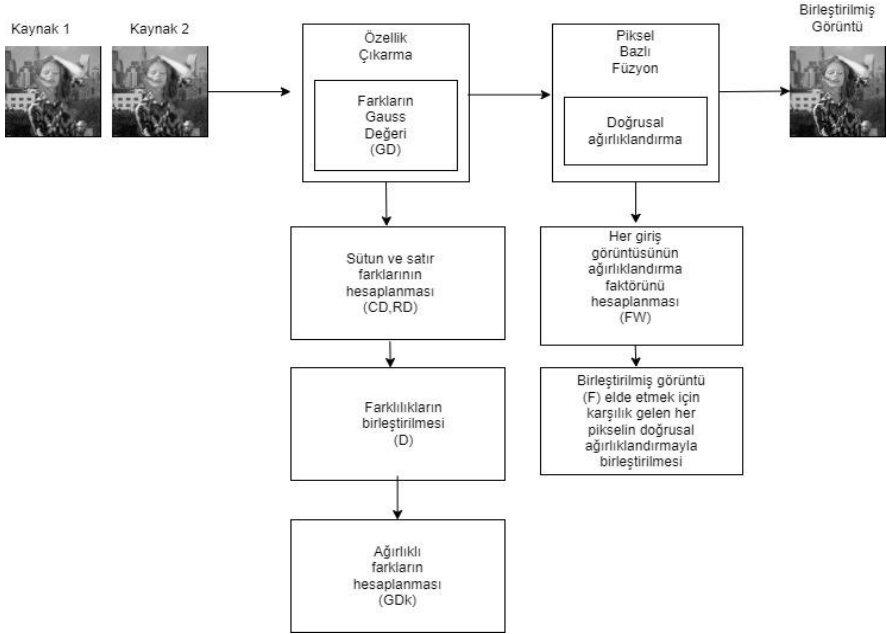


Şekil 4. CNN mimarisi ile Çok Odaklı Görüntü Birleştirme Blok Diyagramı

2.4. Farkların Gauss'u (GDFusion)

GDFusion yöntemi piksel tabanlı görüntü birleştirme yöntemidir [13]. Bu yöntemde herhangi bir dönüşüm işlemi uygulanmadığı için yöntem hızlı birleştirme yapar. GDFusion yönteminde öncelikle her kaynak görüntünün gradyanları birinci dereceden türev bilgilerine dayalı olarak hesaplanır. Bu gradyanlar daha sonra komşu piksellerle birlikte değerlendirilir. Her giriş görüntüsünün sütun ve satır farklılıkları hesaplanır. Sütun ve satır farklılıkları, sırasıyla dikey ve yatay eksenlerdeki kenarları vurgular. Bunları tek bir gösterimde birleştirmek için Öklid mesafesi kullanılarak kenar içeriğine göre her piksele ait özellikler hesaplanır. Her görüntüdeki piksellerin komşuları, farklı giriş görüntülerinin bilgi içeriğine ilgili pikselde, farklar bir 2D Gauss filtresi kullanılarak filtrelenir (yani ağırlıklandırılır). Ağırlık faktörleri, GD ile her giriş görüntüsündeki pikseller için belirlenir. Ağırlıklandırma faktörleri kullanılarak ağırlıklı ortalama elde edilir. GDPSQCV yönteminde ise görüntü arama (pattern search) optimizasyon yöntemi ile Gauss parametreleri Q_{CV} metriğini minimize edecek şekilde optimize edilmektedir.

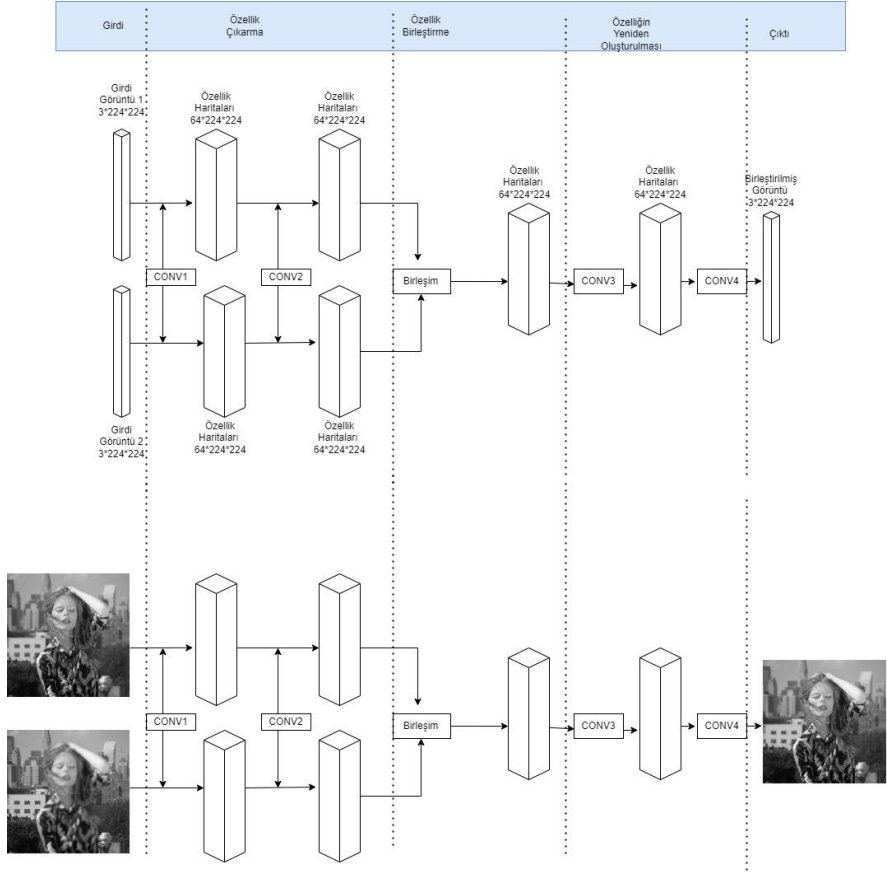
Şekil 5.'te GDFusion yönteminin blok diyagramı verilmiştir.



