

Halil İbrahim Taşkömür

Mail : taskomurhalilibrahim@gmail.com

[linkedin.com/in/hitaskomur](https://www.linkedin.com/in/hitaskomur)

Telefon : +90 544 410 67 71

github.com/hitaskomur

İçerik:

-	Projenin Amacı	2
-	Veri Seti	2
-	Derin Öğrenme Modeli	3
-	Geliştirmeler	6
-	Sonuçlar	8

Projenin Amacı:

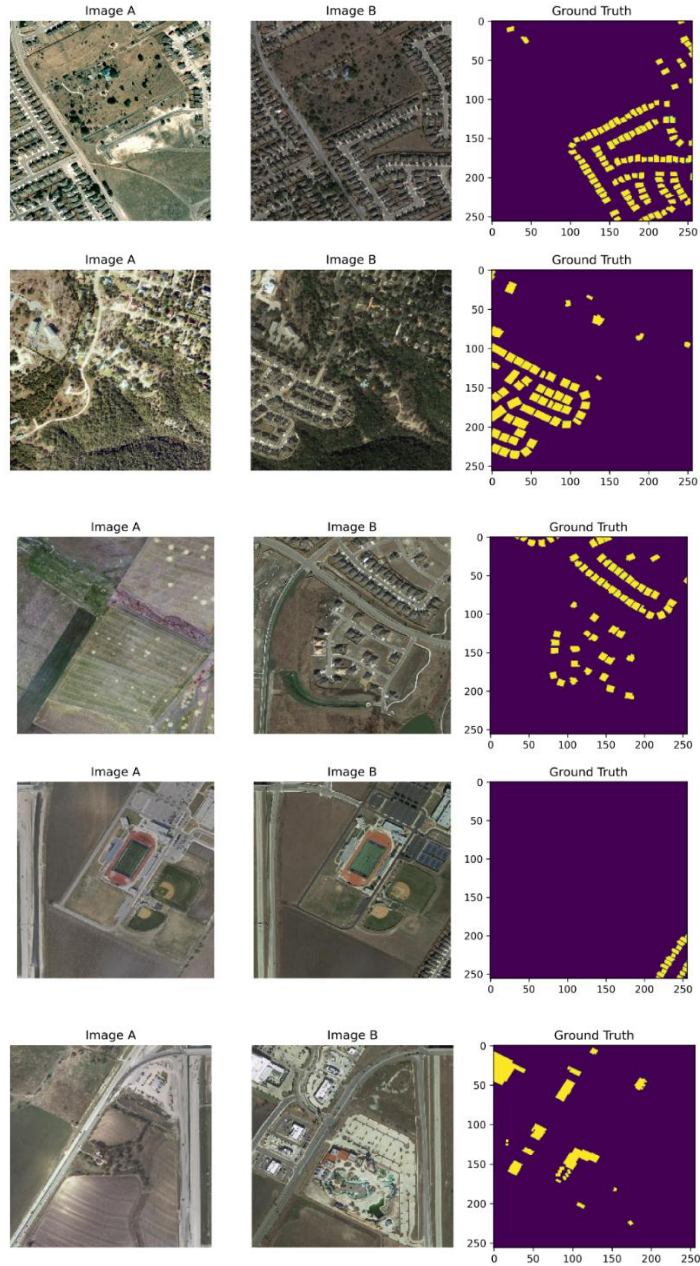
-2 farklı zamanda uydudan alınmış görüntüler arasındaki farklılıkları tespit etmek.

Veri Seti:

-[Levir-CD](#) isimli veri seti kullanılmıştır.

- LEVIR-CD, 1024×1024 piksel boyutunda 637 adet çok yüksek çözünürlüklü (VHR, 0,5 m/piksel) Google Earth (GE) görüntü parçası çiftinden oluşmaktadır. 5 ila 14 yıllık zaman aralıklarına sahip bu bir zamanlı görüntüler, özellikle inşaat artışı olmak üzere arazi kullanımında önemli değişiklikler göstermektedir. LEVIR-CD, villalar, yüksek apartmanlar, küçük garajlar ve büyük depolar gibi çeşitli bina tiplerini kapsamaktadır

Veri Seti Örnekleri



Derin Öğrenme Modeli:

-Verisetinde var olan görüntüleri modele girdi olarak vermeden önce birkaç preprocessing işlemleri uyguladık. Her görüntü için “resize”(yeniden boyutlandırma(256)) işlemi yaptık. Veri setinde var olan ilk ve son görüntüyü RGB(3 kanallı) olarak işleme aldık. İki RGB resmi birleştirme yaptıktan sonra girdi olarak modele verdik. Modelin 6 kanallı bir girdisi oldu($RGB(3)+RGB(3)=6$).

