

Map Generation & Manipulation with Generative Adversarial Networks

Ozan Balci¹, Şemsi Barış Terzi², Özgün Balaban³

0000-0002-9265-2262¹, 0000-0002-3400-7839², 0000-0002-7270-2058³

¹Istanbul Technical University, Faculty of Architecture, Department of Architecture, Istanbul, Turkey

²Istanbul Technical University, Graduate School of Science, Engineering, and Technology, Department of Informatics, Architectural Design Computing, Istanbul, Turkey

³Chair of Design Informatics, Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, Hollanda

Artificial intelligence makes the Big Data processable as the spread of the Internet. Machine learning, which is a type of artificial intelligence, has made it easier to classify, interpret and process data. On the other hand, the development of graphics processing units has facilitated and accelerated the work done by providing the necessary infrastructure for processing data consisting of images. The development of image processing operation will enable the processing and production of architectural data sets consisting of images. This study aims to show the processability of the image set in map scale and to encourage to think and examine how the achievements in this process can be processed in different scales. In accordance with this purpose, satellite images and map images of a game are handled as data using a GAN model. The model, which is trained with satellite and game maps images contain different textures of nature, different types of transportation networks, settlements of different densities and qualities, can perform two-way transformations (game map from satellite image and satellite image from game map). The study focused on the generation of game maps from satellite images, and during the process, tests for the generation of game maps from satellite images of different locations around the world and analyzes based on these tests were conducted. As the last step of the study, to obtain the most successful results, the model inputs were optimized with the findings obtained from the analyzes and by using satellite images of these regions, the game maps of 'Istanbul Historical Peninsula' and 'Istanbul Göksu' regions were produced.

Keywords: Generative Adversarial Networks, Machine Learning, Map Generation

Received: 25.08.2020

Accepted: 22.09.2020

Corresponding Author:

baris@virtualogie.com

Balci, O., Terzi, Ş.B., Balaban, Ö. (2020). Map Generation & Manipulation with Generative Adversarial Networks. JCoDe: Journal of Computational Design, 1(3), 95-114.

Çekişmeli Üretici Ağlar (Gan) ile Harita Üretimi ve Manipülasyonu

Ozan Balcı¹, Şemsi Barış Terzi², Özgün Balaban³

0000-0002-9265-2262¹, 0000-0002-3400-7839², 0000-0002-7270-2058³

¹ İstanbul Teknik Üniversitesi, Mimarlık Fakültesi, Mimarlık Bölümü, İstanbul, Türkiye

² İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilişim Anabilim Dalı, Mimari Tasarımda Bilişim, İstanbul, Türkiye

³ Chair of Design Informatics, Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, Hollanda

Yapay zeka, internetin yaygınlaşması ile yoğunlaşan veri birikimini işlenebilir kılmıştır. Bir yapay zeka türü olan makine öğrenmesi verilerin sınıflandırılması, anlamlandırılması ve işlenmesini kolaylaştırmıştır. Grafik işleme ünitelerinin gelişmesi ise görüntülerden oluşan verilerin işlenmesinde gerekli altyapıyı sağlayarak, yapılan çalışmaları kolaylaştırmış ve hızlandırmıştır. Görüntü işleme süreçlerinin gelişimi görüntülerden oluşan mimari veri setlerinin işlenmesini ve üretilmesini sağlayabilecektir. Bu çalışmada amaç; harita ölçeğindeki görüntü setinin işlenebilirliğinin gösterilmesidir. Bu gösterimden hareketle görüntü işlemenin mimari gösterim tekniklerinde ve farklı ölçeklerde nasıl olabileceğine dair tartışma açabilmektedir. Bu amaç doğrultusunda bir çekişmeli üretici ağ modeli kullanılarak uydu görüntüleri ile bir oyuna ait harita görüntüleri veri olarak ele alınmıştır.

Uydu ve oyun haritasında bulunan, doğaya ait farklı dokular, farklı tip ulaşım ağları, farklı yoğunlukta ve nitelikte olan yerleşim yerleri içeren görüntüler ile eğitilen model, çift yönlü (uydu görüntüsünden oyun haritası ve oyun haritasından uydu görüntüsü) olarak dönüşümler gerçekleştirebilmektedir. Çalışmada, uydu görüntülerinden oyun haritaları üretimine odaklanılmış olup, süreç boyunca dünya üzerinde farklı lokasyonlara ait uydu görüntülerinden oyun haritaları üretimine yönelik testler ve bu testleri baz alan analizler yapılmıştır. Çalışmanın son aşaması olarak analizlerden elde edilen bulgular ile model girdileri en başarılı sonuçlar alınacak şekilde optimize edilmiş ve 'İstanbul Tarihi Yarımada' ve 'İstanbul Göksu' bölgelerine ait uydu görüntüleri kullanılarak bu bölgelerin oyun haritaları üretilmiştir.

Anahtar Kelimeler: Çekişmeli Üretici Ağlar, Makine Öğrenmesi, Harita üretimi

Teslim Tarihi: 25.08.2020

Kabul Tarihi: 27.09.2020

Sorumlu Yazar:

baris@virtualogie.com

Balcı, O., Terzi, Ş.B., Balaban, Ö. (2020). Çekişmeli Üretici Ağlar (Gan) İle Harita Üretimi Ve Manipülasyonu. JCoDe: Journal of Computational Design, 1(3), 95-114.