Şekil 5. GDFusion ile Görüntü Birleştirme Blok Diyagramı

2.5. IFCNN Yöntemi

IFCNN yöntemi CNN tabanlı bir yöntemdir. IFCNN üç modülden oluşur: özellik çıkarma modülü, özellik füzyon modülü ve görüntü yeniden yapılandırma modülü. İlk olarak, görüntü özelliklerini çıkarmak için iki evrişimli katman kullanır. İkinci modülde, birden fazla giriş görüntüsü, özellik birleştirme modülü aracılığıyla birleştirilir. Birleştirme işleminden sonra birleştirilmiş özellikler iki evrişimli katman tarafından yeniden yapılandırılır ve füzyon görüntüsü oluşturulur [38]. IFCNN modelinin blok diyagramı ile detaylı gösterimi Şekil 6.' da gösterilmiştir.

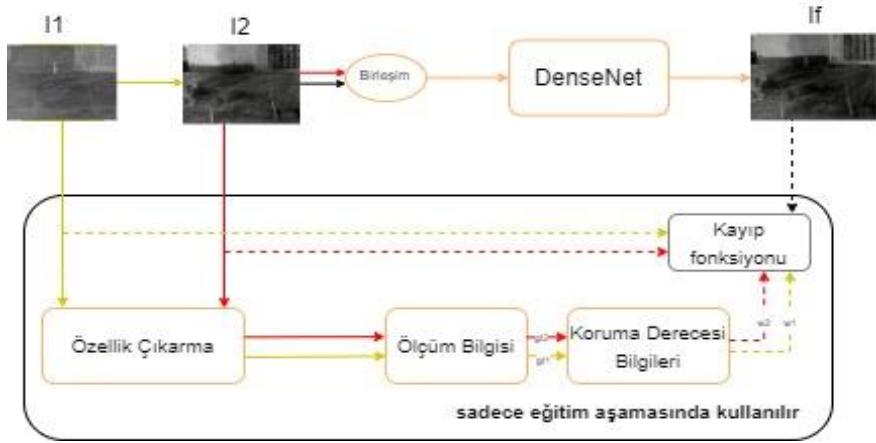


Şekil 6. IFCNN ile Görüntü Birleştirme Blok Diyagramı

2.6. Birleşik Denetimsiz Görüntü Füzyon Ağı Yöntemi (U2Fusion)

Birleşik Denetimsiz Görüntü Füzyon Ağı Yöntemi (U2Fusion), derin öğrenmeye dayalı görüntü birleştirme metodudur [50]. Şekil 7’de U2Fusion yöntemine ait blok diyagram verilmiştir. Bu blok diyagramı sadece eğitim aşamasını göstermektedir. Şekil7’de *I1* ve *I2* olarak gösterilen görüntüler kaynak görüntüyü temsil etmektedir. *I1* ve *I2* olarak gösterilen kaynak görüntülerle, bir If füzyon görüntüsünü oluşturmak için bir DenseNet eğitilir. Özellik

haritaları ile $I1$ ve $I2$ görüntülerinin özellikleri çıkartılır. Daha sonra özellik haritaları üzerinde gerçekleştirilen bilgi ölçümü ile $gI1$ ve $gI2$ ile gösterilen iki ölçüm üretilir. Daha sonra kayıp fonksiyonu için $I1$, $I2$, $w1$ ve $w2$ kullanılır. Füzyon arasındaki benzerliklerin ağırlıklarını tanımlayan resim ve kaynak resimler kaynak görüntülerdeki bilgileri korumak için, iki uyarlanabilir ağırlıklar bilgi koruma dereceleri olarak atanır. $w1$ ve $w2$ olarak ifade edilen bu uyarlanabilir ağırlıklar, bilgi ölçüm sonuçlarına göre tahmin edilmektedir. Eğitim aşamasında $w1$ ve $w2$, kayıp fonksiyonunun tanımlanmasında ölçülür ve uygulanır. Kaynak görüntü için, ağırlık ne kadar yüksek olursa bilginin koruma derecesi de o kadar yüksek olur. Daha sonra kayıp fonksiyonu en aza indirecek şekilde bir DenseNet modülü optimize edilir. DenseNet optimize edildiği için $w1$ ve $w2$ 'nin test aşamasında ölçülmesi gerekmektedir.