Siamese Unet gibi farklı derin öğrenme modellerinde iki görüntü birleştirilmeden modele girdi olarak verilir ve her fotoğraf için ayrı işlem yapılır. Bu da fazla kaynak gerektirir. Bizim yarattığımız modelde **daha az kaynak ile başarı elde ettik.**

-Fully Connected Neural Network algoritması kullanılmıştır. İçerisinde input , encoder, decoder ve output katmanları kullanılmıştır.

Model eğitimi için veri seti içerisinde bulunan train veri seti, validate etmek için validation dataseti, test için test data seti kullanılmıştır.

Model içerisinde aktivasyon fonksiyonları olarak “relu”, “sigmoid(çıktı katmanında)”, padding olarak “same” parametresi, upsampling katmanlarında interpolation olarak “bilinear” parametresi kullanılmıştır.

Model analizini daha anlamlı hale getirebilmek için genel olarak kullanılan metriklerin(accuracy) yanı sıra farklı olarak IoU(Birleşimlerin Kesişimi: segmentasyon ve nesne algılama algoritmalarının doğruluğunu değerlendirmek için kullanılan bir performans ölçümüdür.) metriği kullanılmıştır.

Farklı teknikler ve parametreler denenmesine rağmen en cost-effective model olarak bu model seçilmiştir.(U-Net, Siamese-U-Net, augmentation data etc.).

Model Mimarisi

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import InputLayer, Conv2D, MaxPooling2D, UpSampling2D
from tensorflow.keras.optimizers import Adam

def build_model(input_shape=(IMG_SIZE, IMG_SIZE, 6)):
    model = Sequential()
    model.add(InputLayer(input_shape=input_shape))

    # Encoder
    model.add(Conv2D(32, (3,3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2,2)))

    model.add(Conv2D(64, (3,3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2,2)))

    # Decoder - Conv2DTranspose yerine UpSampling + Conv2D
    model.add(UpSampling2D(size=(2,2), interpolation='bilinear'))
    model.add(Conv2D(64, (3,3), activation='relu', padding='same'))

    model.add(UpSampling2D(size=(2,2), interpolation='bilinear'))
    model.add(Conv2D(32, (3,3), activation='relu', padding='same'))

