

TÜRKHAVACILIK UZAYSANAYİİ

Türk Havacılık ve Uzay Sanayii TSGV'nin Bağlı Ortaklığı ve SSB'nin iştirakidir.

MAKİNE ÖĞRENMESİ KULLANILARAK YAPISAL ANALİZ SÜRELERİNİN İYİLEŞTİRİLMESİ

11 Temmuz 2025

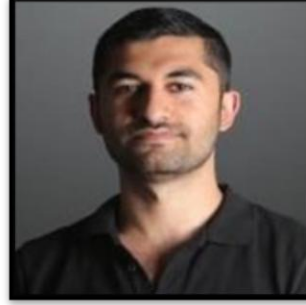
LIFT UP Lisans Odaklı Bitirme Projeleri Programı,



Proje Ekibi Tanıtımı



Doç. Dr. Ali Arı
Akademik Danışman



Alihan Cambaz
Sanayi Danışmanı



Osman Çağlar Baysallı
Sanayi Danışmanı



Barış Özalp
Ekip Üyesi



Elif Kartal
Ekip Üyesi



Fatmanur Ceylan
Ekip Üyesi



Hilal Su Korkmaz
Ekip Üyesi



Problem Tanımı

Problem Tanımı:

Geleneksel yapısal analiz araçlarının kullanımı, çeşitli dezavantajlar ve kısıtlamalarla birlikte önemli zorluklar barındırmaktadır.

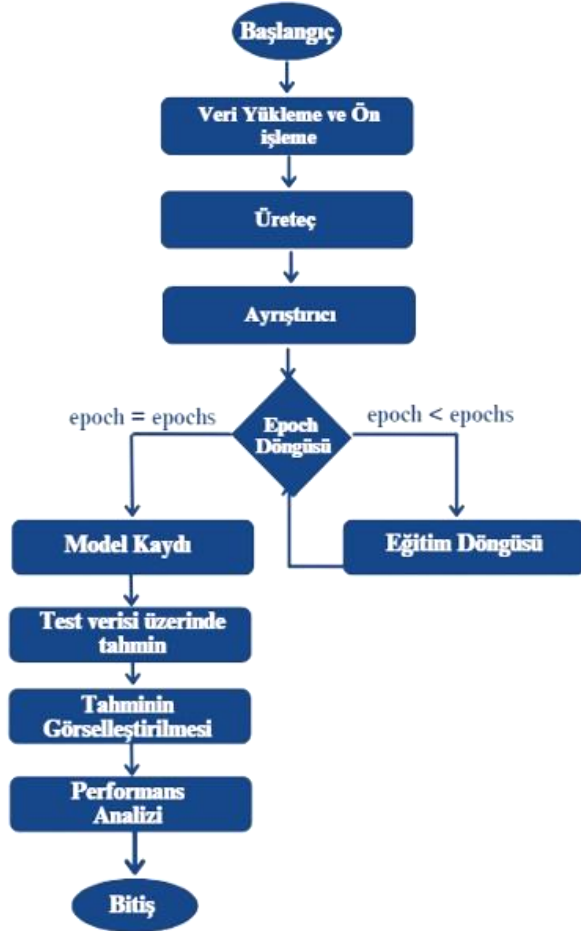
- Yüksek Lisanslama Maliyeti
- Uzun Analiz Süresi
- Teknik Karmaşıklık
- Uyumluluk Sorunları
- Sürdürülebilirlik İhtiyacı

Projenin Hedefi:

Bu projede, mevcut yapısal analiz araçlarının dezavantajlarını gidermeye yönelik derin öğrenmeye dayalı bir yöntem olan **Koşullu Üretken Karşıt Ağlar (Conditional Generative Adversarial Network cGAN)** modeli önerilmiştir ve aşağıdaki hedefler belirlenmiştir:

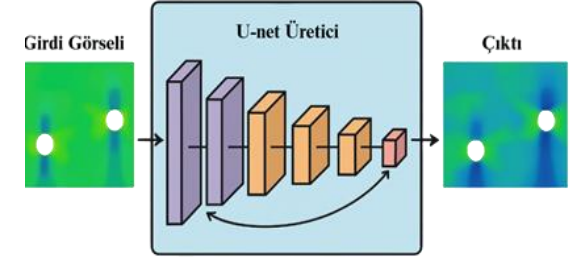
- Lisanslama Maliyetlerinin Azaltılması
- Analiz Sürelerini Kısaltmak
- Kullanım Kolaylığı Sağlamak
- Uyumluluğu Artırmak
- Sürdürülebilir Tasarımlar İçin Destek Sağlamak

