

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/382524122>

Tek Katlı Yapılar için Çekişmeli Üretici Ağlarda Ölçeklendirme ve Kontrol Yöntemleri

Conference Paper · July 2024

CITATIONS

0

READS

48

3 authors, including:



Mertcan Güldilek

Izmir Institute of Technology

2 PUBLICATIONS 0 CITATIONS

SEE PROFILE



Berk Ekici

Izmir Institute of Technology

34 PUBLICATIONS 442 CITATIONS

SEE PROFILE

Tek Katlı Yapılar için Çekişmeli Üretici Ağlarda Ölçeklendirme ve Kontrol Yöntemleri

Mertcan Güldilek¹ ; Berk Ekici²  Mustafa Emre İlal³ 

^{1,2,3}İzmir Yüksek Teknoloji Enstitüsü


¹mertcanguldilek@gmail.com; ²emreilal@iyte.edu.tr; ³berkekici@iyte.edu.tr

Özet

Yapay zeka (YZ) teknolojilerinin, birçok alanda insanlardan daha hızlı ve etkili çözümler sunabilme kapasitesine sahip olduğu gözlemlenmektedir. Mimarlıkta ise, YZ'nin daha işlevsel ve daha yüksek başarılı yapılar oluşturması beklenmektedir. YZ ile mimari plan tasarımı üretme çalışmalarında Çekişmeli Üretici Ağlar (ÇÜA - Generative Adversarial Networks - GAN) kullanımı literatürde tercih edilmektedir. Sunulan bu çalışma kapsamında, literatürde var olan tek katlı bina plan çözüm kütüphanelerinden öğrenip yeni planlar oluşturabilen bir model geliştirilmiştir. U-Net Mimarisi ve Uygulamalı Evrimsel Mesaj Aktarımı (UEMA) ÇÜA modeline entegre edilmiş ve modelin odalar arası ilişkileri daha iyi öğrenmesi ve kenar algısını güçlendirmesi sağlanmıştır. Eğitimi tamamlanmış model için pikselleri gerçek alanlara çevirmek üzere bir ölçeklendirme yöntemi geliştirildikten sonra kullanıcının üretilen planlar üzerinde kontrolün sağlanması için iki ayrı yaklaşım araştırılmıştır. İlk yaklaşım model üretirken oda boyutlarını kısıtlamaya, ikinci yöntem ise üretilen planların kısıtlara uymayanlarını kabul etmemeye dayalıdır. İki yaklaşım da başarılı sonuçlar verirken, ilk yaklaşımda odalar arası ilişkilerde sorunlar gözlemlenmiş, ikinci yaklaşımda ise plan üretim süreci uzamıştır. Kullanıcılar, geliştiren sistemde odaların boyutlarını ve planın toplam metrekaresini belirleyerek tek katlı plan tasarımları üretebilmektedirler. Bu araştırma, tasarım süreçlerinde kullanıcı merkezli tasarım süreçlerinin geliştirilmesinin önemini ve mimari planlamada veriye dayalı yaklaşımların kullanım potansiyelini vurgulamaktadır.

Anahtar Kelimeler: Üretken çekişmeli ağlar, kullanıcı entegrasyonu, plan üretimi.

Scale and Control Methods in Generative Adversarial Networks for Single-Story Houses

Mertcan Güldilek¹ ; Berk Ekici²  Mustafa Emre İlal³ 

^{1,2,3}Izmir Institute of Technology

¹mertcanguldilek@gmail.com; ²emreilal@iyte.edu.tr; ³berkekici@iyte.edu.tr

Abstract

Artificial intelligence (AI) technologies can provide faster and more effective solutions than humans in many fields. In architecture, AI is expected to create more functional and better performing buildings. Generative Adversarial Networks (GAN) are preferred in the literature for generating architectural plan designs with AI. This study developed a model that can learn from existing single-story building plan solution libraries in the literature and generate new plans. U-Net Architecture and Convolutional Message Passing Algorithm (Conv-MPN) were integrated into the GAN model, enabling the model to learn the relationships between rooms better and enhance edge perception. After developing a scaling method to convert pixels to real-world areas for the trained model, two different approaches were investigated to provide user control over generated plans. The first approach is based on restricting the room dimensions while generating the model, while the second method is based based on rejecting generated plans that do not comply with the constraints. While both approaches yielded successful results, the first approach resulted in problems in the relationships between rooms, and the second prolonged the plan generation process. In the developed system, users can control the generation of single-story plan designs by determining the rooms' dimensions and the plan's total square meters. This research emphasizes the importance of developing user-centered design processes and the potential of using data-driven approaches in architectural planning.

Keywords: Generative adversarial networks, user interaction, floor plan generation.

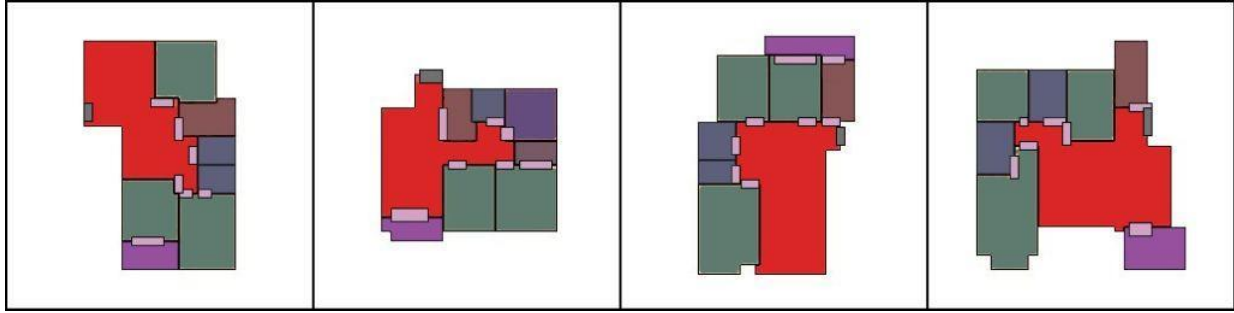
1. Giriş

Yapay zeka teknolojilerinin farklı disiplinlere olan entegrasyonu son birkaç yılda giderek hızlanmış ve pek çok sektörde büyük değişimlere yol açmıştır. Bu teknolojiler, insan gibi düşünüp karar verebilme yetenekleriyle bazı alanlarda insanlardan daha hızlı ve daha etkili çözümler sunma kapasitesine sahiptir. Geleneksel olarak, mimari tasarım süreçleri uzun zaman alır ve karmaşık kararları içerir. Yapay zekanın bu alana entegrasyonu ile tasarım süreçlerini hızlandırmak, proje maliyetlerini azaltmak ve daha işlevsel yapılar ortaya çıkarmak olası görünmektedir (Baduge vd., 2022).

Mimarlık alanında, tek katlı yapı tasarımlarını geliştirmek için Çekişmeli Üretici Ağlar (ÇÜA – “Generative Adversarial Networks - GAN”) kullanan bir dizi araştırma bulunmaktadır. Bu araştırmalar arasında, Aalaei v.d. (2023), Nauata v.d. (2020) ve Chaillou (2020) tarafından yürütülen çalışmalar öne çıkmaktadır. Chaillou (2020), ÇÜA'ların kullanım potansiyelini somut bir şekilde ortaya koyarak, mimari planların adeta fotoğraf çekilir gibi üretilebildiği bir yöntem geliştirdiğini belirtmiş ve bu yöntemi bir web sitesi aracılığıyla kullanıcıların deneyimine sunmuştur. Nauata ve ekibinin 2020'de gerçekleştirdiği çalışma, House-Gan adını taşıyan ve çizgeler ile ÇÜA tabanlı yaklaşımların birleştirildiği bir tasarım sistemini tanıtmıştır. Sistem, kullanıcıların istedikleri mekanlar arasındaki ilişkileri çizge olarak modellemesinden sonra bu ilişkilere sahip planlar üretmektedir. Aalaei ve ekibinin çalışması ise, üç farklı ÇÜA'yı kapsayan bir ekosistem sunmaktadır. Kullanıcılar, tasarım sürecine başlamak için bu ekosisteme balon diyagramları ve alan sınırları gibi girdiler sağlamaktadır. Aalaei ve ekibinin araştırması, üç ÇÜA'nın kullanımı sayesinde elde edilen iş bölümünün avantajlarını ön plana çıkararak, farklı prensipteki ÇÜA'ların kullanımında iş bölümünün önemini vurgulamaktadır. Bu çalışmalar, mimari tasarım süreçlerinde ÇÜA'ların kullanılmasının alternatif tasarım çözümleri sağlama potansiyeline vurgu yapmaktadır.