1. GİRİŞ

İnternetin yaygınlaşması ile içinden çıkılmaz bir yoğunluğa ulaşan veri birikimi, günümüzde büyük veri denilen kavramı oluşturmuştur. Bu veri birikiminin karmaşıklığı, sınıflandırılma ve işleme zorluğu büyük verinin değerlendirilmesinin önünde bir engel olarak görülmüştür. Büyük verinin anlamlandırılıp işlenmesi, makine öğreniminin gelişimi ile son yıllarda mümkün hale gelmiştir. Büyük verinin bir kısmını oluşturan görüntülerin sayısal görüntülere dönüştürülmesi ve günümüzde bilgisayarların grafik işlem ünitelerinin hızlı gelişimi sayesinde makine öğrenmesi algoritmaları tarafından işlenmesinin mümkün kılınması, çalışmaların hızlanmasını ve kolaylaşmasını sağlamıştır. Bu çalışmada verilerin işlenebilmesi için Python kodlama dili ve Google Colab operatörü kullanılmıştır.

1.1 Görüntü İşleme

Görüntü işleme görüntülerin sayısal verilere dönüştürülüp üzerinde işlemler yapılmasını sağlayarak anlamlandırılmasını, maniple edilmesini ve yeni görüntüler elde edilmesini sağlamaktadır. Bu görüntüler makine öğrenmesi yöntemleri ile de işlenebilir. Python kodlama dilinde birçok modül ile görüntü işleme yapılabilir. 'Numpy', 'OpenCV', 'matplotlib', 'PIL' bu modüllerden bazılarıdır. Bu modüller sayesinde görüntüler, her pikselinin RGB (red – green - blue) veya HSV (hue – saturation - value) gibi renk uzaylarındaki karşılıklarından oluşan matrislere dönüştürülebilir, bu verilerden grafikler oluşturulabilir ve geri görüntülere dönüştürülebilir (Gezer, 2018).

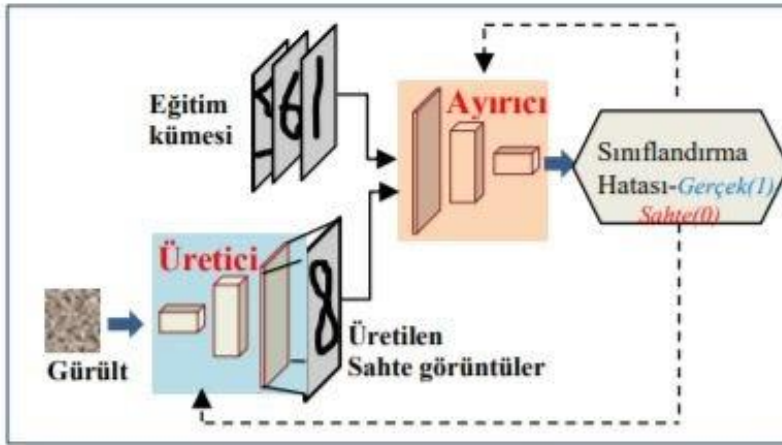
1.2 Derin Öğrenme, Makine Öğrenmesi Ve Yapay Zeka

Makine öğrenmesi örnekler ve verilerle öğretilen algoritmaların veriler üzerinden çıkarımlar yaparak yeni veriler üretmesi olarak tanımlanabilir (Mitchel, 1997). Yapay zeka ise 'Doğal sistemlerin yapabildiği (zekice olsun veya olmasın) her bilişsel etkinliği (gerekirse bedenleri olan) yapay sistemlere, daha da yüksek başarımlı düzeylerinde nasıl yaptırabileceğimizi inceleyen bilim dalıdır.'(Say, 2018) Dolayısıyla makine öğrenimi, bir yapay zeka yöntemi olarak ele alınabilir. Bir makine öğrenmesi sınıfı olan derin öğrenme ise çıkarımda bulunmak ve yeni veriler üretmek için öğrenme işleminde kullanılan verilerin birçok katmanının doğrusal olmayan bir şekilde kullanılmasıdır. Derin öğrenmede girdiler üzerindeki birden çok

katmanın öğrenilmesi ve bu katmanlardan oluşturulan hiyerarşik bir düzen söz konusudur (Şeker ve diğ., 2017).

1.3 Çekişmeli Üretici Ağlar Ve Türleri

Öğrenme için kullanılan girdilerin karmaşık yapılarını öğrenen derin öğrenme teknikleri son yıllarda büyük bir ilerleme katetmiştir. (Wu ve diğ., 2017) Ian Goodfellow ve ekibi tarafından 2014 yılında üretilen görüntü işleme yeteneğine sahip ilk model olan çekişmeli üretici ağlar (generative adversarial network, GAN), derin öğrenmeye görüntü işleme konusunda büyük yetenek kazandırmıştır. (Wason, 2018) (Goodfellow ve diğ., 2014) Çekişmeli üretici ağlarda üretici (generative) ve ayırıcı (discriminator) olarak iki algoritma çalışır (**Şekil 1**). Üretici öğrenim verilerinden aldığı verilere benzer veriler üretmeye çalışırken ayırıcı bu verilerin gerçek veya sahte olduğunu anlamaya çalışır. Üreticinin üretmiş olduğu veriler ayırıcı tarafından değerlendirilir ve ayırıcının üreticiye verdiği puanlar doğrultusunda üretilen yeni veriler birçok tekrar (epoch) sonrasında gerçek verilere benzemeye başlar. Bu modelin önerilmesinden sonra birçok GAN modeli oluşturulmaya başlanmıştır. Üreticiye verilen bir koşulla üretim yapabilen cGAN, veride havuzlamanın kullanılmadığı DCGAN, üreticinin oluşturduğu gürültüye gizli kod veya ek bilgi eklenen InfoGAN çekişmeli üretici ağ modellerinden bazılarıdır. Ayrıca bu makalede daha detaylı bahsedilecek pix2pix ve CycleGAN da çekişmeli üretici ağ modelleridir.