Şekil 7. U2Fusion ile Görüntü Birleştirme Blok Diyagramı

3. DeneySEL Sonuçlar

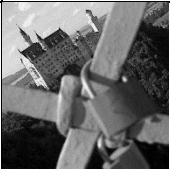
Çalışmada çok odaklı ve çok sensörlü kızılötesi ve görünür görüntü birleştirme yöntemleri karşılaştırılmıştır. Elde edilen birleşik görüntüler Q_{scd} , $Q_p^{AB/F}$, Q_{cb} , Q_{SF} ve Q_{brq} kalite metrikleri ile değerlendirilmiştir. Deneyler Intel i7 3.7 GHz işlemcili, 16 GB

RAM özelliklerine sahip bilgisayar ile MATLAB programının R2023b sürümünde gerçekleştirilmiştir.

3.1. Materyal

Çalışmada iki farklı görüntü grubu kullanılmıştır. Çok odaklı görüntü çiftleri [51] ve çok sensörlü kızılötesi ve görünür görüntü çiftleri [52] üzerinde birleştirme işlemi yapılmıştır. Tablo 1’de çok odaklı görüntü seti gösterilmektedir. Tablo 2’de ise kızılötesi (IR) ve görünür (VI) görüntü seti gösterilmektedir. Tablo1’de gösterilen çok odaklı görüntüler Lytro veri setinden alınmıştır [51].

Tablo 1. Çok Odaklı Görüntü Çifti

	#Resim1	#Resim2	#Resim3	#Resim4
Çok Odaklı Kaynak Görüntü 1				
Çok Odaklı Kaynak Görüntü 2				

Tablo 2. Çok Sensörlü Görüntü Çifti

	#Resim1	#Resim2	#Resim3	#Resim4
IR Görüntüler				
VI Görüntüler				

3.2. Görüntü Kalite Metrikleri

Çalışmada kullanılan yöntemler beş farklı kalite metriği ile değerlendirilmiştir. Bu metrikler Farkların Korelasyonlarının Toplamı (The sum of the correlations of differences- Q_{scd}), Kenar Tabanlı Kalite Metriği (Edge Based Quality Metric - $Q_p^{AB/F}$), Chen-Blum Kalite Metriği (Chen-Blum metric- Q_{cb}), Uzamsal Frekans (Spatial Frequency- Q_{SF}) ve Uzamsal Alan Referanssız Görüntü Kalite Metriği (Blind Referenceless Image Spatial Quality Evaluator, Q_{brq})’dir. Bu metriklerden Q_{brq} metriği için elde edilen değer küçük olması diğer dört metrik için elde edilen değer büyük olması birleştirmenin daha başarılı olduğunu ifade eder.

3.2.1. Farkların Korelasyonlarının Toplamı (The sum of the correlations of differences- Q_{scd})

Q_{scd} birleştirilmiş görüntü ve kaynak görüntüleri kullanılarak hesaplanan fark görüntüleri arasındaki korelasyona bağlı olan bir metriktir. Birleştirilmiş görüntü ve kaynak görüntüler arasındaki korelasyonları doğrudan kullanarak kaynaşmış görüntü kalitesini değerlendirmek yerine kaynak görüntüleri ve bunların birleştirilmiş görüntü üzerindeki etkisini göz önüne alarak kaliteyi ölçer [53].

S_1 ve S_2 kaynak görüntüler, F birleştirilmiş görüntü, D_1 ve D_2 fark görüntüleri olmak üzere;

$$Q_{scd} = r(D_1, S_1) + r(D_2, S_2) \quad (1)$$

Denklem (2), $r(\cdot)$ işlevi arasındaki korelasyonu hesaplar.

$$r(D_k, S_k) = \frac{\sum_i \sum_j (D_k(i,j) - \bar{D}_k)(S_k(i,j) - \bar{S}_k)}{\sqrt{\sum_i \sum_j (D_k(i,j) - \bar{D}_k)^2 \sum_i \sum_j (S_k(i,j) - \bar{S}_k)^2}} \quad (2)$$

3.2.2. Kenar Tabanlı Kalite Metriği (Edge Based Quality Metric - $Q_p^{AB/F}$)

$Q_p^{AB/F}$ metriği, kaynak görüntülerden birleşik görüntüyü oluşturmak için taşınan kenar bilgilerini kullanır. Bu bilgilerden yola çıkarak birleşik görüntünün kalitesini ölçer [54].

Kenar Tabanlı Kalite metriğinin matematiksel ifadesi Denklem (3)' de verilmiştir:

$$Q_p^{AB/F} = \frac{\sum_{n=1}^N \sum_{m=1}^M Q^{AF}(n,m) w^A(n,m) + Q^{BF}(n,m) w^B(n,m)}{\sum_{i=1}^N \sum_{j=1}^M (w^A(i,j) + w^B(i,j))} \quad (3)$$

Kenar koruma değeri $Q^{AF}(n,m)$ ve $Q^{BF}(n,m)$ sırasıyla $w^A(n,m)$ ve $w^B(n,m)$ ile ağırlıklandırılır. Başarılı bir sonuç için $Q_p^{AB/F}$ metriğinin elde ettiği değerin yüksek olması istenir.