Tek katlı yapı tasarımlarının üretim sürecine kullanıcıların dahil olmalarının önemi literatürde vurgulanmıştır. Kullanıcı katkısının potansiyelini ve bu katkının sınırlarını keşfetmeye yönelik çalışmalar yürütülmektedir. Kullanıcı merkezli tasarım süreçlerinin geliştirilmesi, mimari tasarım alanında önemli bir araştırma alanı olarak öne çıkmaktadır. Örneğin (Bahrehand vd.,2017), kullanıcı tercihlerine dayalı olarak sezgisel optimizasyon ile üretilen tek katlı yapı planlarının üretimini yönlendirmek için kural tabanlı bir dizi kısıtlayıcı parametre geliştirmiştir. Bu çalışma, mimari tasarım süreçlerinin matematiksel ve kural tabanlı yaklaşımlarla açıklanabileceğini göstermiştir. Ancak, sezgisel optimizasyon algoritmalarının çalışma prensiplerinden farklılık gösteren ÇÜA kullanımı, kullanıcı katılımını daha etkin bir şekilde entegre etmek amacıyla ek metodolojik geliştirmelerin yapılmasını gerektirmektedir.

Bu araştırma, tek katlı binalar için otomatik plan tasarımı üzerine yoğunlaşmaktadır. Yapay zekanın bir alt dalı olan Çekişmeli Üretici Ağlar, gerçeğe yakın görüntüler oluşturabilme yetenekleriyle ön plana çıkmaktadır. Çalışma kapsamında, var olan plan çözümlerinden öğrenip yeni planlar oluşturabilen bir model geliştirilmektedir. Bu model, örneklerinin **Şekil 1** de verildiği RPLAN veri setiyle (Wu vd., 2019) eğitilmiştir. Uygulamalı Evrişimli Sinir Ağları (UESA - Conv-MPN) (Zhang vd., 2021) ve U-Net mimarisi (Ronneberger vd., 2015) kullanarak geliştirilen bu sistem, kullanıcıların girdilerine göre özelleştirilebilen, odaların düzeni ve planın genel yapısını içeren tasarımlar üretebilmektedir.



Şekil 1. Mimarlar tarafından oluşturulmuş planların toplandığı RPLAN veri setinden örnekler (Wu vd., 2019).

Sunulan bu çalışmanın amacı, yapay zeka teknolojileri ile mimari plan çözerken kullanıcıların kendi ihtiyaç ve tercihlerine göre odaların boyutlarını ve düzenlerini özelleştirebilmelerine olanak tanıyarak, kullanıcılara süreç üzerinde daha fazla kontrol sunmaktır. Böylece, yapay zeka kullanımı ile mimarlıkta tasarım süreçleri daha etkin ve verimli olabilir. Bu çaba, yapay zekanın mimarlıkta kullanılabilirliğini ve potansiyelini göstermekle kalmayacak, aynı zamanda kullanıcıları tasarım sürecinde daha fazla alternatif değerlendirmeye de teşvik edecektir.

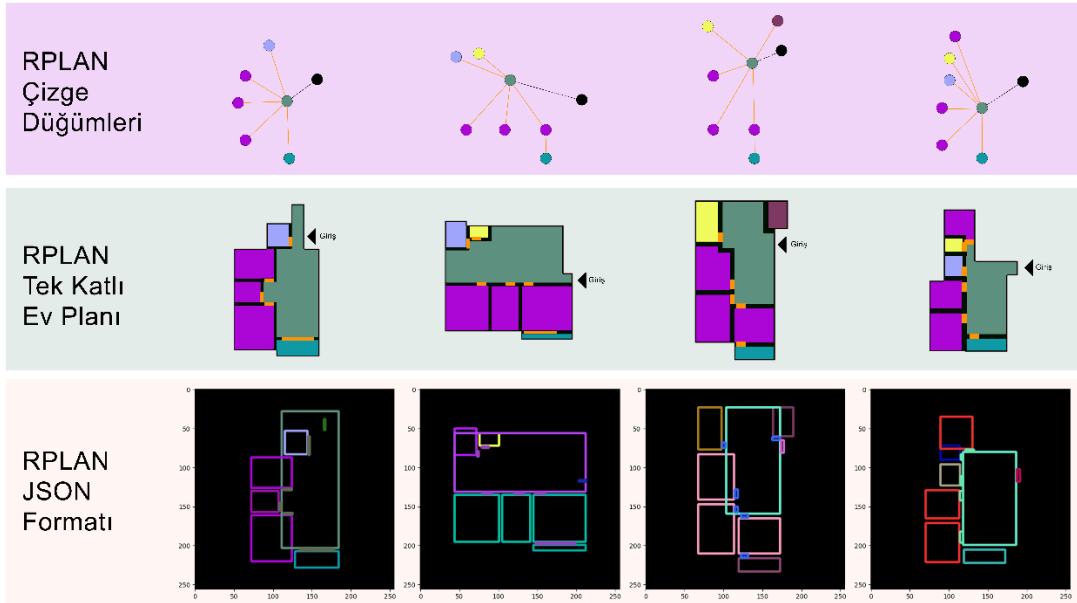
2. Model Oluşumu

Çalışma, var olan plan çözümlerinden öğrenip kullanıcı talep ve tercihleri ile belirlenen çerçevede otomatik olarak tek katlı yapılar için alternatif planlar üretebilen bir yapay zeka modelinin geliştirilmesine odaklanmaktadır. Kullanılan ÇÜA modeli Nauata vd. tarafından geliştirilmiş olan “HouseGAN++” modelidir (Nauata vd., 2021). Kullanılan ÇÜA’nın yapısı olan U-Net mimarisi, bu hedefe ulaşmak için özel olarak oluşturulmuş ve kurgulanmıştır. Çalışma Nauata v.d. (2021) tarafından geliştirilen kodlama ve model üzerine inşa edilmiştir. Model oluşturulduktan sonra kullanıcı etkileşimi için yöntemler araştırılmıştır. Modelin

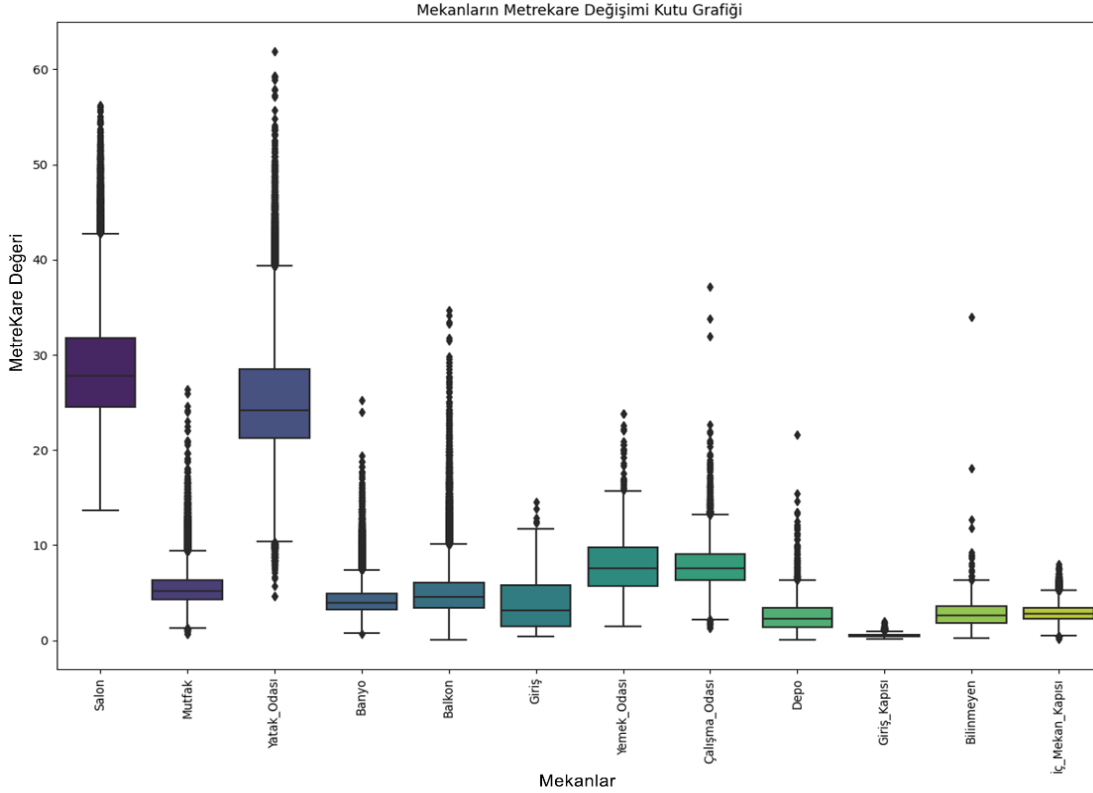
kullanıcı odaklı üretim yapması asıl amaçtır. Bu bölümde modelin geliştirilme süreci ve geliştirilme sürecinde izlenen yöntemler ve tercihler açıklanacaktır.