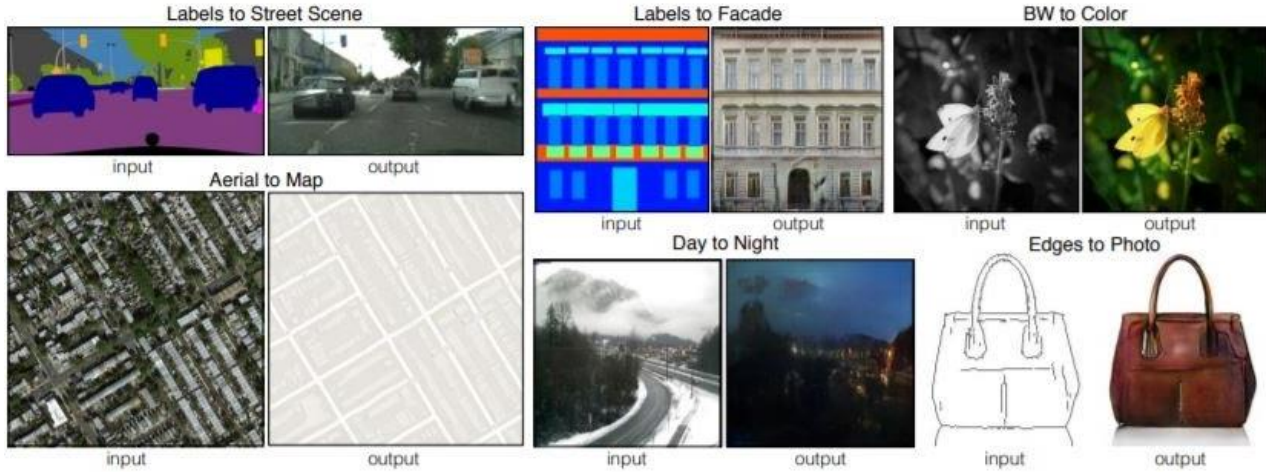


Şekil 1: Çekişmeli üretici ağ çalışma prensibi (Çelik & Talu, 2019)

2. ÇEKİŞMELİ ÜRETİCİ AĞLAR VE ÇÖZÜMLEME SÜRECİ

Görüntü işleme ve çözümlemede yapay zeka kullanımında çığır açan çekişmeli üretici ağlar girdi olarak üretilmek istenen çıktılara benzer görüntülere ihtiyaç duymaktadır. GAN modelleri arasında bulunan pix2pix ve CycleGAN, Berkeley Yapay Zeka Araştırma (BAIR) Laboratuvarından bir ekibin ürettiği modellerdir. Pix2pix görüntüden görüntüye dönüştürme modelidir (**Şekil 2**). Görüntülerdeki pikselleri birebir eşleştirerek çalışmaktadır. Bu modelde öğrenme için görüntüler sınıflandırılarak birbirini karşılayan -aynı görselin siyah beyazı, bir ürünün çizimi ve fotoğrafı veya aynı noktadan çekilmiş bir fotoğrafın gecesini gündüzü- veri kümeleri oluşturulmaktadır. Bu veri setleri ile eğitilen model yeni görüntüler üretirken bir koşula ihtiyaç duymaktadır. Bu model ile bir gündüz görselinin gecesini, bir çizimin veya boyamanın gerçekçi görüntüsü oluşturulabilmektedir (Isola ve diğ., 2018).

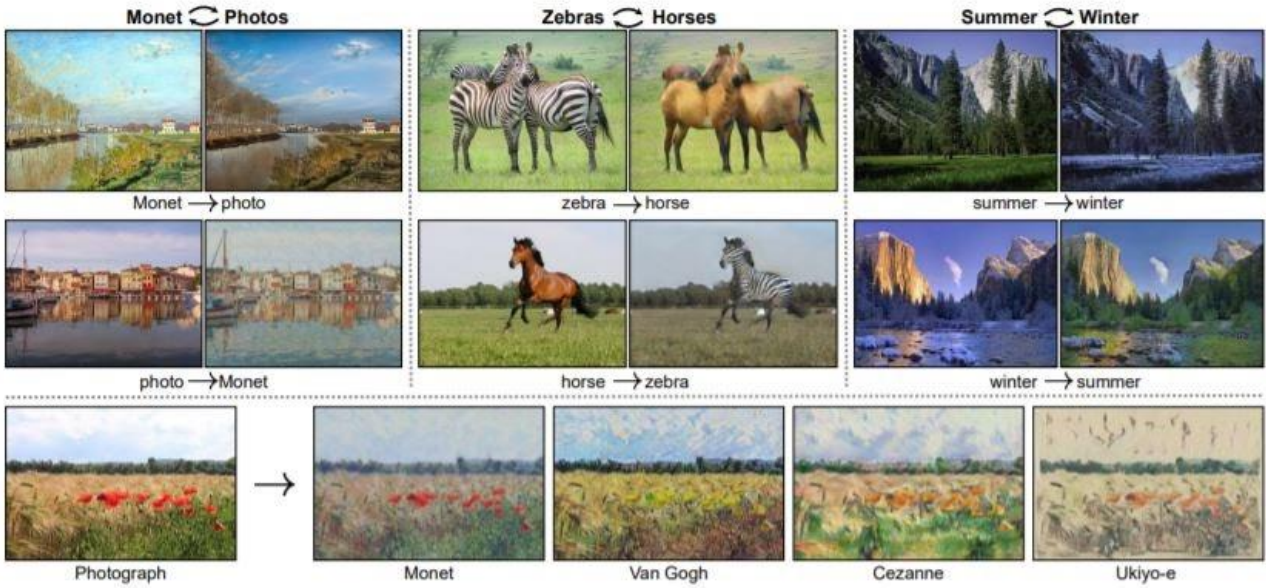
Şekil 2: Pix2pix örnek uygulamaları (Isola ve diğ., 2018)