Gerçekleştirilen Araştırma Faaliyetleri

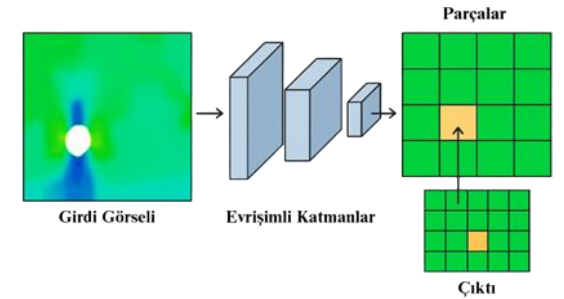


- Modelin tüm boyutlara uygun sonuçlar üretebilmesi için görsel veriler 512x512 piksel boyutlarına yeniden boyutlandırıldı. Görüntü kalitesinin korunması amacıyla padding yöntemi kullanıldı.
- **U-Net** tabanlı bir üretici (generator) modeli kullanıldı.
 - Kodlayıcı (Encoder) katmanlar
 - Kod çözücü (Decoder) katmanlar
 - Atlama bağlantıları (Skip connections)
- **PatchGAN** tabanlı bir ayrıştırıcı (discriminator) modeli kullanıldı.
- Model, 300 epoch ve 1 batch boyutu ile eğitilerek kaydedildi.
- Kaydedilen model ile tahmin görselleri oluşturuldu.
- Elde edilen sonuçlar, çeşitli performans metrikleriyle değerlendirildi.