2.1. Veri Hazırlığı ve Ön İşleme

Üretken ÇÜA'nın oluşturulması için öncelikle bir veri seti gerekir. Bu veri setini kullanan model, her bir tekrarda öğrenerek kendisini problem için eğitir. Kat planı veri setleri incelendiğinde RPLAN (Wu vd., 2019), CubiCasa5k (Kalervo vd., 2019), Rent 3D Dataset (Chenxi Liu vd., 2015) tek katlı plan üretimi için uygun veri setleridir. Bu çalışma kapsamında modelin oluşturulması için tek katlı ev planlarını barındıran ve çeşitli tasarım seçenekleri sunan RPLAN veri seti model eğitimi için tercih edilmiştir. Tanımlanan problem için daha sade olması nedeniyle tercih edilen RPLAN, 80 bin adet tek katlı planı içermektedir. RPLAN veri setindeki PNG görüntüleri, model eğitimi için JSON formatına dönüştürülmüştür. Dönüştürülme süreci, veri setinde bulunan görüntülerdeki her odanın konumunu, boyutunu ve odalar arası ilişkilerin eklenmesini içermektedir (Şekil 2).



Şekil 2. RPLAN veri setinde bulunan verilerin mekansal ilişkiler ve formasyonu.



Şekil 3. ÇÜA modelinin eğitim verisinde bulunan mekânların yüzey alanlarındaki deęişkenlik.

JSON formatındaki planlara örnekler Şekil 2’de görölmektedir. Bu görsellerdeki mekânlar rastgele belirlenen farklı renk kodları kullanarak birbirinden ayrılmıştır. Her bir görselde tekrardan belirlenen bu renkler, aynı renge sahip olan mekânlar için eşlenmiştir. Kapılar, çizilen dikdörtgen boyutunun farklı olmasından dolayı mekânların arasına yerleştirilmiştir (Wu vd., 2019). Bu sayede, mekânların birbirleriyle olan ilişkileri belirlenmiş olur.

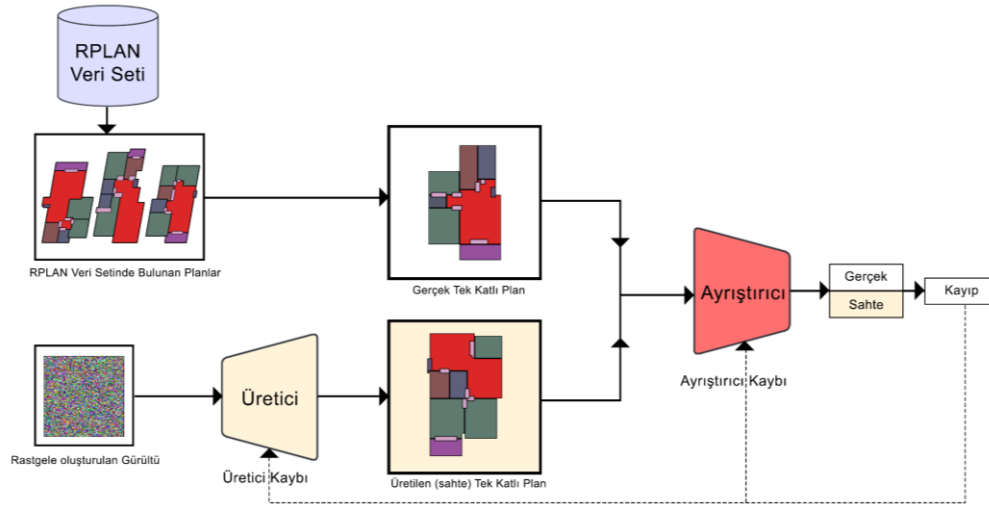
2.2. Modelin Oluşturulması

ÇÜA’lar, gerçekçi plan şemaları üretmeye çalışan bir üretici ve üretilen planları gerçek planlardan ayırt etmeye çalışan bir ayrıştırıcı olmak üzere iki ağıdan oluşmaktadır. Model eğitiminde ayrıştırıcı, üreticinin

oluşturduğu görüntüleri değerlendirerek öğrenme sürecinde birbirlerini geliştirirler. Üretici ve ayrıştırıcı U-Net Mimarisi (Ronneberger vd., 2015) kullanılarak oluşturulmuştur. U-Net mimarisi gereği, girdi olan görüntü boyutu derin öğrenme katmanları sayesinde küçültülerek, girdi görüntülerde bulunan “özellikler” ve “ilişkiler” öğrenilir. U-Net Mimarisine ek olarak, Uygulamalı Evrişimsel Mesaj Aktarımı (UEMA) (Zhang vd., 2021) modele entegre edilmiş ve modelin odalar arası ilişkileri daha iyi öğrenmesi ve kenar algısını güçlendirmesi sağlanmıştır. Ayrıca modelin oluşturulmasında ve geliştirilmesinde kullanılan kütüphaneler sırasıyla Pytorch (Paszke vd., 2019) ve OpenCV (Bradski, G., 2000) kütüphaneleridir.

2.3. Model Eğitimi

Eğitim döngüsü, belirli bir süre boyunca devam eder ve her bir zaman dilimi, veri setinden gelen bilgi yığınlarını işler. Bu bilgileri işlerken, farklı bileşenlere ayrılmaktadır: maskeler, çizge düğümler ve kenarlar. Bu bileşenler, gerçek ve sahte planlar arasındaki ayrımı yapmak için ÇÜA eğitimi için temel doğruluklar oluşturmak üzere kullanılır. Eğitim döngüsü, gerçek ve üretilmiş örnekler arasındaki ayrımı yapmak için ayrıştırıcının parametrelerini ayarlayarak ayrıştırıcının eğitilmesiyle başlar. Bu süreç, eğitimin stabilitesini sağlamak için hataların(kayıp) hesaplanmasını ve ayrıştırıcı hatasının en aza düşürülmesini içermektedir (Şekil 4).

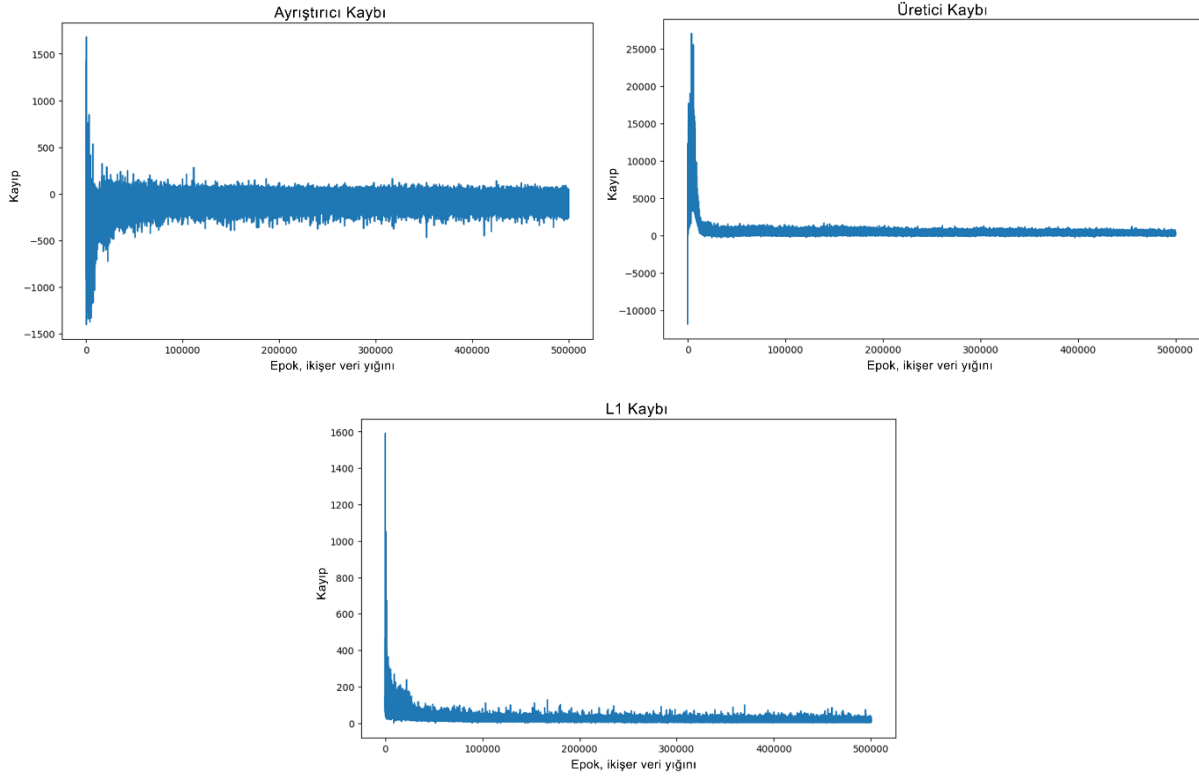


Şekil 4. ÇÜA Modelinin Eğitim Döngü Akış Şeması.