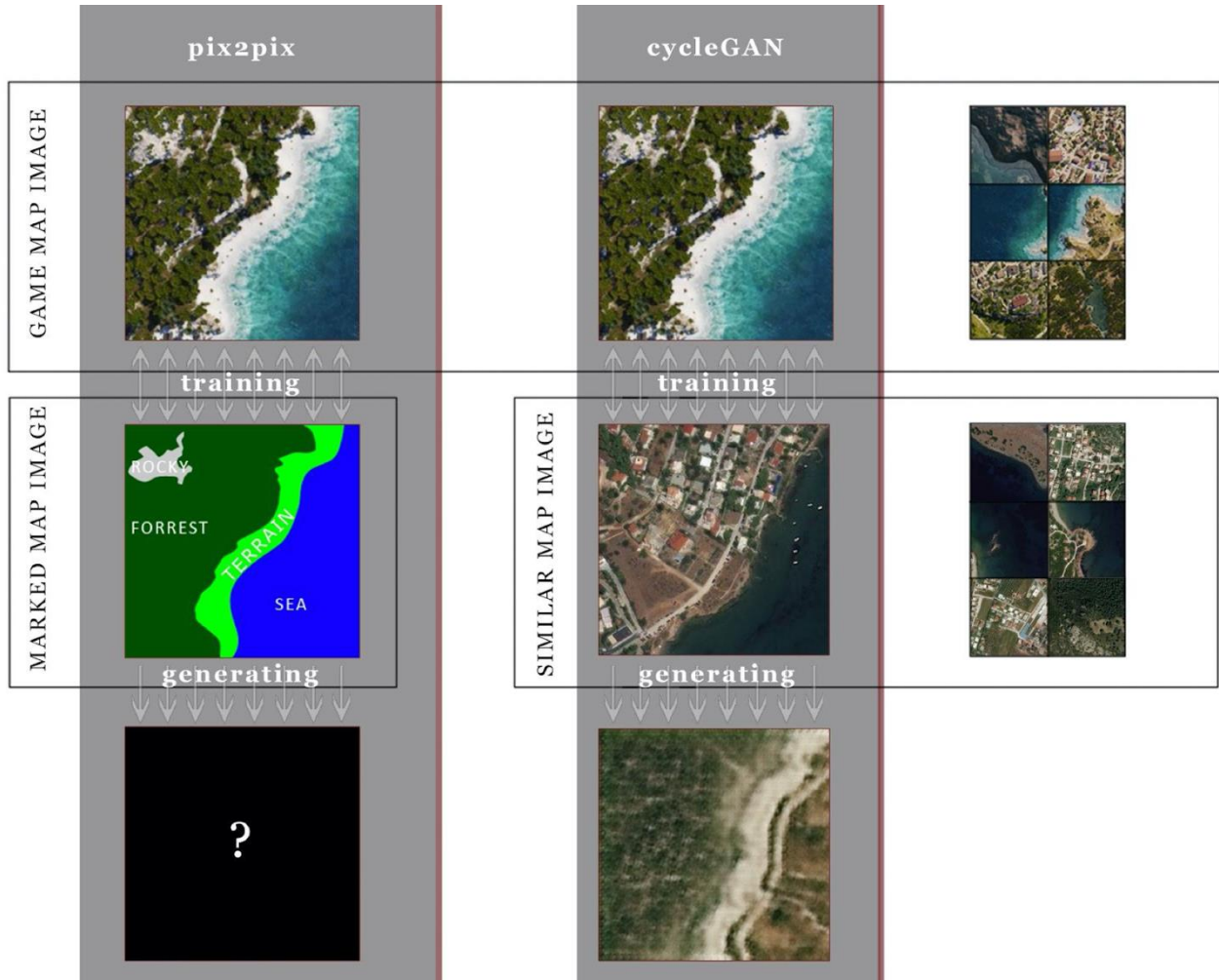
CycleGAN ise pikselleri birebir eşleşen veri setlerine değil sadece benzer veri setlerine ihtiyaç duyan bir GAN modelidir (**Şekil 3**). CycleGAN'ın anlaşılması için verilebilecek en temel örnek zebra – at (zebra - horse) çalışmasıdır. Zebra ve at görüntüleri ile eğitilen algoritma test veri kümelerindeki at görüntülerini zebraya, zebra görüntülerini ise ata dönüştürebilmektedir. CycleGAN, pix2pix modelinin aksine çift taraflı çalışabilmektedir. Böylece tersine uygulanan işlemlerde algoritmanın yeniden ve başka verilerle eğitilmesi gerekmemektedir.

Algoritma iç içe iki GAN modelini kullanarak görüntüyü önce ulaşılmak istenen görüntüye sonra ise tekrar eski haline dönüştürmektedir. Dönüştürülen ulaşılmak istenen görüntünün, üreticinin ürettiği görüntüye ne kadar uyum sağladığını gösteren ve asıl haline ne kadar benzediğini gösteren iki farklı kayıp oluşturulmaktadır. Daha sonra ayırıcının etkisi ile bu kayıplar yapılan tekrarlar sonucu azaltılarak görüntünün dönüşümü sağlanmaktadır. Fotoğrafları ünlü ressamların eserlerine dönüştüren veya tam tersini; yaz fotoğrafını kış fotoğrafına dönüştüren veya tam tersini yapan algoritmalar da CycleGAN modellerine örnek olarak verilebilir (Isola ve diğ., 2018).