Üreteç Mimarisi



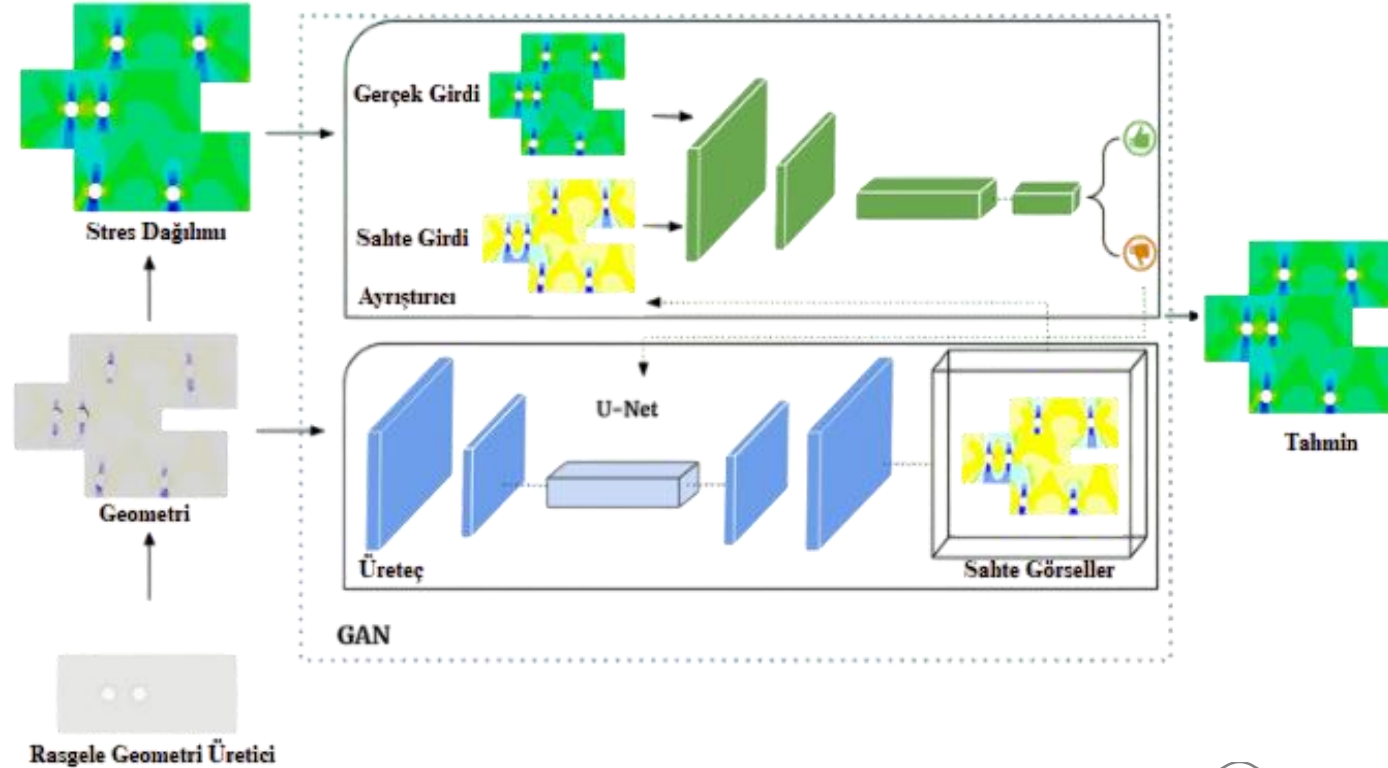
Ayrıştırıcı Mimarisi



Gerçekleştirilen Araştırma Faaliyetleri

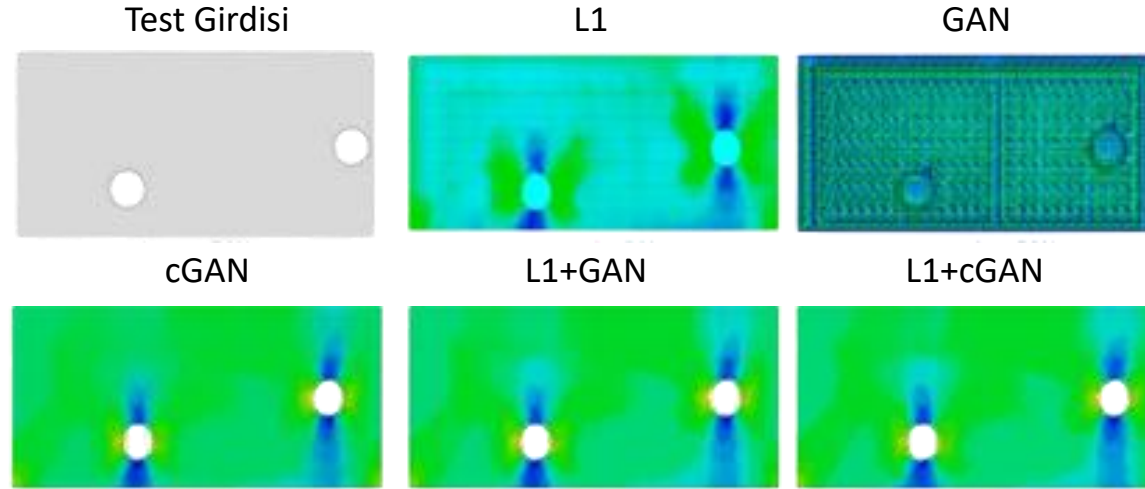
U-Net Üreteç ve PatchGAN Ayırıştırıcısının Birlikte Çalışma Prensipleri

- Üreteç girdi verilerden yola çıkarak yeni bir görsel oluşturur(sahte görsel)
 - Encoder blokları, giriş verisinden anlamlı ve sıkıştırılmış özellikleri çıkarırken; decoder blokları, bu sıkıştırılmış özellikleri kullanarak orijinal boyutlarda ve anlamlı bir çıktı üretir.
- Ayırıştırıcıya hem üreteç tarafından oluşturulan yapay veri hem de gerçek veri gönderilir. Ayırıştırıcı, gelen verinin gerçek mi yoksa üretilmiş mi olduğunu tahmin etmeye çalışırken, PatchGAN yaklaşımıyla **her bir alt bölgeyi (patch) ayrı ayrı değerlendirerek yerel detayların doğruluğunu ölçer.**



Gerçekleştirilen Araştırma Faaliyetleri

Kayıp Fonksiyonları ve Seçim Süreci



Üreteç'in Ayrıştırıcıyı kandırarak gerçekçi görüntüler üretmesini teşvik eder. **L1+cGAN kaybı**, üretilen ve hedef görüntüler arasındaki ortalama mutlak farkı ölçerek yapısal benzerliği ve piksel düzeyinde doğruluğu artırır.