Ayrıştırıcı güncellendikten sonra, üretici eğitilir ve üretici, ayrıştırıcıyı kandıracak kadar gerçekçi plan örnekleri üretmeyi amaçlar. Bu süreçte, hata, ayrıştırıcıyı kandırma yeteneği ve bir mesafe kaybından oluşur. Üretici, belirli bir tekrar sayısından sonra güncellenir ve bu, güncellemelerin sıklığını kontrol eder. Döngü, zaman dilimi, yığın ve hem ayrıştırıcı hem de üreticiyle ilgili hataları belirterek ilerlemeyi kaydeder.

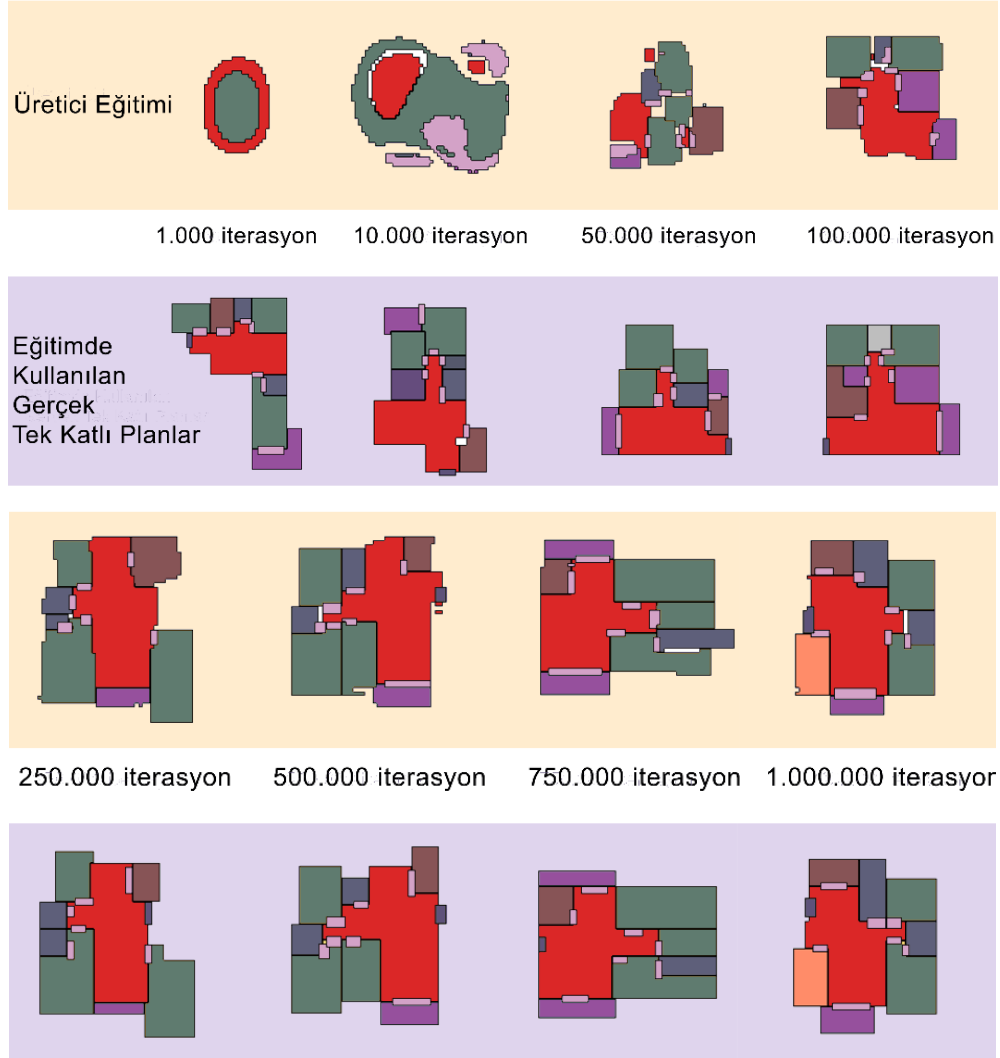
Ayrıca, model kontrol noktaları ve görselleştirmeler belirli aralıklarla kaydedilir ve bu, modelin performansını değerlendirme ve eğitimi boyunca ilerlemesini takip etme olanağı sağlar. Tüm zaman dilimleri tamamlanana kadar süreç devam eder ve bu da eğitim döngüsünün sonunu işaret eder. Eğitim döngüsü boyunca üretici ve ayrıştırıcı arasında üç adet hata takip edilir: ayrıştırıcı kaybı, üretici kaybı ve L1 (Mutlak Sapma) kaybı. Ayrıştırıcı kaybı, ayrıştırıcının sahte verileri doğru bir şekilde tespit edebilme yeteneğini ölçer.

Üretici kaybı, üreticinin sahte tek katlı planların ne kadar gerçekçi bir şekilde üretebildiğini gösterir. Mutlak sapma ise, üretici tarafından üretilen planların gerçek planlara olan farkını ölçmek için kullanılır ve üretilen planların gerçek planlara olan benzerliğini değerlendirir. **Şekil 5'** e göre ayrıştırıcı, üretici ve L1 hataları başlarda çok fazla olurken, ÇÜA modeli eğitiminde tekrar arttıkça hatalar azalmaktadır.



Şekil 5. ÇÜA Modelinin Eğitimde tekrara bağlı değişen hataları.

Eğitim sırasında model, çeşitli plan organizasyonları oluşturur. ÇÜA'nın eğitimi sırasında üretilen tek katlı plan organizasyonlarının gelişimi tekrar sayısı ilerledikçe artmaktadır (Şekil 6). Eğitilen modelin, ayrıştırıcı, üretici ve L1 hataları azaldıkça üretilen tek katlı planların mantıklı olduğu gözlenmektedir. Üretilen planların, RPLAN veri setinde bulunan örneklere görsel olarak benzedikleri söylenebilmektedir.

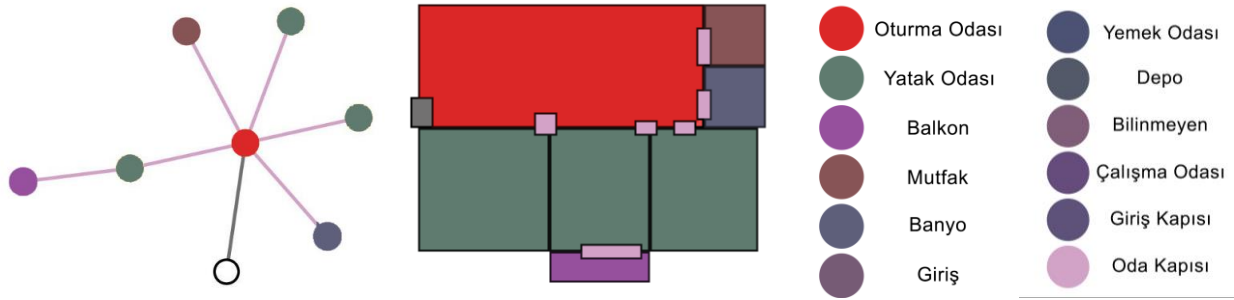


Şekil 6. Eğitim döngüsü boyunca gelişen ÇÜA'nın her tekrarda ürettiği plan şemaları.