Şekil 3: CycleGAN örnek uygulamaları (Isola ve diğ., 2018)



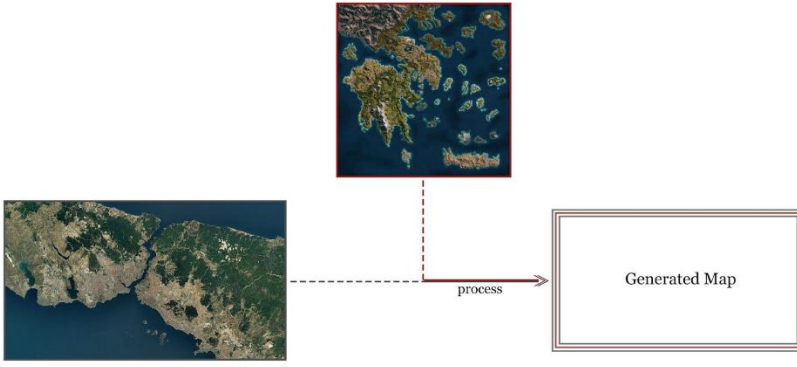
Bu çalışmada kullanılması hedeflenen GAN modeli belirlenirken bu iki model değerlendirilmiştir. Pix2pix modelinin eşlenmiş veriye ihtiyaç duyması, eğitimde kullanılacak verilerin önceden işlenmesi gerekliliğini doğurmuştur. Eğitim ve üretilmesi istenen test verilerinin işlenmesinin zorluğu ve modelin eğitilmesinde eşlenmiş veriye ihtiyaç duyulmaması nedeniyle CycleGAN kullanılması kararlaştırılmıştır (Şekil 4).



3. İŞLEM SÜRECİ

Şekil 4: Çekişmeli üretici ağlardan cycleGAN ve pix2pix karşılaştırma diyagramı

Çalışmada, belirlenmiş bir ölçekte elde edilen uydu görüntülerini kullanarak oyun haritaları üretilmesi hedeflenmiştir (**Şekil 5**). Bir diğer deyişle çalışmanın amacı; dünya üzerinde bulunan bir lokasyonun haritasının, seçilen oyuna ait harita stilinde yeniden üretilmesidir. Oyun olarak Ubisoft tarafından geliştirilen 'Assassin's Creed: Odyssey' tercih edilmiş ve bahsi geçen oyunun 'Greece' isimli haritası kullanılmıştır.



Şekil 5: Harita üretim diyagramı

Çalışma, bir CycleGAN modeli olan 'Summer to Winter Yosemite' kodu üzerinden geliştirilmiş olup, süreç esnasında veri seti olarak Google Haritalar ve Bing Haritalara ait uydu görüntüleri ve bahsi geçen oyuna ait 'Greece' haritası kullanılmıştır.

3.1 Çalışma Planı

Çalışmanın geliştirilmesi için tamamlanması gereken adımlar, çalışma takviminde 3 farklı iş paketi altında toplanmıştır. Bunlar:

1. Veri seti oluşturulması
2. Modelin eğitilmesi
3. Çıktıların değerlendirilmesi

Bahsi geçen iş paketleri, eş zamanlı değil, yukarıda verilen sıra ile ardışık olarak çalışılmış ve hedeflenen sonuç elde edilene kadar süreç devam etmiştir.

3.2 Veri Seti Oluşturulması

CycleGAN modelinin ihtiyacı olan iki ayrı veri kümesi bu çalışma özelinde 'Oyun Haritası' ve 'Uydu Haritası' olarak isimlendirilmiştir.

'Oyun Haritası' isimli veri kümesi, daha önce belirtilen oyunun haritasını oluşturan görüntüleri içermektedir. İstenen harita, çeşitli interaktif oyun haritalarını bünyesinde barındıran Mapgenie.io web sitesinde saptanmış ve veri kazıma (data scraping) için yöntem geliştirilmiştir.

İlk olarak haritanın, koordinat sistemi üzerine yerleşmiş 256*256 piksel boyutlarında görüntülerden oluştuğu tespit edilmiştir. Bu görüntülerin bağlantı adreslerinde, koordinat sistemindeki yerlerine karşılık gelen değerlerinin örüntüsü tespit edilip, Python dilinde yazılan bir komut dizisi ile tüm görüntüler elde edilmiştir.

Örnek olarak 'https://cdn.mapgenie.io/images/tiles/ac-odyssey/greece/11/585/563.jpg' bağlantı adresinde '11' görüntünün ölçeğini, '585' görüntünün X koordinatını, '563' ise Y koordinatını temsil etmektedir.

Aynı metot, Python kodu ile veri kazıma, diğer veri kümesi olan 'Uydu Haritası' oluşturulurken de kullanılmıştır. 'Oyun Haritası' veri kümesinden farklı olarak 'Uydu Haritası' için iki farklı kaynağa ait veriler test edilmiştir. İlk çalışmalar Google Haritalar üzerinden yapılmıştır. Fakat veriler elde edilirken belirli periyotlarda işlemin Google tarafından engellenmesi, bölgeye göre uydu görüntü kalitelerinin farklılaşması, deniz görüntülerinde Google etiketlerinin bulunması gibi birtakım sebeplerden dolayı 'Uydu Haritası' veri kümesi için yeni bir kaynak arayışı gerekli olmuştur. İncelenen alternatif kaynaklar arasında Bing Haritalar'da karar kılınmış ve veri kazıma çalışmaları Bing üzerinden devam etmiştir. Bu verileri oluşturan uydu görüntüleri 2020 yılına aittir ve Bing Haritaların tanımladığı 17. yakınlaştırma seviyesindedir. Örnek olarak 'https://www.bing.com/maps ... cp=37.936172~23.950817&lvl=17&style=h&v=2&sV=2&form=S00027' bağlantı adresinde 'cp=37.936172~23.950817' koordinatları, lvl=17' yakınlaştırma seviyesini belirtmektedir.