Araştırma Sonuçları ve Çıktılar

- Model, Google Colab üzerinde **A100 GPU** kullanılarak, 300 epoch, 1 batch boyutu ve 10 görsel veri çifti ile yaklaşık **30 dakikada** eğitildi.
- Model kaydedildikten sonra, her bir girdi verisi için **tahmin sonuçları yaklaşık 3 saniyede** üretildi.

```
[30] loss_function = tf.keras.losses.BinaryCrossentropy(from_logits=True)

def generator_loss(disc_fake_output, fake_output, target):
    gan_loss = loss_function(tf.ones_like(disc_fake_output), disc_fake_output)
    t1_loss = tf.reduce_mean(tf.abs(target - fake_output))
    return gan_loss + (50 * t1_loss)

def discriminator_loss(real_output, fake_output):
    real_loss = loss_function(tf.ones_like(real_output), real_output)
    fake_loss = loss_function(tf.zeros_like(fake_output), fake_output)
    return real_loss + fake_loss

generator_optimizer = Adam(2e-4, beta_1=0.5)
discriminator_optimizer = Adam(2e-4, beta_1=0.5)

[31] for i, (input_image, target_image) in enumerate(dataset.take(5)):
    print(f"Input {i}: {input_image.shape}, Target {i}: {target_image.shape}")

Input 0: [1, 512, 512, 3], Target 0: [1, 512, 512, 3]
Input 1: [1, 512, 512, 3], Target 1: [1, 512, 512, 3]
Input 2: [1, 512, 512, 3], Target 2: [1, 512, 512, 3]
Input 3: [1, 512, 512, 3], Target 3: [1, 512, 512, 3]
Input 4: [1, 512, 512, 3], Target 4: [1, 512, 512, 3]

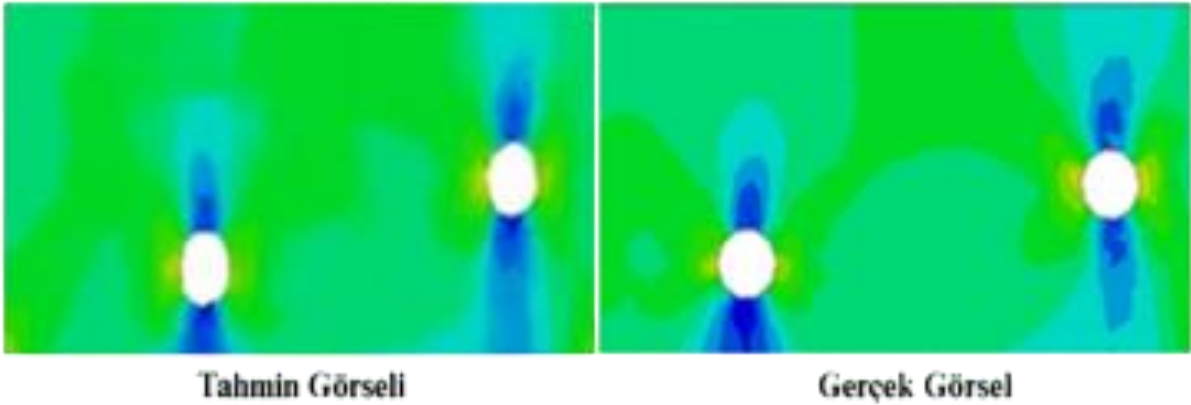
generator = tf.keras.models.load_model('content/aircrafts/030616e/generator.h5')
new_input = load_and_preprocess('content/aircrafts/030616e/airplane_11_input.jpg')
new_input = tf.expand_dims(new_input, axis=0)
generated_output = generator(new_input, training=False)

plt.imshow((generated_output[0] - 1) * 127.5).astype(np.uint8))
plt.axis('off')
plt.title('Model tarafından üretilen görsel')
plt.show()
```