3. Plan Üretimi, Ölçeklendirme ve Kullanıcı Etkileşim Yöntemleri

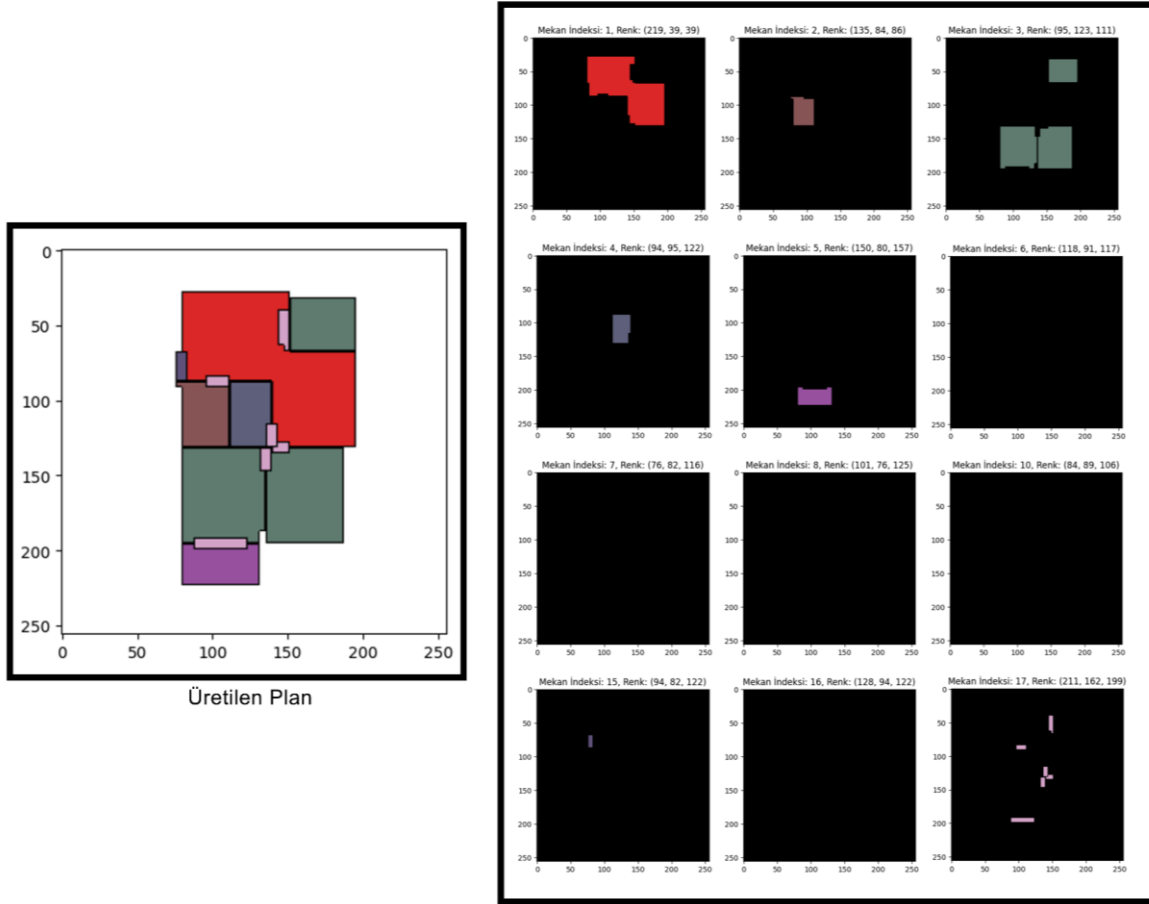
3.1. Plan Üretim Yöntemi

Üretici model, kullanıcı tarafından sağlanan JSON formatındaki bilgileri kullanarak çalışmaktadır. Bu sayede üretici model, eğitildiği şekilde davranarak yeni ürünler ortaya koymaktadır. ÇÜA modeline girdi olarak kullanıcılar, odaları temsil eden çizge düğümleri ve kapıları temsil eden kenarlardan oluşan bir çizge oluştururlar. Üretilen tek katlı planın organizasyonu, çizgeyi yansıtmaktadır (**Şekil 7**). Bu sayede, oluşan tek katlı plan organizasyonu, üretimi, kullanıcının aradığı mekan ilişkileri doğrultusunda gelişmektedir.



Şekil 7. Sol- Oluşturulan Çizge düğümleri (Balon Diyagram), Orta- Üretici tarafından üretilen tek katlı ev planı.

Üretim sürecinin başarıyla tamamlanmasının ardından, kullanıcı etkileşimlerini iyileştirmeyi amaçlayan tasarım aşamasına entegre edilmek üzere, çeşitli dijital yöntemler tasarlanmıştır. İlk adım olarak, üretim sürecinde kullanılan ve farklı mekanları temsil etmek için renk kodları ataması yapılmıştır. Bu renk kodlarının her biri, farklı bir mekanı ifade etmektedir (**Şekil 8**).

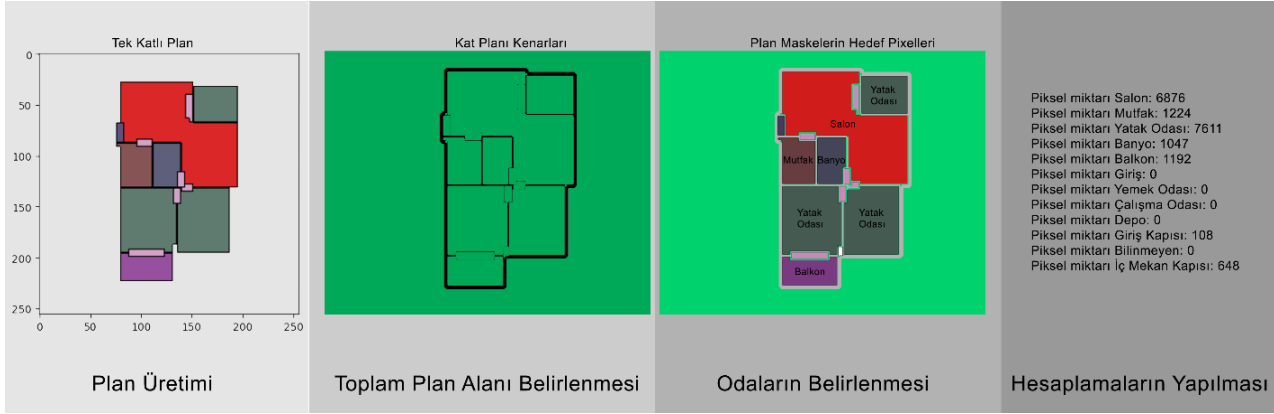


Şekil 8. Sol görsel- Üretilen tek katlı plan, Sağ görsel- mekanlar ve plandaki piksel dağılımları.

3.2. Ölçeklendirme Katsayısı (ÖK)

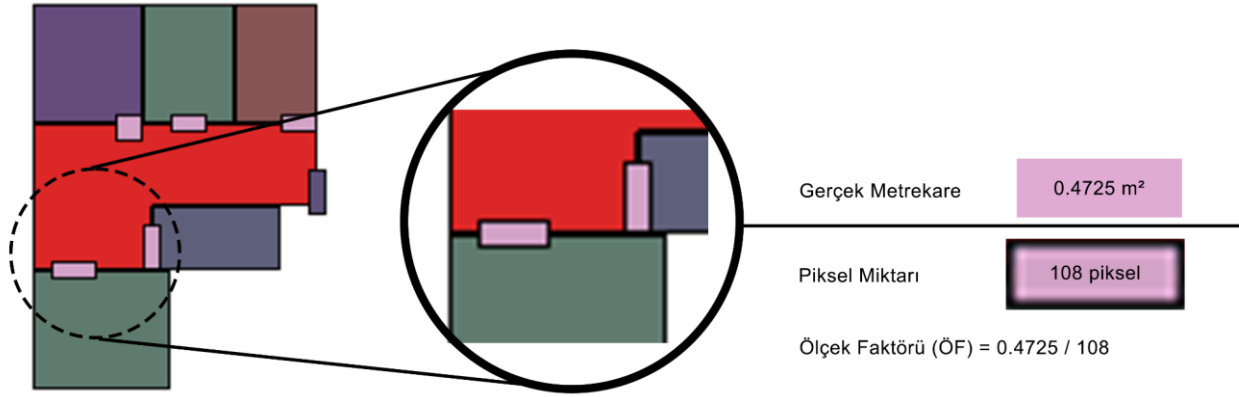
Planların gerçek dünya ölçülerine uygun olarak ölçeklendirilmesi için piksellere boyut tanımlamak gerekmektedir. Geliştirilen ölçeklendirme yöntemi, kullanıcılara plan üzerinde daha fazla kontrol sağlamaktadır. Kullanıcılar, odaların boyutlarını ayarlayarak kendi ihtiyaç ve tercihlerine uygun planları oluşturabilirler. Üretilen tek katlı planları girdi olarak alan bir yöntem geliştirilmiştir (Şekil 9). Önce planda beyaz arka fon çıkarılarak dış duvarlar belirlenir. İkinci aşamada, odaların belirlenmesiyle piksel dağılımı ve

büyüklikleri toplanır. İleride analizlerde kullanılmak üzere kaydedilir. Üçüncü ve son aşamada her bir oda için kaydedilen piksel değerlerinin bir raporu kullanıcıya sunulur.



Şekil 9. Geliştirilen ölçeklendirme yöntemi öncesi veri toplama aşamaları.