3.2.3. Chen-Blum Kalite Metriği (Chen-Blum metric- Q_{cb})

Q_{cb} beş adımda oluşan ve insan algılarından ilham alarak geliştirilmiş bir referanssız görüntü kalite metriğidir [55]. Bu beş adım yerel kontrast hesaplama, kontrast duyarlılığı filtreleme, belirginlik haritası oluşturma, kontrast korunumu hesaplama, global kalite haritasıdır. Q_{cb} kalite metriği, global kalite haritasının ortalama değerleri hesaplanarak oluşturulur.

3.2.4. Uzamsal Frekans (Spatial Frequency- Q_{SF})

Uzamsal Frekans, birinci dereceden türev tabanlı satır ve sütun hesaplanarak elde edilen bir metriktir. RF ve CF, sırasıyla birleşik görüntünün satır ve sütun frekanslarıdır [56]. Birleştirilmiş görüntünün uzamsal frekansı şu şekilde hesaplanır:

$$SF = \sqrt{RF^2 + CF^2} \quad (4)$$

$$RF = \left[\frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (F(i,j) - F(i,j-1))^2 \right]^{1/2} \quad (5)$$

$$CF = \left[\frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (F(i,j) - F(i-1,j))^2 \right]^{1/2} \quad (6)$$

3.2.5. Uzamsal Alan Referanssız Görüntü Kalite Metriği (Blind Referenceless Image Spatial Quality Evaluator, Q_{brq})

Uzamsal alan referanssız görüntü kalite metriği (Blind Referenceless Image Spatial Quality Evaluator) referanssız bir görüntü kalite metriğidir, uzamsal alanda çalışır. Bu metrik üç aşamada çalışmasını yürütür. İlk olarak görüntünün doğal sahne istatistiği çıkartılır, daha sonra özellik vektörleri hesaplanır ve son adımda görüntünün kalite değeri hesaplanır. Birleştirme performansının başarılı kabul edilebilmesi için Q_{brq} metriğinin sayısal değeri küçük olması istenir. [57].

3.3. Yöntemlerden Elde Edilen Sayısal ve Görsel Sonuçlar

Çok odaklı görüntü çiftleri için elde edilen sayısal ve görsel sonuçlar Bölüm 3.3.1 de, IR ve VI görüntü çiftlerine ait sayısal ve görsel sonuçlar Bölüm 3.3.2’de tablo halinde gösterilmiştir. Başarılı bulunan sonuçlar kırmızı renk ile belirtilmiştir.

3.3.1.Çok Odaklı Görüntü Birleştirme Sayısal Sonuçları

Tablo 3. #Resim1’in Sayısal sonuçları

#Resim1	Q_{scd}	$Q_p^{AB/F}$	Q_{cb}	Q_{SF}	Q_{brq}	CPU İşlem süresi
DWT	0.5049	0.6749	0.7612	24.3973	24.7403	1.5792
CNN	0.3985	0.7209	0.8209	24.7216	10.9018	9.7759
JSA	0.3364	0.7221	0.8341	24.6811	15.9699	10.9636
GDPSQCV	0.3046	0.6741	0.7066	15.6966	24.7955	6.3882

Tablo 3’de çok odaklı görüntü çifti #Resim1’e ait sayısal değerlendirme sonuçları verilmiştir. JSA ve CNN yöntemlerin daha başarılı olduğu Tablo 3’ten görülmektedir. CPU işlem süresi açısından DWT yöntemi en hızlı sonuç vermiştir.

Tablo 4. #Resim2'nin Sayısal Sonuçları

#Resim2	Q_{scd}	$Q_p^{AB/F}$	Q_{cb}	Q_{SF}	Q_{brq}	CPU İşlem süresi
DWT	0.4874	0.6981	0.7929	22.5258	20.3366	0.1736
CNN	0.4470	0.7384	0.8382	22.6967	20.9380	9.4124
JSA	0.4487	0.7399	0.8527	22.7287	27.8933	8.8602
GDPSQCV	0.1382	0.6355	0.7042	12.3093	33.2908	12.1037

Tablo 4 çok odaklı görüntü çifti #Resim2'ye ait sayısal sonuçları göstermektedir. Sonuçlar değerlendirildiğinde JSA yöntemi $Q_p^{AB/F}$, Q_{cb} ve Q_{SF} kalite metrikleri için en başarılı yöntem olmuştur. Q_{brq} ve Q_{scd} metrikleri için ve işlem süresi açısından ise DWT yöntemi en iyi birleştirme performansına sahiptir.

Tablo 5. çok odaklı görüntü çifti #Resim3' e ait sayısal değerlendirme sonucunu göstermektedir. #Resim3 için JSA yöntemi Q_{cb} ve Q_{SF} kalite metrikleri için, GDPSQCV yöntemi Q_{brq} kalite metriği için ve DWT yöntemi Q_{scd} kalite metriği açısından en başarılı olmuştur. Ayrıca DWT yöntemi işlem süresi açısından en başarılıdır.