İki veri kümesi ('Oyun Haritası' ve 'Uydu Haritası') için üç farklı kaynaktan 50.000'in üzerinde veri elde edilmiş ve bu veriler niteliklerine göre iki ana başlık altında sınıflandırılmıştır;

- Tip A: Deniz içermeyen (sadece kara içeren) görüntüler
- Tip B: Deniz içeren görüntüler
 - Tip B.1: Sadece deniz içeren görüntüler
 - Tip B.2: Kara ve denizi aynı anda içeren görüntüler

Yukarıda belirtildiği gibi sınıflandırılan veriler farklı kombinasyonlarda kullanılarak farklı veri setleri elde edilmiş ve bu veri setlerinin detaylarına aşağıdaki tabloda (**Tablo 1**) yer verilmiştir.

No	Versiyon	'Oyun Haritası' Veri Kümesi		'Uydu Haritası' Veri Kümesi	
		Kaynak	Kullanılan Veri Tipi ve Adedi	Kaynak	Kullanılan Veri Tipi ve Adedi
1.	Versiyon 1.0	Mapgenie.io (AC: Odyssey, Greece Map)	1500 adet Tip A ve Tip B	Google Maps	1500 adet Tip A ve Tip B
2.	Versiyon 2.0	Mapgenie.io (AC: Odyssey, Greece Map)	1500 adet Tip A ve Tip B	Bing Maps	1500 adet Tip A ve Tip B
3.	Versiyon 2.1 1. Modül	Mapgenie.io (AC: Odyssey, Greece Map)	1500 adet Tip A	Bing Maps	1500 adet Tip A
4.	Versiyon 2.1 2. Modül	Mapgenie.io (AC: Odyssey, Greece Map)	1300 adet Tip B.1 200 adet Tip B.2	Bing Maps	1300 adet Tip B.1 200 adet Tip B.2

Tablo 1: Veri setleri içerik tablosu

3.3 Modelin Eğitilmesi

Veri setlerinin yanı sıra, tekrar değeri, veri olarak kullanılan görüntülerin boyutları ve yineleme (iteration) değeri çıkan sonuçlar üzerinde etkisi olan başlıca girdilerdir. Çalışma sürecinde bahsi geçen girdilere aşağıda bulunan tablodaki (**Tablo 2**) değerler verilip sonuçları analiz edilmiştir.

Tekrar Değeri	Görüntü Boyutu	Veri Seti	Yineleme Değeri
3000	128 * 128 piksel	Versiyon 1.0	200
-	256 * 256 piksel	Versiyon 2.0	500
-	-	Versiyon 2.1 (1. Modül)	1000
-	-	Versiyon 2.1 (2. Modül)	2500
-	-	-	3000

Tablo 2: Girdiler tablosu

Özet olarak model 8 farklı düzende eğitilmiş ve bu 8 eğitimden elde edilen 40 farklı sonuç tablosu incelenip başarılı kabul edilen model kullanılarak çalışmanın sonuç ürünleri olan 'İstanbul - Tarihi Yarımada' ve 'İstanbul - Göksu' haritaları üretilmiştir.

3.4 Çıktıların Analizi

Elde edilen çıktılar analiz edilirken her bir girdinin öğrenimi nasıl etkilediğini tespit etmek amacıyla analizlerde tüm girdilerin aynı fakat sadece analiz başlığında yazan girdinin farklı olduğu sonuçlar karşılaştırılmıştır. İncelemeler, modelin 'Uydu Haritası' veri kümesinden 'Oyun Haritası' veri kümesine uyarlayarak ürettiği

sonuçları kapsamaktadır. Diğer (oyun haritasından uydu görüntüleri elde edilen) sonuçlar göz önünde bulundurulmamıştır.

Değerlendirme başlığı altında kullanılan şekiller 4*8'lik bir ızgara yapı üzerine yerleşmiş 32 adet görüntüden oluşmaktadır. Birinci, üçüncü, beşinci ve yedinci sütunlarda bulunan 16 görüntü 'Uydu Haritası' veri setine ait görüntüler; ikinci, dördüncü, altıncı ve sekizinci sütunlarda bulunan 16 görüntü ise modelin ürettiği görüntüler olmaktadır.

3.4.1 Görüntü Çözünürlüğünün Öğrenime Etkisi

Eş veri sayısı ve veri içeriğine sahip fakat sadece görüntü boyutlarında farklılık olan veri setleri kullanılarak 3000 tekrar değerinde eğitilmiş iki modelin 1000 yineleme değerinde elde edilen görüntüleri aşağıda verilmiştir.

Şekil 6: 128*128 piksel boyutlarında veriler kullanılarak eğitilen model





Görüntü boyutunun 128*128 pikselden 256*256 piksele yükseltilmesi, sonuçları çözünürlük ve detay seviyesi açısından olumlu etkilemiştir. Fakat bunun yanı sıra tespit edilen bir diğer farklılık, çözünürlüğün artması ile eğitilen modelin öğrenmesi gereken doku tipi miktarı artmıştır ve bu da dönüşüm esnasında bazı dokuların kimliğini kaybetmesine sebep olmuştur. Örnek olarak 128*128 piksel boyutlarında elde edilen sonuçlarda orman, tarla ve toprak dokuları korunmuşken, 256*256 piksel boyutlarında elde edilen sonuçlarda ormanların kayalık dokusuna, tarlaların toprak dokusuna ve toprakların da karanlık bir dokuya dönüştüğü görülmektedir (**Şekil 6**), (**Şekil 7**). Modelin eğitilmesinde işlem gücü yetersizliği ve eğitim sürelerinin uzaması nedenleriyle daha yüksek çözünürlüklerde veriler kullanılmamıştır; çözünürlük, işlem gücü ve işlem süresi optimizasyonu sağlanmıştır.