ÇÜA'lar, tek katlı plan organizasyonlarını piksel tabanlı olarak üretir. Bu nedenle, piksel tabanlı (raster) formatın işlenmesi gerekmektedir. Bir dış kapının piksel sayısını mimari ölçütlerle ilişkilendirerek, gerçek dünya boyutlarına çevirebilmek için bir ölçeklendirme katsayısı belirlenir (Şekil 10). Bu katsayı, planların ölçülmesi ve ölçümlerin gerçek dünya alan değerlerine çevrilmesi için kullanılmaktadır. Üretilmiş planlarda, sadece kapı ve duvar gibi tasarım elemanlarının piksel miktarına bakıldığı takdirde gerçek dünya hesaplarına ölçeklenebilmektedir. Bu çalışmada, kapılar referans olarak kullanılmıştır.



Şekil 10. Ölçeklendirme Katsayısı Oluşturulması.

Kapıların gerçekteki standart ölçüleri düşünüldüğünde 90 cm x 5.25cm olarak referans alınmıştır. Planlarda bulunan piksel miktarı düşünüldüğünde 0.475 metrekareye karşılık gelmektedir. Kapı alanı, 108 piksel olarak ölçülmüştür. Bu referans ile ölçeklendirme katsayısı (ÖK) belirlenmiştir. ÖK sayesinde üretilen planların oda başına denk gelen piksel sayısını metrekare birimine çevirmek mümkün olmuştur.

ÖK kullanılarak eğitim veri setindeki planların metrekare araştırılması yapılmıştır (**Tablo 1**). Bu sayede, ÇÜA'ların öğrendiği verilerin metrekare değerleri ortaya çıkmaktadır. Amaç, ÇÜA'nın hangi değer aralıklarında plan üretebildiğini göstermektir. ÇÜA'nın eğitimde öğrendiği metrekare değerleri mekan bazlı ve adet bazlı olduğundan ÇÜA'ya girdi verildiğinde kendisi nasıl tasarım yapacağına karar vermesi gerekmektedir. Bazı mekanların metrekare değerlerinin fazla olmasının sebebi ise birden fazla bulunmasıdır. Her planda bulunan mekanlar aynı sayıda ve aynı boyutta olmadığı için aralıklar belirlenmiştir.

Tablo 1. RPLAN eğitim veri setinde bulunan planların metrekare değerleri ve sayısı.

| Mekan İsimleri | Metrekare Değeri | # adet bulunur |
|-----------------|-----------------------|----------------|
| Salon | 13-56 m ² | 1 |
| Mutfak | 4-26 m ² | 0-1 |
| Yatak Odaları | 8-61 m ² | 0-3 |
| Banyo | 1.5-25 m ² | 1-2 |
| Balkon | 2-34 m ² | 1-2 |
| Giriş Alanı | 0-14 m ² | 0-1 |
| Yemek Odası | 0-23 m ² | 0-1 |
| Çalışma Odası | 0-37 m ² | 0-2 |
| Depo | 0-21 m ² | 0-1 |
| Giriş Kapısı | 0.15-2 m ² | 1 |
| Bilinmeyen | 0-33 m ² | 0-1 |
| İç Mekan Kapısı | 0.15-7 m ² | 1-9 |

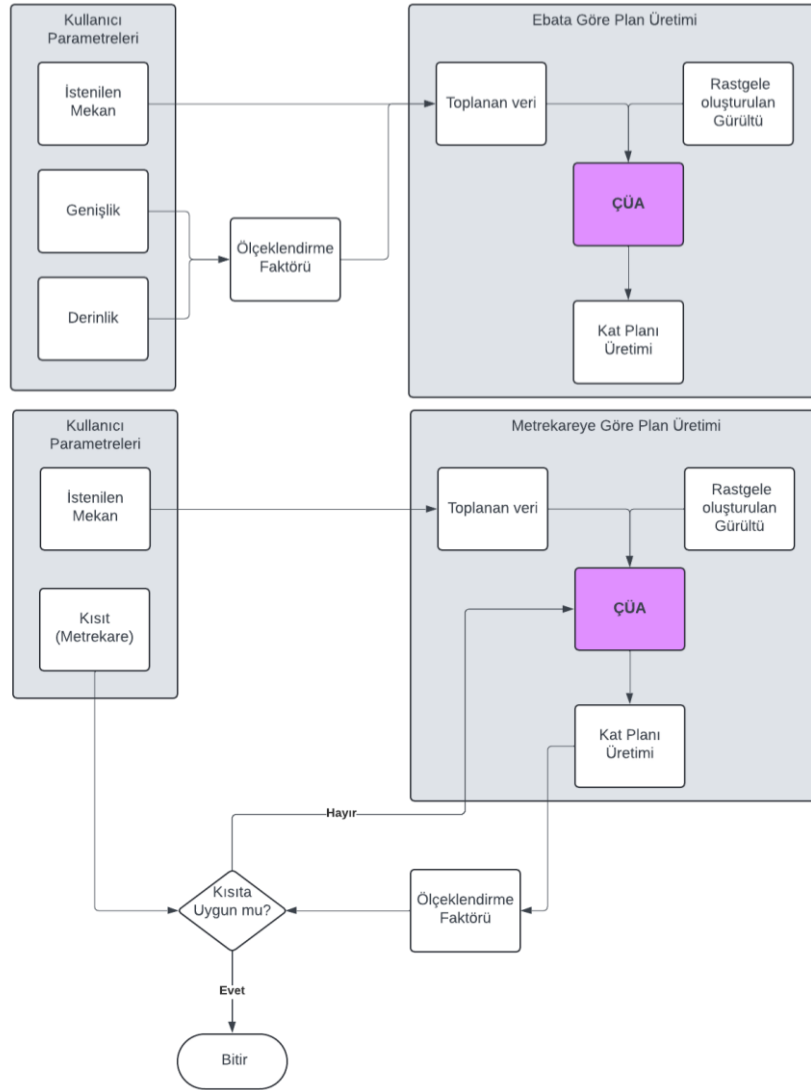
Mekanların metrekare değişim aralıklarına bakıldığında en büyük farkın salon ve yatak odasında olduğu görülmüştür. Örnek olarak salon, tablo 1'e göre veri setindeki her örnekte bulunduğu ve metrekaresi oldukça değişken olduğu için en büyük fark bundan kaynaklanmaktadır.

3.3. Kullanıcı Etkileşim Yöntemleri

ÇÜA kat planını ürettikten sonra ÖK yöntemi kullanılarak kat planının kenarları belirlenir ve devamında planlar renklere ayrılarak odaların metrekareleri hesaplanır. Kullanıcı tercihleri ile üretilen odaların metrekareleri arasında bir kıyaslama yapılır, eğer kullanıcı tercihleri sağlanıyorsa ise üretilen plan başarılı bulunur. Eğer üretilen plan başarılı bulunmazsa ÇÜA yeni bir plan üretir.

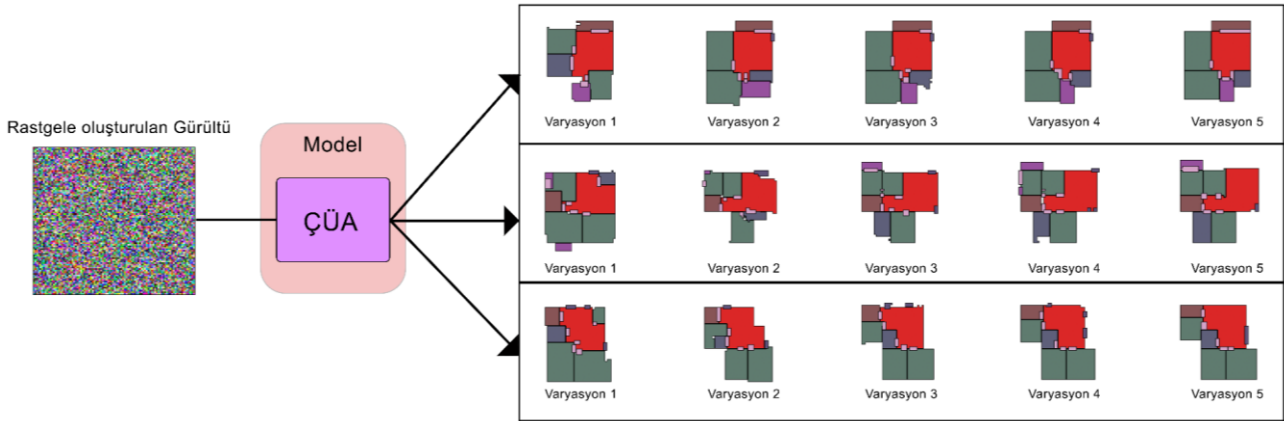
ÖK yöntemi ile ÇÜA'lar tarafından üretilen tek katlı planların gerçek dünya ölçülerine uyarlanması tamamlandıktan sonra kullanıcının istekleri doğrultusunda bir planı oluşturabilmek için iki adet yöntem sunulmuştur. Bu yöntemlerden ilkinde kullanıcı parametreleri arasında; istenilen mekanlar, mekan genişliği

ve uzunluğu bulunmaktadır. Kullanıcı istediği odanın boyutlarını belirler ve yöntem, girilen girdiyi yerleştirerek dikdörtgen çizer ve alan tanımlar (Şekil 11-Üst görsel).