Tablo 5. #Resim3'ün Sayısal Sonuçları

#Resim 3	Q_{scd}	$Q_p^{AB/F}$	Q_{cb}	Q_{SF}	Q_{brq}	CPU İşlem süresi
DWT	0.7339	0.8544	0.8587	33.2685	29.4890	0.2093
CNN	0.4018	0.8775	0.8845	33.3629	26.1114	9.6416
JSA	0.3587	0.8761	0.8902	33.5924	25.8838	14.5884
GDPSQCV	0.4422	0.7466	0.8105	22.4696	21.6534	6.6655

Tablo 6 çok odaklı görüntü çifti #Resim4'e ait sonuçları göstermektedir. Sonuçlar değerlendirildiğinde DWT yöntemi Q_{scd} ve Q_{brq} kalite metrikleri açısından en iyi birleştirme performansına sahip iken, JSA yöntemi $Q_P^{AB/F}$ ve Q_{cb} kalite metrikleri açısından en başarılı yöntem olmuştur. İşlem süresi açısından yine DWT yöntemini en hızlı işlem süresine sahiptir.

Tablo 6. #Resim4'ün Sayısal Sonuçları

#Resim 4	Q_{scd}	$Q_P^{AB/F}$	Q_{cb}	Q_{SF}	Q_{brq}	CPU İşlem süresi
DWT	0.6308	0.7010	0.7469	30.8415	5.7428	0.1072
CNN	0.5590	0.7369	0.8007	31.3901	9.7261	9.3686
JJSA	0.5372	0.7429	0.8330	31.3895	11.7529	9.5895
GDPSQCV	0.2968	0.6700	0.7227	19.9933	17.5206	8.7829

3.3.2.Çok Odaklı Görüntü Birleştirme Görsel Sonuçları

Tablo 7. #Resim1, #Resim2, #Resim3, #Resim4 Görüntülerine Ait Görsel Sonuçlar

	CNN	DWT	JSA	GDPSQCV
#Resim1				
#Resim2				
#Resim3				
#Resim4				

Görüntü birleştirmede yöntemlerin birleştirme performansları kıyaslanırken sadece sayısal sonuçlarla değerlendirme yapmak yetersizdir. Görsel sonuçlarla sayısal sonuçların birbirini destekler nitelikte olması istenir. Tablo 7 #Resim1, #Resim2, #Resim3, #Resim4 görüntülerine ait görsel sonuçları göstermektedir. Bu tablodan JSA ve DWT yöntemlerinin elde ettiği sonuçların daha net olduğu anlaşılmaktadır.

3.3.3. Çok-sensörlü IR ve VI Görüntü Çiftlerine Ait Sayısal Sonuçlar

Tablo 8. #Resim1 IR-VI Görüntülerin Sayısal Sonuçları

#Resim1 IR-VI	Q_{scd}	$Q_P^{AB/F}$	Q_{cb}	Q_{SF}	Q_{brq}	CPU İşlem süresi
IFCNN	0.9333	0.6095	0.4915	10.3244	23.0257	-
DWT	0,9541	0,5384	0,4240	11,1630	12,5719	0,6634
U2Fused	1.2407	0.5357	0.5247	9.2912	28.0740	-
GDPSQCV	0.7948	0.6417	0.5129	9.3746	30.8978	29.9775

Tablo 8’de IR ve VI görüntü çifti #Resim1 için elde edilen sonuçlar verilmiştir. Tablo incelendiğinde U2Fused yöntemi Q_{scd} ve Q_{cb} metrikleri için en başarılı yöntem olmuştur. DWT yöntemi Q_{SF} ve Q_{brq} açısından en başarılı olmuştur. GDPSQCV yöntemi ise Q_{cb} açısından başarılı olmuştur.

Tablo 9. #Resim2 IR_VI Görüntülerin Sayısal Sonuçları

#Resim2 IR-VI (mancall)	Q_{scd}	$Q_P^{AB/F}$	Q_{cb}	Q_{SF}	Q_{brq}	CPU İşlem süresi
IFCNN	1.4125	0.4749	0.4941	9.1559	33.5892	-
DWT	1.3185	0.4310	0.5191	13.3577	26.4635	0.0817
U2Fused	1.5419	0.4283	0.5252	10.3116	40.8973	-
GDPSQCV	1.1282	0.4732	0.4302	6.7978	28.5947	23.2882

Tablo 9’da IR ve VI görüntü çifti #Resim2 için elde edilen sonuçlar verilmiştir. U2Fused yöntemi Q_{scd} ve Q_{cb} kalite metrikleri için en başarılı yöntem olmuştur. DWT yöntemi Q_{SF} ve Q_{brq} açısından en başarılı olmuştur. IFCNN yöntemi ise $Q_P^{AB/F}$ ve Q_{cb} açısından başarılı olmuştur.