Şekil 7: 256*256 piksel boyutlarında veriler kullanılarak eğitilen model

3.4.2 Tekrar Değerinin Öğrenime Etkisi

Tekrar değerinin artması, daha fazla yineleme imkanı vermektedir. Bu durum doğrultusunda aynı veri seti kullanılarak yüksek tekrar değeri ile eğitilen algoritmanın hata değeri düşüş göstermektedir. Aşağıda verilen grafiklerde tekrar sayısının artması hata değerinin gittikçe düştüğünü ve yeterli tekrardan sonra belirli bir aralıkta seyrettiğini göstermektedir. Bu nedenle çalışmada yeterli tekrar değeri 3000 olarak belirlenmiş ve eğitim bu tekrar değeri ile yapılmıştır. Girdi verileri ile eğitilen algoritmada ayırıcı, üreticinin ürettiği gürültünün hata değerini puanlamış ve üretici tekrar arttıkça parabolik bir azalma gösteren hata değeri ile yeni veriler üretmiştir.



Şekil 8: Hata değer grafiği

3.4.3 Verileri Sınıflandırmanın Öğrenime Etkisi

Veri kümelerinin birbirine benzerliğinin sonuç üzerindeki etkisini gözlemlemek amacıyla tüm girdilerin aynı olduğu fakat veri kümelerinin farklı olduğu denemeler yapılmıştır. İlk olarak **Tablo 1**'de Versiyon 2.0 isimli veri setleri kullanılmış, Tip A (deniz içermeyen) ve Tip B (deniz içeren) verileri aynı modele öğretilmiştir. Ardından Versiyon 2.1 isimli veri setleri kullanılarak Tip A (deniz içermeyen) verilerini barındıran 1. modül bir modele, Tip B (deniz içeren) verilerini barındıran 2. modül bir diğer modele öğretilmiştir. 256*256 piksel boyutlarında görüntüler kullanılarak 1000 tekrar ve 1000 yineleme değerlerinde elde edilen sonuçlar aşağıdadır.



Şekil 9: Versiyon 2.0 isimli veri setleri ile eğitilen model



Şekil 10: Versiyon 2.1 - 1. Modül isimli veri setleri ile eğitilen model



Şekil 11: Versiyon 2.1 - 2. Modül isimli veri setleri ile eğitilen model

Elde edilen sonuçlar göz önünde bulundurulduğunda verilerin sınıflandırılması çıkan sonuçları olumlu yönde etkilemiştir. Versiyon 2.0 veri setleri ile yapılan çalışmada kara dokusu olması gereken yerlerde denize rastlanmıştır (**Şekil 9**). Versiyon 2.1 - 1. Modül ile yapılan çalışmada iki veri kümesinde de deniz görüntüleri bulunmadığından dolayı, deniz - kara dokularının birbirlerine karışması ile ilgili bir hataya rastlanılmamıştır (**Şekil 10**). Fakat aynı versiyonun 2. modülü, kara ve deniz görüntülerini içeren, ile yapılan çalışmada deniz - kara dokularının birbirlerine karışmasından doğan hatalar mevcuttur (**Şekil 11**). Söz konusu hataların Versiyon 2.0 ile yapılan çalışmalara kıyasla daha az olduğu görülmektedir.

3.5 Örnek Çalışma: İstanbul Haritası Üretimi

Süreç boyunca edinilen bilgiler ışığında, çalışmanın son aşaması olarak İstanbul - Tarihi Yarımada ve İstanbul - Göksu haritaları üretilmesi planlanmıştır. Gerekli olan uydu görüntüleri Bing Haritalar'dan elde edilmiş (**Şekil 12a**) (**Şekil 13a**), Versiyon 2.1 veri setleri kullanılarak 2 farklı modül ile eğitilen modeller üzerinden haritalar üretilmiştir (**Şekil 12b**) (**Şekil 13b**). 3000 tekrar değerinde, 128*128 piksel boyutunda görüntüler kullanılarak eğitilen modelin 1000 yineleme değerindeki çıktıları aşağıda verilmiştir.



Şekil 12 (a): Tarihi Yarımada uydu görüntüsü

Şekil 12 (b): Tarihi Yarımada üretilen görüntü



Şekil 13 (a): Göksu uydu görüntüsü

Şekil 13 (b): Göksu üretilen görüntü

Örnek çalışmalarda üretilen haritalar değerlendirildiğinde bina ve orman dokularının, kıyı şeritlerinin başarılı bir şekilde oluşturulduğu görülmektedir. Bunların yanısıra Göksu örneğinde görüldüğü üzere model, oyun haritasında olmayan otoyolları ve köprüleri, aynı amaca (ulaşım ağı) hizmet eden patikalar olarak üretmiştir. Tarihi Yarımada örneğinde ise yine oyun haritasında bulunmayan teknelerin ve suda bıraktıkları izlerin başarılı bir şekilde üretildiği görülmektedir. Ancak yine aynı örnekte uydu görüntüsünde deniz ve yeşil alanların renk

değerlerinin yakınlığı nedeniyle model, yoğun yeşil alan olan yerleri deniz olarak algılayıp hatalı bir üretim gerçekleştirmiştir.