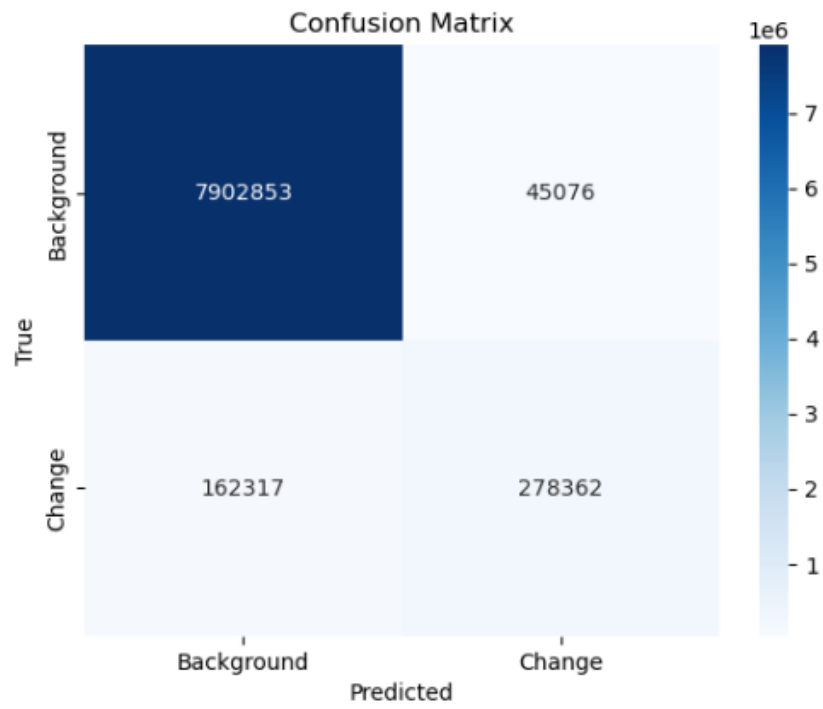
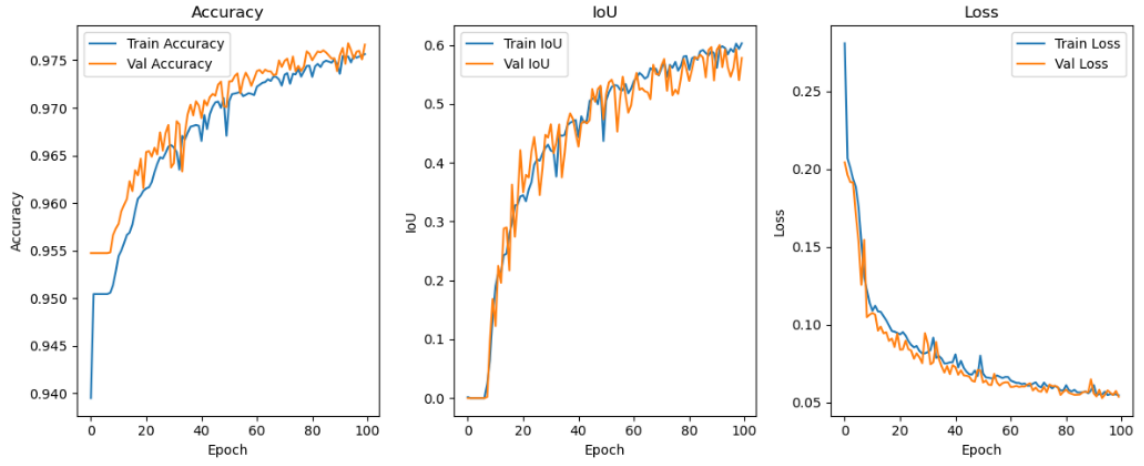
    # Last Layer (Predict Mask)
    model.add(Conv2D(1, (1,1), activation='sigmoid', padding='same'))