Tablo 10. #Resim3 IR_VI Görüntülerin Sayısal Sonuçları

#Resim3 IR-VI (manwithbag)	Q_{scd}	$Q_p^{AB/F}$	Q_{cb}	Q_{SF}	Q_{brq}	CPU İşlem süresi
IFCNN	1.4192	0.7439	0.5322	24.2393	12.9255	-
DWT	1.3108	0.6363	0.5248	20.0856	21.1022	0.0798
U2FUsed	1.4854	0.6115	0.6163	16.0731	21.2891	-
GDPSQCV	1.1778	0.7542	0.6588	16.8286	25.2455	21.6963

Tablo 10’da IR ve VI görüntü çifti #Resim3 için elde edilen sonuçlar verilmiştir. U2Fused yöntemi Q_{scd} kalite metriği için en başarılı yöntem olmuştur. IFCNN yöntemi Q_{SF} ve Q_{brq} açısından en başarılı olmuştur. GDPSQCV yöntemi ise $Q_p^{AB/F}$ ve Q_{cb} metrikleri açısından başarılı olmuştur.

Tablo11. Çok Sensörlü Görüntülerin Sayısal Sonuçları

#Resim4 IR-VI (manlight)	QSCD	Qp_ABf	Qcb	QSF	Qbrq	CPU İşlem süresi
IFCNN	1.4192	0.7439	0.5322	24.2393	12.9363	-
DWT	1.4815	0.7118	0.4771	27.7514	17.1533	0.0531
U2FUsed	1.5199	0.6660	0.5773	21.3155	17.2787	-
GDPSQCV	1.2395	0.7898	0.6010	23.0581	26.5014	33.1905

Tablo 11’de, IR ve VI görüntü çifti #Resim4 için elde edilen sonuçlar verilmiştir. U2Fused yöntemi Q_{scd} kalite metriği için en başarılı yöntem olmuştur. IFCNN yöntemi Q_{brq} açısından en başarılı olmuştur. GDPSQCV yöntemi ise $Q_p^{AB/F}$ ve Q_{cb} metrikleri açısından başarılı olmuştur. DWT yöntemi ise Q_{SF} kalite metriği açısından en başarılı yöntem olmuştur.

3.3.4. Çok Sensörlü IR ve VI Görüntü Çiftlerine Ait Görsel Sonuçlar

Tablo 12. Çok Sensörlü Görüntülerin Sayısal Sonuçları

	IFCNN	DWT	U2Fused	GDPSQCV
#Resim1 IR-VI				
#Resim2 IR-VI				
#Resim3 IR-VI				
#Resim4 IR-VI				

IR ve VI görüntü çiftlerinin görsel sonuçları değerlendirildiğinden U2Fusion yönteminin elde ettiği görsel sonuçların daha net olduğu anlaşılmaktadır. Bu durum sayısal sonuçlarda U2Fusion yönteminin elde ettiği başarıyı desteklemektedir.

Tablo13. *Çalışmada Kullanılan Birleştirme Yöntemleri ve Özellikleri*

Yöntem	Sınıfı	Ağ Tipi
DWT	Dönüşüm Alanı	-
CNN	Derin Öğrenme	CNN
Jellfish	Blok Tabanlı	-
GDPSQCV	Piksel Tabanlı	-
IFCNN	Derin Öğrenme	CNN
U2Fusion	Derin Öğrenme	DenseNet

4.SONUÇ

Görüntü füzyonu birçok alanda ihtiyaç duyulan bir işlemdir. Görüntü birleştirme için literatürde farklı birleştirme yöntemleri mevcuttur. Bu çalışmada kameraların sınırlı alan derinliğini artırmak ve iyileştirilmiş gece görüşü sağlamak için, Ayrık Dalgacık Dönüşümü (DWT), Deniz Anası Arama Algoritması (JSA), Evrişimsel Sinir Ağı (CNN) ve Farkların Gauss (GDPSQCV), Evrişimsel Sinir Ağını temel alan IFCNN yöntemi ve Birleşik Denetimsiz Görüntü Füzyon Ağı (U2Fusion) yöntemleri arasında birleştirme performansı kıyaslaması yapılmıştır. Çok odaklı görüntü çiftlerinin birleştirilmesinde DWT ve JSA yöntemleri farklı kalite metrikleri için başarılı olmuştur. Kızılötesi ve görünür görüntü çiftlerinin birleştirilmesinde ise U2Fusion, GDPSQCV, IFCNN ve DWT yöntemlerinin hepsi farklı görüntülerde farklı kalite metrikleri açısından birbirine yakın başarı elde etmiştir.