4. SONUÇ

Bu çalışma harita ölçeğindeki görüntü setinin işlenebilirliğini göstermek ve bu süreçteki kazanımların farklı ölçekteki mimari gösterim tekniklerini nasıl işleyebileceğini düşündürmek için yapılmıştır. Bu süreçte model farklı veri setleri, tekrar değerleri ve görüntü boyutları ile eğitilmiş, 3.4 bölümünde anlatıldığı gibi çıktılar analiz edilmiştir.

Elde edilen sonuçlar doğrultusunda,

- 1000 - 2000 arası tekrar değeri;
- 256*256 piksel boyutlarındaki görüntülerden oluşan veri seti;
- 1500 - 2000 adet görüntü içeren, verileri Tip A ve Tip B biçiminde sınıflandırılarak oluşturulan veri kümeleri;

ile öğrenimi yapılacak modelin, harita üretimi için başarılı sonuçlar vereceği öngörülmektedir.

Kullanılan CycleGan modelinde, eğitimde kullanılan 'Oyun Haritası' ve 'Uydu Haritası' veri tiplerinin birebir eşleşmesine gerek duyulmamıştır. Model dokular arası renk geçişlerinde başarılı olmuş, eğitim verisinde karşılığı olmayan objeleri üretebilmiştir. Bu olumlu sonuçlar karşısında bazı test veri setlerinde benzer dokuları birbirine karıştırmış, farklı açılarda ve şiddetlerdeki gölgeleri farklı dokular olarak üretmiştir. Ayrıca farklı ölçeklerdeki 'Oyun Haritası' ve 'Uydu Haritası' veri tiplerinde başarısız sonuçlar ortaya koymuştur. Her ne kadar CycleGAN modeli olumsuz yönlere sahip olsa da çalışmayı belirli bir noktaya taşımıştır.

Çalışma süreci boyunca verilerin, veri setinin ve diğer girdilerin CycleGAN modelinde sonuç ürün üzerindeki etkileri tespit edilmiş; girdilerin optimizasyonu konusunda bilgiler elde edilmiştir. Bunun yanı sıra çeşitli veri kazıma metotlarının öğrenilmesi de çalışma kazanımları arasında yerini almıştır.

Yapılan çalışmalar sonucunda, CycleGAN modelinin görüntü işleme yeteneğinin haritalar üzerindeki etkinliği gözlemlenmiştir. Kullanılan veri setlerinde değişiklik gösterilerek farklı görüntüler üretilmesine olanak sağlayan bu modelde iki farklı harita seti ile üretilen görüntüler, görüntü işlemenin mimari gösterim tekniklerindeki potansiyeli hakkında fikir vermektedir. Harita ölçeğinde yapılmış olan bu çalışma, kent veya yapı gibi farklı ölçeklerde geleneksel gösterim tekniklerinin makine öğrenmesi ile işlenebilirliğini göstermektedir. Mimari teknik çizimlerin, diyagramatik çizimlere; uydu görüntülerinin farklı haritalamalara dönüştürülebileceği çalışmalar bunlara örnek olarak gösterilebilir. Bu model ile, yine harita ölçeğinde, tarihin farklı dönemlerine ait varsayımsal haritalamalar, gelecekte farklı gezegenlerdeki kolonileşmelere yönelik simülasyonlar, geçmişin farklı seyrettiği varsayımlar olarak geliştirilen günümüz senaryolarına ait kentleşmeleri konu edinen çalışmalar; mimari ölçeğe indiğimizde ise canlandırım için belirlenen stile uygun hızlı ve sınırsız sayıda plan ve cephe dokuları üretilebilir.

Teşekkür

Bu çalışma İstanbul Teknik Üniversitesi, Mimari Tasarımda Bilişim Yüksek Lisans programı 2019-2020 Bahar Yarıyılı dersi olan Mimari Tasarımda Özel Konular: Makine Öğrenmesi kapsamında yapılan bir projeden üretilmiştir.

Referanslar

Çelik, G., Talu, M. F., (2019). Çekişmeli Üretken Ağ Modellerinin Görüntü Üretme Performanslarının İncelenmesi. *BAUN Fen Bil. Enst. Dergisi*, 22(1), 181-192. doi:<https://doi.org/10.25092/baunfbed.679608>

Gezer, M. (2017). "Python ile Görüntü İşlemede Örnek Bir Uygulama", Ed.: Uçar, M. Kürşad., Mühendislikte Yapay Zeka Uygulamaları, Sakarya, *Sakarya Üniversitesi Kütüphanesi Yayınları*, 65-79. isbn: 978-605-4735-98-3

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. ve Bengio, Y, (2014). "Generative Adversarial Networks", In *Proc. 27th Int. Conf. Neural Information Processing Systems*, (pp. 2672–2680), Montreal. arXiv:1406.2661

Isola, P., Zhu, J., Park, T., Efros, A. A., (2018) Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. arXiv:1703.10593v6

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2018). Image-to-Image Translation with Conditional Adversarial Networks. arXiv:1611.07004v3

Mitchell, Tom M. (1997). Machine Learning, New York, McGraw-Hill Science isbn: 0070428077

Şeker, A., Diri, B., Balık, H. H., (2017). Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme. *Gazi Mühendislik Bilimleri Dergisi*, 3(3), 47-64.

Wason, R., (2018). Deep Learning: Evolution and Expansion, *Cognitive Systems Research*, 52, 701-708. doi: <https://doi.org/10.1016/j.cogsys.2018.08.023>

Wu, X., Xu, K. ve Hall, P., (2017). A survey of Image Synthesis and Editing with Generative Adversarial Networks, *Tsinghua Science and Technology*, 22(6), 660–674. doi: <http://doi.org/10.23919/TST.2017.8195348>