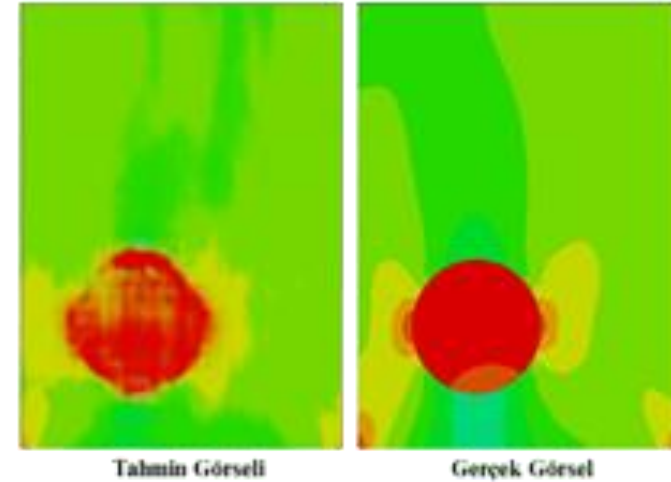

Araştırma Sonuçları ve Çıktılar

Geliştirilen modelin iki farklı veri seti üzerinde elde ettiği sonuçlar aşağıdaki gibidir. Veri setleri, geometrik şekil ve renk açısından farklılık göstererek modelin çeşitlilik karşısındaki başarısını test etmemize olanak sağlamaktadır.

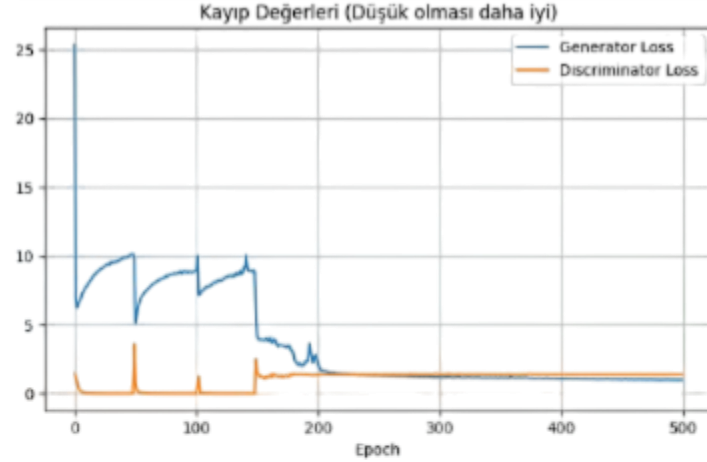
İzotropik Yapılarda Uygulama (Metal Yapılar)



Anizotropik Yapılarda Uygulama (Kompozit(CFRP))

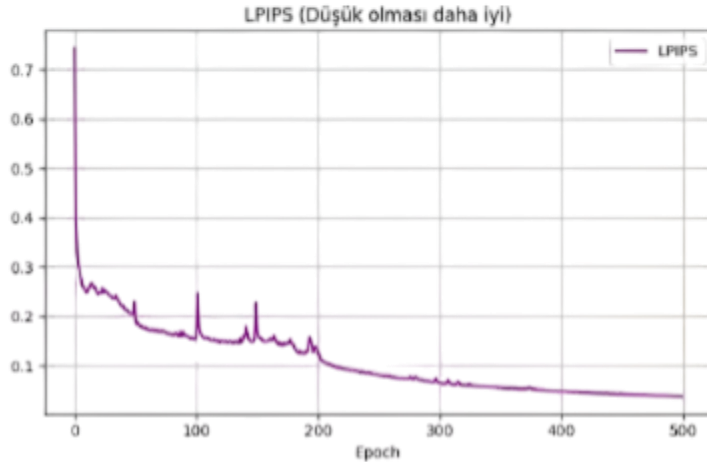


Araştırma Sonuçları ve Çıktılar



Üreteç ve ayırıştırıcı kaybı grafiği:

Eğitim sürecindeki öğrenme dinamiklerini anlamak için kullanılır. Üreteç kaybı, ayırıştırıcıyı yanıltma başarısını temsil ederken, ayırıştırıcı kaybı ise gerçek ve üretilmiş veriyi ayırt etme başarısını gösterir. Eğitim boyunca bu iki kayıp, rekabetçi bir şekilde birbirine yakın bir dengeye ulaşırsa, modelin başarılı bir şekilde öğrenme sağladığı kabul edilir.