KAYNAKÇA

- [1] Ghassemian, H. (2016) A review of remote sensing image fusion methods. *Information Fusion*, 32, 75-89.
- [2] Li, Q., Yang, X., Wu, W., Liu, K., & Jeon, G. (2018). Multi-focus image fusion method for vision sensor systems via dictionary learning with guided filter. *Sensors*, 18(7), 2143.
- [3] Meher, B., Agrawal, S., Panda, R., & Abraham, A. (2019). A survey on region based image fusion methods. *Information Fusion*, 48, 119-132.
- [4] Song, Y., Li, M., Li, Q., & Sun, L. (2006, December). A new wavelet based multi-focus image fusion scheme and its application on optical microscopy. In *2006 IEEE International Conference on Robotics and Biomimetics* (pp. 401-405). IEEE.
- [5] Wang, Z., & Ma, Y. (2008). Medical image fusion using m-PCNN. *Information Fusion*, 9(2), 176-185.
- [6] Singh, R., Vatsa, M., & Noore, A. (2008). Integrated multilevel image fusion and match score fusion of visible and infrared face images for robust face recognition. *Pattern Recognition*, 41(3), 880-893.
- [7] Goshtasby, A. A., & Nikolov, S. G. (2007). Guest editorial: Image fusion: Advances in the state of the art. *Information Fusion: Special Issue on Image Fusion: Advances in the State of the Art*, 8, 114-118.
- [8] Zhang, Y., Zhang, L., Bai, X., & Zhang, L. (2017). Infrared and visual image fusion through infrared feature extraction and visual information preservation. *Infrared Physics & Technology*, 83, 227-237.
- [9] Bhat, S., & Koundal, D. (2021). Multi-focus image fusion techniques: a survey. *Artificial Intelligence Review*, 54, 5735-5787.

[10] Maruthi, R., & Lakshmi, I. (2017). Multi-focus image fusion methods—a survey. *Comput Eng*, 19(4), 9-25.

[11] Aslantas, V., & Kurban, R. (2010). Fusion of multi-focus images using differential evolution algorithm. *Expert Systems with Applications*, 37(12), 8861-8870.

[12] Tian, J., Chen, L., Ma, L., & Yu, W. (2011). Multi-focus image fusion using a bilateral gradient-based sharpness criterion. *Optics communications*, 284(1), 80-87.

[13] Kurban, R. (2023). Gaussian of Differences: A Simple and Efficient General Image Fusion Method. *Entropy*, 25(8), 1215.

[14] Li, S., Kwok, J. T., & Wang, Y. (2001). Combination of images with diverse focuses using the spatial frequency. *Information fusion*, 2(3), 169-176.

[15] Zhang, X., Han, J., & Liu, P. (2005). Restoration and fusion optimization scheme of multifocus image using genetic search strategies. *Optica Applicata*, 35(4).

[16] Aslantas, V., & Kurban, R. (2010, June). Extending depth-of-field of a digital camera using particle swarm optimization based image fusion. In *IEEE International Symposium on Consumer Electronics (ISCE 2010)* (pp. 1-5). IEEE.

[17] Aslantaş, V., & Kurban, R. (2010, July). Multi-focus Image Fusion in Spatial Domain using Artificial Bee Colony Optimization Algorithm. *International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2010)* pp.1-5.

[18] Çıtıl, F., Kurban, R., Durmuş, A., & Karaköse, E. (2022). Fusion of Multi-Focus Images using Jellyfish Search Optimizer. *Avrupa Bilim ve Teknoloji Dergisi*, (37), 147-155.

[19] F. Çakıroğlu, R. Kurban, A. Durmuş, and E. Karaköse, "Multi-focus image fusion by using swarm and physics based metaheuristic algorithms: a comparative study with archimedes, atomic orbital search, equilibrium, particle swarm,

artificial bee colony and jellyfish search optimizers," *Multimedia Tools Applications* vol. 82, no. 29, pp. 44859-44883, 2023.

[20] Wang, Z., Ma, Y., & Gu, J. (2010). Multi-focus image fusion using PCNN. *Pattern Recognition*, 43(6), 2003-2016.

[21] Patil, U., & Mudengudi, U. (2011, November). Image fusion using hierarchical PCA. In *2011 international conference on image information processing* (pp. 1-6). IEEE.

[22] Yang, Y., Ding, M., Huang, S., Que, Y., Wan, W., Yang, M., & Sun, J. (2017). Multi-focus image fusion via clustering PCA based joint dictionary learning. *IEEE Access*, 5, 16985-16997.

[23] Tyagi, T., Gupta, P., & Singh, P. (2020, January). A hybrid multi-focus image fusion technique using SWT and PCA. In *2020 10th international conference on cloud computing, data science & engineering (confluence)* (pp. 491-497). IEEE.

[24] Phamila, Y. A. V., & Amutha, R. (2014). Discrete Cosine Transform based fusion of multi-focus images for visual sensor networks. *Signal Processing*, 95, 161-170.

[25] Liu, Y., Wang, L., Cheng, J., Li, C., & Chen, X. (2020). Multi-focus image fusion: A survey of the state of the art. *Information Fusion*, 64, 71-91.

[26] Yang, Y., Huang, S., Gao, J., & Qian, Z. (2014). Multi-focus image fusion using an effective discrete wavelet transform based algorithm. *Measurement science review*, 14(2), 102-108.

[27] Kannan, K., & Perumal, S. A. (2007, December). Optimal decomposition level of discrete wavelet transform for pixel based fusion of multi-focused images. In *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)* (Vol. 3, pp. 314-318). IEEE.

[28] Singh, S., Singh, H., Gehlot, A., Kaur, J., & Gagandeep. (2023). IR and visible image fusion using DWT and bilateral filter. *Microsystem Technologies*, 29(4), 457-467.