Şekil 11. Üst görsel - Boyutlara göre Plan Üretim Yöntemi, Alt görsel- Metrekareye göre Plan Üretim Yöntemi.

İkinci yöntem, metrekareye göre plan üretim yöntemi olarak adlandırılmaktadır. Kullanıcı, istediği mekanlara uzunluk ve genişlik girmek yerine, sabit bir metrekare değeri tanımlayabilir (**Şekil 11**- Alt görsel) . Bu sayede, eğitilen ÇÜA, mekanları eğitim setinde var olan ilişkileri koruyarak, birbirinden ayrı konumlandırırken metrekare kısıtına uygun planlar oluşturur. İkinci yöntem, RPLAN veri setinin test için ayrılmış planlarının koordinatlarını kullanarak istenilen planı oluşturmaya çalışır. Yerleştirilen mekanların genişlikleri ve ilişkileri, her bir tekrarda sıfırdan oluşan rastgele gürültüler sayesinde planların metrekare boyutlarında değişiklik sağlar (**Şekil 12**). Her bir plan durumunun çeşitli varyasyonları sağlandıktan sonra, kullanıcı tarafından atanan metrekare kısıt parametresi ile oluşturulan varyasyonların metrekare büyüklükleri arasında bir kıyaslama yapılır. Eğer ÇÜA tarafından üretilen tek katlı planlar, hedef metrekare kısıtına göre belirlenen en fazla sapma değerinden küçükse, bu plan başarılı kabul edilir.

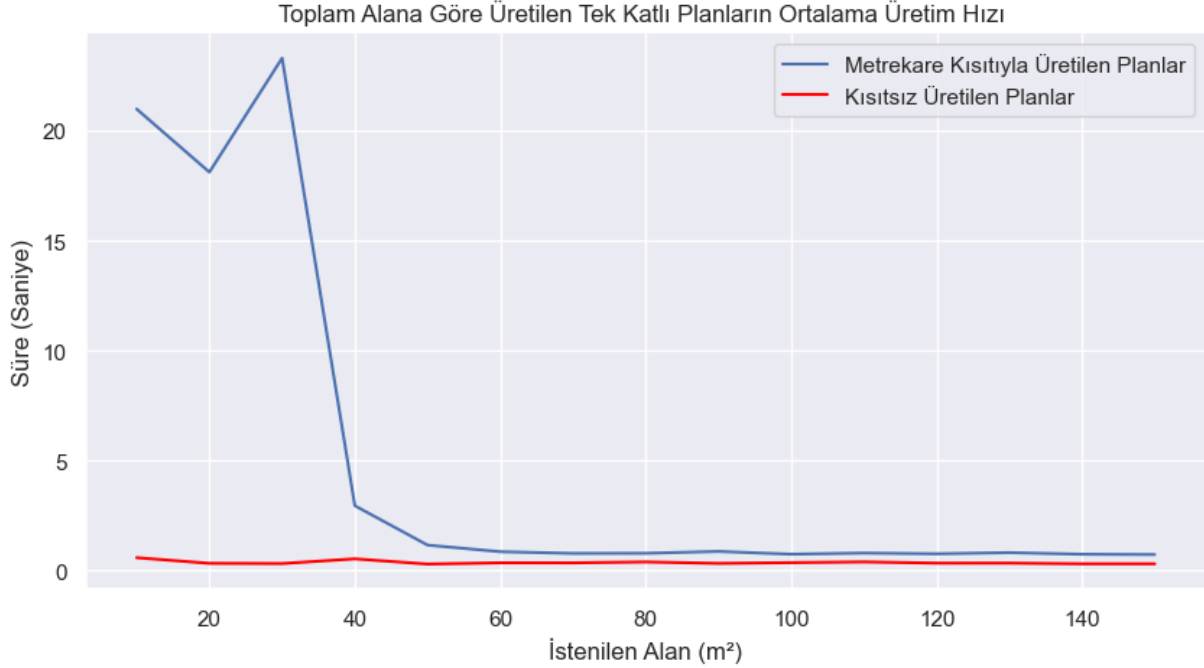


Şekil 12. Rastgele oluşturulan gürültünün aynı planlara olan farklılaştırma etkisi.

Her iki yöntemin de artıları ve eksileri bulunmaktadır. İlk yöntemde, kullanıcılara boyutlar konusunda daha fazla özgürlük sağlanabilirken (genişlik, derinlik), ÇÜA'nın öğrendiği görünmez ilişkileri üretim aşamasında kullanamaması söz konusudur. Bu, ÇÜA'nın beklenen ilişkilerin yaratılmamasına neden olabilmektedir (**Şekil 14**).

İkinci yöntemde ise, ÇÜA'nın eğitimde öğrendiği ilişkiler korunmaktadır. Ancak, istenilen metrekare aralığını elde edebilmek için oldukça fazla tekrar yapmak gerekebilmektedir, bu da süreci uzatabilmektedir (**Şekil 13**).

Bunun nedeni, algoritmanın başlangıç noktası olarak kullandığı gürültünün rastgele olmasıdır. Ayrıca, kullanıcıların metrekare değerlerini oldukça büyük veya küçük sayılar vererek ÇÜA'yı kontrol etmesi mümkün değildir. Örneğin, balkon için 100 metrekare vermek, ÇÜA'yı sınırsız döngüye sokabilmektedir çünkü ÇÜA'nın eğitim gördüğü RPLAN veri setinde daha önce buna benzer bir örnek bulunmamaktadır (**Tablo 1**).



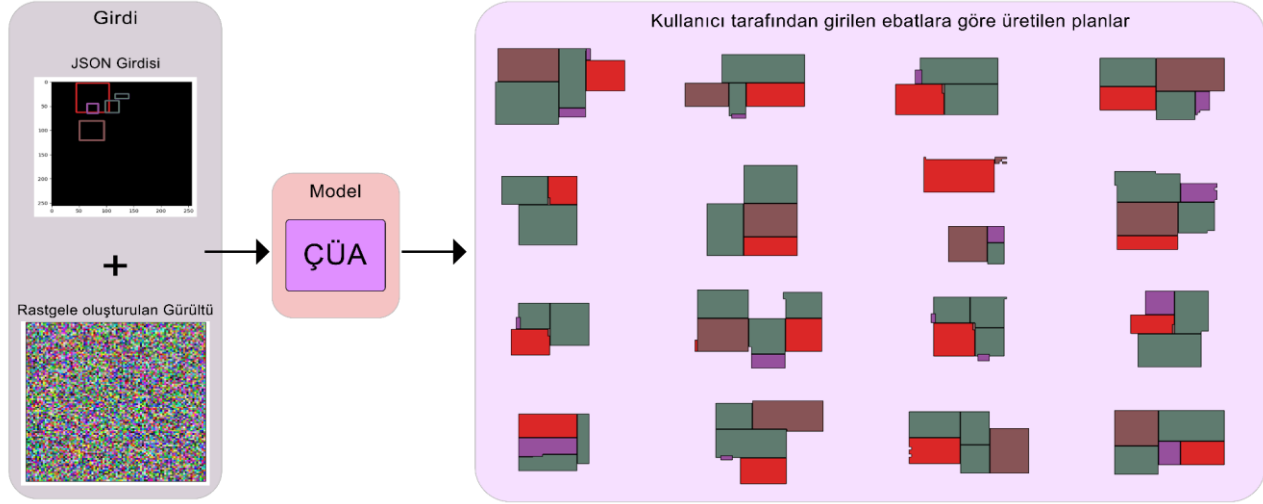
Şekil 13. En fazla toplam alan kısıtı ile üretilen tek katlı planların ortalama üretim hızı.

4. Bulgular

4.1. Boyutlara göre Üretilen Tek Katlı Planlar

Boyutlara göre üretilen plan şemalarına bakıldığında, mekanların birbirleri ile olan ilişkileri zayıf kaldığı görülmüştür (**Şekil 14**). Bazı mekanların diğerlerinden ayrık olduğu planlar üretilmektedir. Ayrıca, mekanların büyüklükleri dengeli bir hiyerarşiyi takip etmemektedir. Fakat kullanıcı, istediği mekanları istediği boyutlara yakın olacak şekilde üretebilmiştir.

Bu da kullanıcıya belirli bir özgürlük kazandırarak istediğı plan şemasını temsili olarak elde edebilmesini sağlamıştır. JSON girdisinin tıpatıp aynısını üretmek mümkün değildir çünkü rastgele oluşturulan gürültü her üretimde değişmektedir.