    model.compile(optimizer=Adam(), loss='binary_crossentropy', metrics=['accuracy', iou_metric])
    return model
base_model = build_model()
base_model.summary()
```

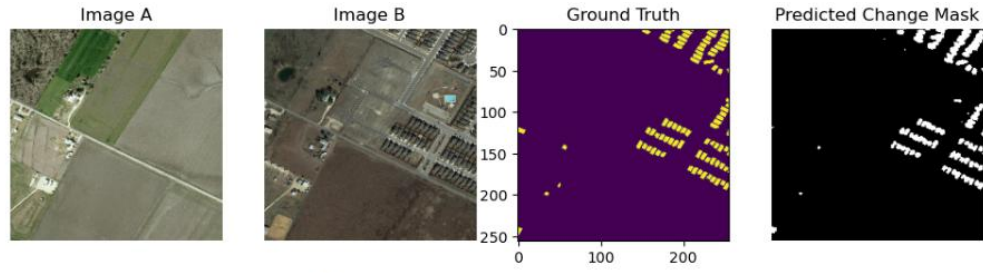
Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 256, 256, 32)	1760
max_pooling2d (MaxPooling2D)	(None, 128, 128, 32)	0
conv2d_1 (Conv2D)	(None, 128, 128, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 64, 64, 64)	0
up_sampling2d (UpSampling2D)	(None, 128, 128, 64)	0
conv2d_2 (Conv2D)	(None, 128, 128, 64)	36928
up_sampling2d_1 (UpSampling2D)	(None, 256, 256, 64)	0
conv2d_3 (Conv2D)	(None, 256, 256, 32)	18464
conv2d_4 (Conv2D)	(None, 256, 256, 1)	33
=====		
Total params: 75,681		
Trainable params: 75,681		
Non-trainable params: 0		

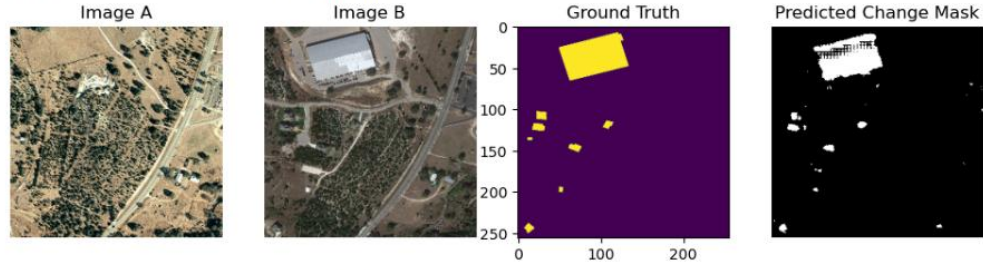
Model Parametrelerinin Sonuçları



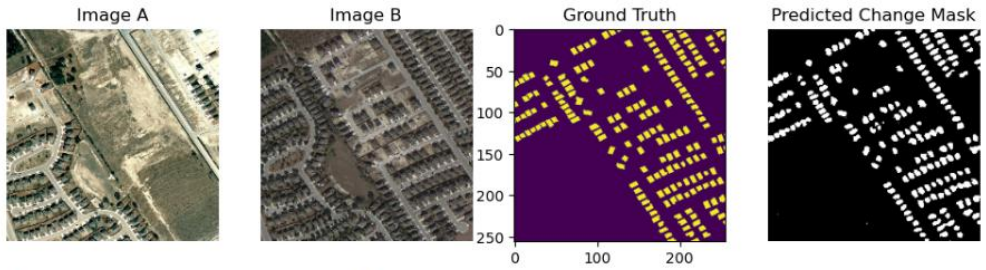
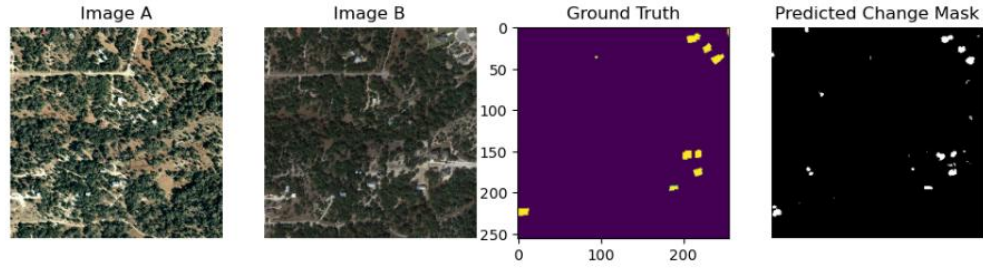
Test Sonuçlarını Görselleştirme



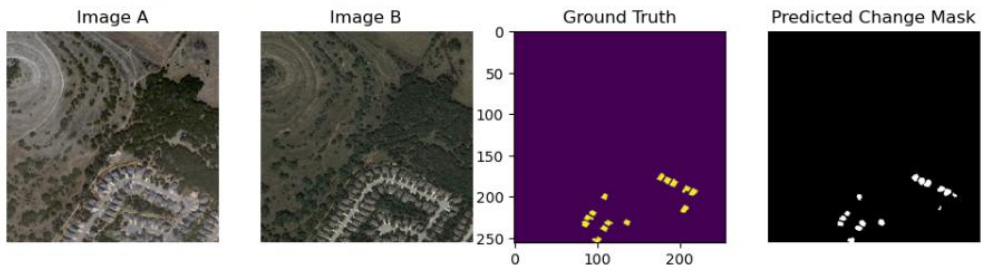
1/1 [=====] - 0s 32ms/step



1/1 [=====] - 0s 30ms/step



1/1 [=====] - 0s 26ms/step



Test Değerlendirme Sonuçları

-Test Loss: 0.0573, -Test Accuracy: 0.9733, -Test IoU: 0.6190

-Precision: 0.8413579301898346, -Recall: 0.6983813614898826,