- [29] Mingrui, C., Junyi, Y., & Guanghui, C. (2015, September). Multi-focus image fusion algorithm using LP transformation and PCNN. In *2015 6th IEEE International conference on software engineering and service science (ICSESS)* (pp. 237-241). IEEE.
- [30] Wang, W., & Chang, F. (2011). A Multi-focus Image Fusion Method Based on Laplacian Pyramid. *J. Comput.*, 6(12), 2559-2566.
- [31] Aymaz, S., & Köse, C. (2017, November). Multi-focus image fusion using stationary wavelet transform (SWT) with principal component analysis (PCA). In *2017 10th International Conference on Electrical and Electronics Engineering (ELECO)* (pp. 1176-1180). IEEE.
- [32] Jindal, M., Bajal, E., Chakraborty, A., Singh, P., Diwakar, M., & Kumar, N. (2021). A novel multi-focus image fusion paradigm: A hybrid approach. *Materials Today: Proceedings*, 37, 2952-2958.
- [33] Liu, Y., Chen, X., Peng, H., & Wang, Z. (2017). Multi-focus image fusion with a deep convolutional neural network. *Information Fusion*, 36, 191-207.
- [34] Liu, Y., Chen, X., Cheng, J., Peng, H., & Wang, Z. (2018). Infrared and visible image fusion with convolutional neural networks. *International Journal of Wavelets, Multiresolution and Information Processing*, 16(03), 1850018.
- [35] Tang, H., Xiao, B., Li, W., & Wang, G. (2018). Pixel convolutional neural network for multi-focus image fusion. *Information Sciences*, 433, 125-141.
- [36] Li, M., Cai, W., & Tan, Z. (2006). A region-based multi-sensor image fusion scheme using pulse-coupled neural network. *Pattern Recognition Letters*, 27(16), 1948-1956.

[37] Lai, R., Li, Y., Guan, J., & Xiong, A. (2019). Multi-scale visual attention deep convolutional neural network for multi-focus image fusion. *IEEE Access*, 7, 114385-114399.

[38] Zhang, Y., Liu, Y., Sun, P., Yan, H., Zhao, X., & Zhang, L. (2020). IFCNN: A general image fusion framework based on convolutional neural network. *Information Fusion*, 54, 99-118.

[39] Xu, H., Ma, J., Jiang, J., Guo, X., & Ling, H. (2020). U2Fusion: A unified unsupervised image fusion network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1), 502-518.

[40] Wang, Z., Ma, Y., & Zhang, Y. (2023). Review of pixel-level remote sensing image fusion based on deep learning. *Information Fusion*, 90, 36-58.

[41] R. Kurban (2012), Resim uzayında blok seçmeye dayalı yeni görüntü birleştirme yöntemleri. Erciyes Üniversitesi Fen Bilimleri Enstitüsü.

[42] Pajares, G., & De La Cruz, J. M. (2004). A wavelet-based image fusion tutorial. *Pattern recognition*, 37(9), 1855-1872.

[43] Fernandes, F. C., Selesnick, I. W., Van Spaendonck, R. L., & Burrus, C. S. (2003). Complex wavelet transforms with allpass filters. *Signal Processing*, 83(8), 1689-1706.

[44] Chou, J. S., & Truong, D. N. (2021). A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Applied Mathematics and Computation*, 389, 125535.

[45] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, 8, 1-74.

[46] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

[47] HE, Kaiming, et al. Deep residual learning for image recognition. En *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. p. 770-778.

[48] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[49] Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7), 3523-3542.

[50] H. Xu, J. Ma, J. Jiang, X. Guo, and H. Ling, "U2Fusion: A unified unsupervised image fusion network," *IEEE Transactions on Pattern Analysis Machine Intelligence* vol. 44, no. 1, pp. 502-518, 2020.

[51] M. Nejati, S. Samavi, and S. Shirani, "Multi-focus image fusion using dictionary-based sparse representation," *Information Fusion* vol. 25, pp. 72-84, 2015.

[52] Z. Zhou, B. Wang, S. Li, and M. Dong, "Perceptual fusion of infrared and visible images through a hybrid multi-scale decomposition with Gaussian and bilateral filters," *Information Fusion* vol. 30, pp. 15-26, 2016.

[53] V. Aslantas and E. Bendes, "A new image quality metric for image fusion: The sum of the correlations of differences," *Aeu-international Journal of electronics communications* vol. 69, no. 12, pp. 1890-1896, 2015.

[54] C. S. Xydeas and V. Petrovic, "Objective image fusion performance measure," *Electronics letters* vol. 36, no. 4, pp. 308-309, 2000.

[55] Y. Chen and R. S. Blum, "A new automated quality assessment algorithm for image fusion," *Image vision computing* vol. 27, no. 10, pp. 1421-1432, 2009.

[56] A. M. Eskicioglu and P. S. Fisher, "Image quality measures and their performance," *IEEE Transactions on communications* vol. 43, no. 12, pp. 2959-2965, 1995.

[57] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on image processing* vol. 21, no. 12, pp. 4695-4708, 2012.