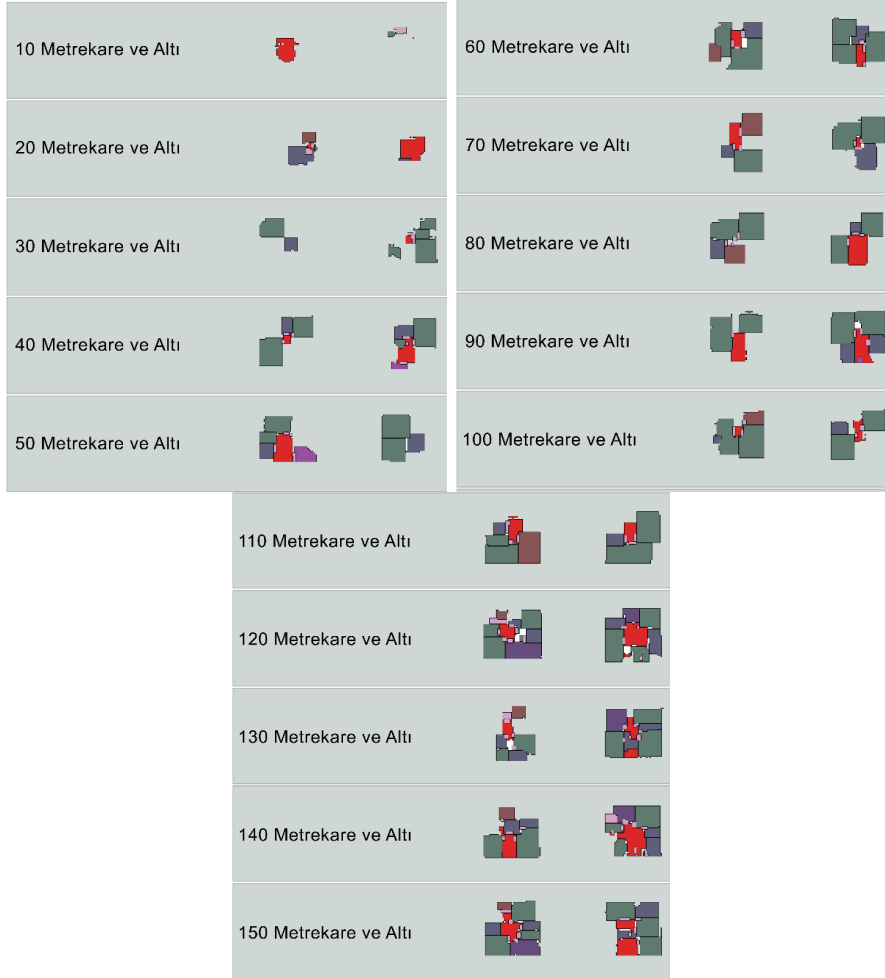


Şekil 14. Boyutlara göre üretilen Plan Şemaları.

4.2. Toplam alana göre Üretilen Tek Katlı Planlar

Toplam alan kısıtlanarak üretilen plan şemaları girilen metrekare değerine göre oldukça çeşitlilik göstermektedir (Şekil 15).

- 10 metrekare ve altı, bir mekanı bile tam anlamıyla tanımlayamamaktadır.
- 20 metrekare ve altında bazı mekanların tamamen tanımlı oldukları görülmektedir.
- 30 metrekareden itibaren birden fazla mekanın tanımlanabildiği görülmektedir.
- İlk plan şemaları 50 metrekare ve altından itibaren başlamaktadır çünkü 2 veya 3 mekanı düzgünce aynı anda barındırabilirler.
- 120 metrekare ve altından itibaren, binaların tek katlı plan organizasyon şemaları ortaya çıkmaktadır.
- 150 metrekarede bütün bir plan oluşturulmuştur.



Şekil 15. Toplam alana göre üretilen Plan Şemaları.

5. Sonuç

Bu araştırmada RPLAN veri seti kullanılarak tek katlı yapı planları üreten bir ÇÜA modeli eğitilmiştir. Bu model ile üretilen kat planlarının boyutlandırılabilmesi için bir ölçeklendirme katsayısı belirleme yöntemi geliştirilmiştir. Boyutları belirlenen kat planlarının kullanıcı taleplerini karşılayabilmesi için ise iki farklı

yaklaşım önerilmiştir. İlk yaklaşım modelin yeni planları boyutları kısıtlanmış odalar kullanarak üretmesine dayanırken, ikinci yaklaşım üretilen planların kısıtlara uymayanlarının elenmesine dayanmaktadır. Bulgulara bakıldığında, boyutlara göre üretilen plan yöntemi ile tek katlı planlar istenilen boyutlarda üretebilmiştir. Ancak üretilen planlarda mekanların birbirleri ile olan ilişkilerinde eğitim için kullanılmış olan veri setinde bulunmayan ilişkiler gözlemlenmiştir. Metrekareye göre üretilen plan yönteminde ise yine istenilen etki sağlanmış ve metrekare bazında kullanıcının talep ettiği boyutlarda planlar elde edilebilmiştir. Öte yandan istenilen metrekare aralıklarında bir planın üretimi görece uzun sürmektedir. Mimarların, yapay zeka tabanlı sistemlerden etkin faydalanabilmeleri için beklentilerini hızlı ve doğru tarif edebilmelerini sağlayan yeni etkileşim yöntemleri gereklidir. Geliştirilecek olan bu etkileşim yöntemleri mimarların üretken sistemleri daha etkin yönlendirebilmelerinin önünü açacaktır.

KAYNAKLAR

- Aalaei, M., Saadi, M., Rahbar, M., & Ekhlasi, A. (2023). Architectural layout generation using a graph-constrained conditional Generative Adversarial Network (GAN). *Automation in Construction*, 155, 105053. <https://doi.org/10.1016/j.autcon.2023.105053>
- Baduge, S. K., Thilakarathna, S., Perera, J. S., Arashpour, M., Sharafi, P., Teodosio, B., Shringi, A., & Mendis, P. (2022). Artificial intelligence and smart vision for building and construction 4.0: Machine and deep learning methods and applications. *Automation in Construction*, 141, 104440. <https://doi.org/10.1016/j.autcon.2022.104440>
- Bahrehand, A., Batard, T., Marques, R., Evans, A., & Blat, J. (2017). Optimizing layout using spatial quality metrics and user preferences. *Graphical Models*, 93, 25–38. <https://doi.org/10.1016/j.gmod.2017.08.003>
- Chaillou, S. (2020). ArchiGAN: Artificial Intelligence x Architecture. In P. F. Yuan, M. Xie, N. Leach, J. Yao, & X. Wang (Eds.), *Architectural Intelligence* (pp. 117–127). Springer Nature Singapore. https://doi.org/10.1007/978-981-15-6568-7_8
- Nauata, N., Chang, K.-H., Cheng, C.-Y., Mori, G., & Furukawa, Y. (2020). House-GAN: Relational Generative Adversarial Networks for Graph-Constrained House Layout Generation. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), *Computer Vision – ECCV 2020* (Vol. 12346, pp. 162–177). Springer International Publishing. https://doi.org/10.1007/978-3-030-58452-8_10
- Nauata, N., Hosseini, S., Chang, K.-H., Chu, H., Cheng, C.-Y., & Furukawa, Y. (2021). House-GAN++: Generative Adversarial Layout Refinement Network towards Intelligent Computational Agent for Professional Architects. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13627–13636. <https://doi.org/10.1109/CVPR46437.2021.01342>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation* (arXiv:1505.04597). arXiv. <http://arxiv.org/abs/1505.04597>
- Wu, W., Fu, X.-M., Tang, R., Wang, Y., Qi, Y.-H., & Liu, L. (2019). Data-driven interior plan generation for residential buildings. *ACM Transactions on Graphics*, 38(6), 1–12. <https://doi.org/10.1145/3355089.3356556>

- Zhang, F., Nauata, N., & Furukawa, Y. (2021). *Conv-MPN: Convolutional Message Passing Neural Network for Structured Outdoor Architecture Reconstruction* (arXiv:1912.01756). arXiv. <http://arxiv.org/abs/1912.01756>
- Kalervo, A., Ylioinas, J., Häikiö, M., Karhu, A., & Kannala, J. (2019). CubiCasa5K: A Dataset and an Improved Multi-task Model for Floorplan Image Analysis. In M. Felsberg, P.-E. Forssén, I.-M. Sintorn, & J. Unger (Eds.), *Image Analysis* (Vol. 11482, pp. 28–40). Springer International Publishing. https://doi.org/10.1007/978-3-030-20205-7_3
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library (arXiv:1912.01703). arXiv. <http://arxiv.org/abs/1912.01703>
- Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.