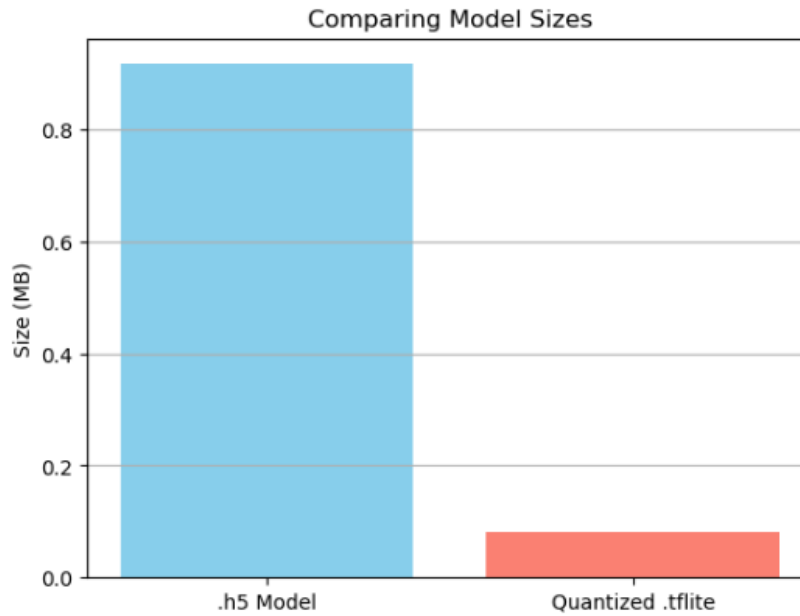
-F1 Score: 0.7632314119168575

Geliştirmeler:

-Quantization(Az kaynağa sahip donanımlarda(mobil,embedded systems etc.)kullanılabilmesi için boyut küçültme yapılmıştır.)

◆ Keras Model:
Precision: 0.7926137284783243
Recall: 0.7562726232094793
F1 Score: 0.7740168450383128

◆ Quantized TFLite Model:
Precision: 0.7991544736929022
Recall: 0.7485691303406243
F1 Score: 0.773035143769968



H5 Model Size: 0.92 MB
TFLite Quantized Model Size: 0.08 MB

Quantizing Rate: 11.15 times smaller

- Streamlit arayüz ile web sitesi

Change Detection with Satellite Images

Timezone 1 Image (T1)

Drag and drop file here
Limit 200MB per file • PNG, JPG, JPEG

Browse files

test_20.png 2.1MB X

Timezone 2 Image (T2)

Drag and drop file here
Limit 200MB per file • PNG, JPG, JPEG

Browse files

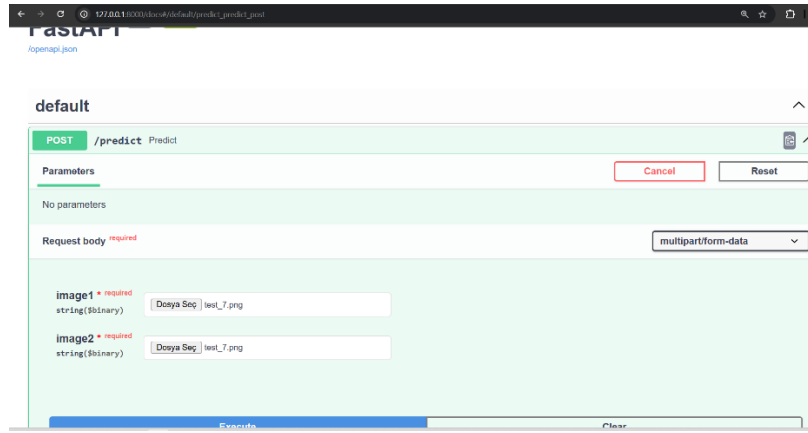
test_20.png 2.1MB X

Loaded Images

Image T1 Image T2



-Fastapi ile Api oluřturuldu.



Sonuçlar:

-Modelde spesifik olarak istenilen alanlar veya yapılar için daha detaylı fine-tuning yapılabilir.

-Veri arttırımı ile model sonuçları tekrardan değerlendirilebilir.

-Kullanılacak olan alana göre geliřtirmeler yapılabilir.