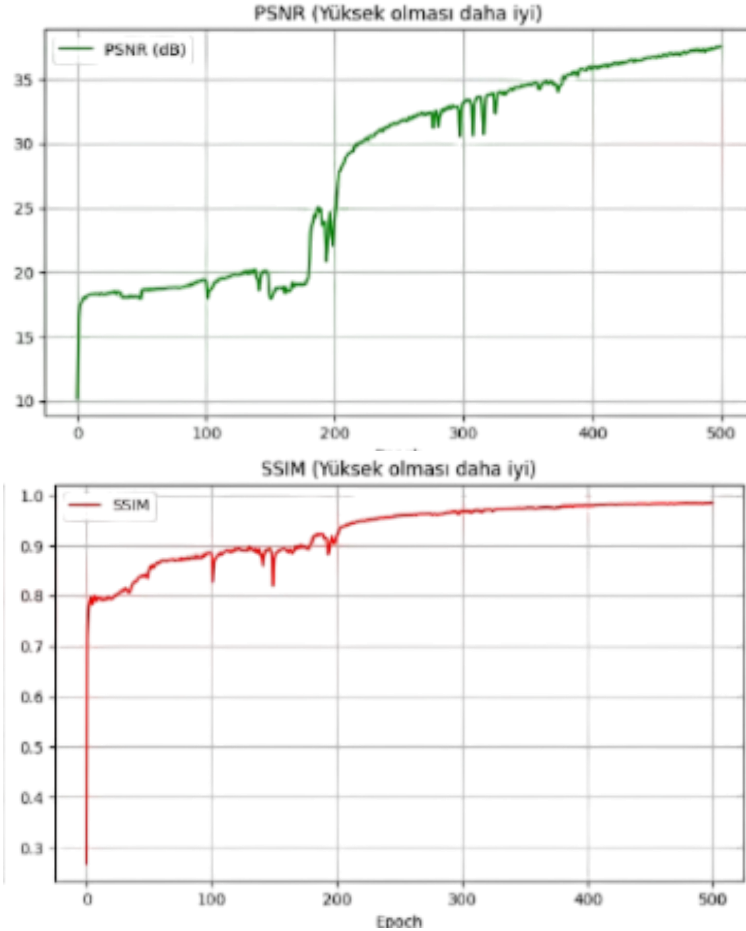


LPIPS (Learned Perceptual Image Patch Similarity)

İki görüntü arasındaki algısal benzerliği ölçmek için kullanılan bir metriktir.

Derin öğrenme modellerinden elde edilen özellik haritalarını kullanarak iki görüntü arasındaki görsel farkları insan algısına daha yakın bir şekilde değerlendirir. Daha **düşük** bir LPIPS değeri, görüntülerin birbirine algısal olarak daha yakın olduğunu ifade eder ve modelin çıktılarının gerçek görüntülerle ne kadar uyumlu olduğunu belirlemek için kullanılır.

Araştırma Sonuçları ve Çıktılar



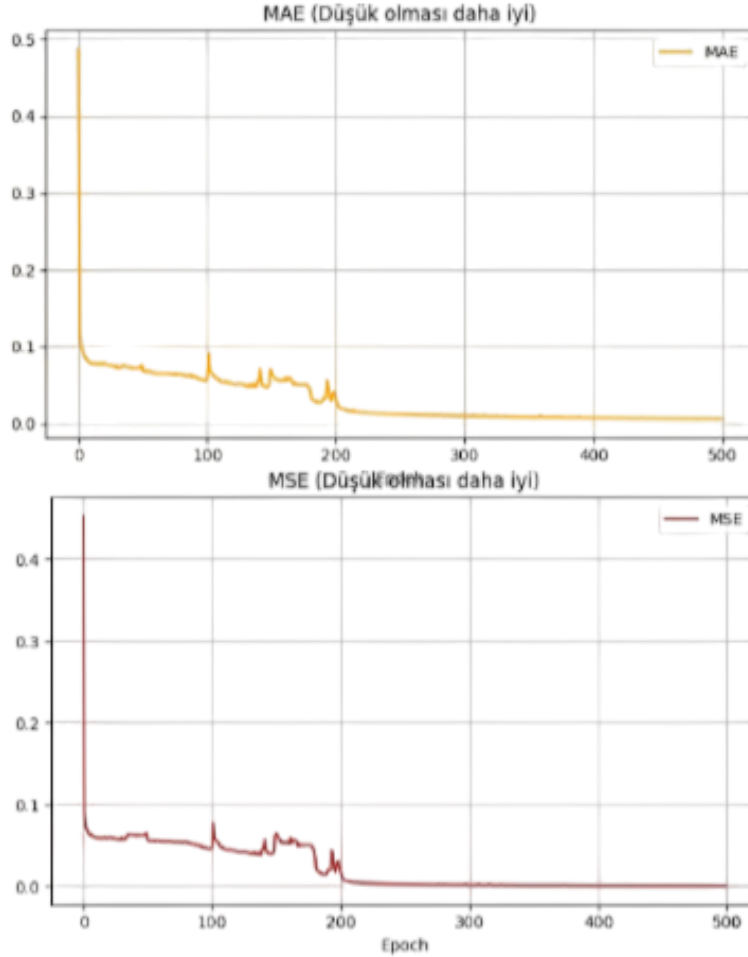
PSNR (Peak Signal-to-Noise Ratio)

PSNR, iki görüntü arasındaki kalite farkını ölçen bir metriktir ve görüntüler arasındaki farkın logaritmik ölçeği olarak ifade edilir. Daha **yüksek** PSNR değerleri, iki görüntü arasındaki farkın düşük olduğunu ve dolayısıyla daha iyi bir kaliteye işaret eder.

SSIM (Structural Similarity Index Measure)

SSIM, iki görüntü arasındaki yapısal benzerliği ölçmek için kullanılan bir metriktir. İnsan algısına daha yakın bir değerlendirme sunar ve parlaklık, kontrast ve yapı gibi faktörleri dikkate alır. SSIM değerinin **1'e yakın olması**, iki görüntünün yapısal olarak çok benzer olduğunu ifade eder.

Araştırma Sonuçları ve Çıktılar



➤ MSE (Mean Squared Error)

MSE, iki görüntü arasındaki piksel farklarının karesinin ortalamasını hesaplayan bir metriktir. Daha düşük bir MSE değeri, iki görüntü arasındaki farkın daha az olduğunu ve dolayısıyla modelin daha başarılı olduğunu gösterir.

➤ MAE (Mean Absolute Error)

MAE, iki görüntü arasındaki piksel farklarının mutlak değerlerinin ortalamasını ölçer. Daha düşük bir MAE değeri, modelin tahmin ettiği görüntünün gerçek görüntüye daha yakın olduğunu ifade eder ve genellikle hata ölçümleri için sade bir metrik olarak tercih edilir.



Proje Ekibinin Kazanımları ve Projenin TUSAŞ'a Olası Katkısı

Proje Ekibinin Kazanımları:

- Makine öğrenmesi ve derin öğrenme alanlarında uzmanlık kazanıldı.
- U-Net ve PatchGAN mimarilerinin mühendislik problemlerine entegrasyonu deneyimlendi.
- Hiperparametre optimizasyonunun model performansı üzerinde etkileri öğrenildi.

Projenin TUSAŞ'a Olası Katkısı:

- **Maliyet Tasarrufu**
- **Zaman Tasarrufu**
- **Tasarım Optimizasyonu**
- **Prototip Sürecinde Avantaj**
- **Kaynak Verimliliği**

neriler, Alınan Dersler ve Gelecekteki alıřmalar

neriler:

Modelin performansını artırmak ve daha etkili sonular elde etmek iin ařağıdaki neriler deęerlendirilebilir:

- Veri eřitlilięi
- Farklı GPU Trleri
- Hiperparametre Optimizasyonu

Alınan Dersler:

- Model Seimi ve Veri Uyumu
- Donanım Seiminin Etkisi
- Hiperparametre Ayarlarının Hassasiyeti
- Performans Metriklerinin Rol
- Daha Fazla Veriyle İyileřtirme Potansiyeli
- Veri Artırma Tekniklerinin nemi

Öneriler, Alınan Dersler ve Gelecekteki Çalışmalar

Gelecekteki Çalışmalar:

- Hiperparametre optimizasyonu ve veri çeşitliliği sağlanarak mevcut modelin performansı iyileştirilebilir.
- Modelin 3D analizlerde de kullanılabilmesi amacıyla geliştirililebilir.
- Sonuç model bir arayüze entegre edilerek kullanıcı deneyimi iyileştirilebilir.



Teşekkür Ederiz

Türk Havacılık ve Uzay Sanayii

TSKGV'nin bağlı ortaklığı ve SSB'nin iştirakidir.

Fethiye Mh. Havacılık Blv. No:17

06980 Kahramankazan-Ankara/TÜRKİYE

T +90 (312) 8111800 • F +90 (312) 8111425

www.tusas.com.tr