# Makine Öğreniminde Zemin Etüdü

Elif Özkan 211307057 Kocaeli Üniversitesi Bilişim Sistemleri Mühendisliği

Özet—Makine öğrenimi algoritmalarını kullanarak deprem farkındalığı için araziden Zemin etüdü yapmak istedim. Elimde dört kategori vardı arazi yapısını tespit edebildiği ve makine öğrenimi algoritmalarını kullanarak zeminin kumlu karstik sulak veya çamurlu olup olmadığını tespit etmek istedim. Bunun için Web Crawling yoluyla topladığım verilerimi test ve train olarak ayırdım.Daha sonrasında Levit ,Microsoft Vit, Swin gibi modellerle eğitmeye çalıştım.Bu projede temel amacım arazi yapılarından Zemin etüdü gerçekleştirerek uygun Zemin yapılarına inşa edilerek deprem farkındalığı yaratmak

Abstract—I wanted to conduct a ground survey from the field for earthquake awareness using maching learning algorithms I had four categories that could detect the land structure and I wanted to use machine learning algorithms to determine whether the ground was sandy,karst,wetland or muddy. For this purpose I divided the data I collected through web crawling into test and train. Then I tried to train it with models such as Levit, Microsoft Vit, Swin. My main aim in this project is to create earthquake awareness by carrying out ground surveys from land structures and building them on suitable ground structures.

### Keywords—Vit,Levit,Beit,Modelleme,Swin,Zemin,colab

# I. GOOGLE COLAB VE WEB CRAWLING SÜRECI

Google Colab ortamına ilk olarak elimde 4 kategoriye ait olan test ve train verilerini yükledim veri çoğaltma işlemini yaptıktan sonra test ve train olarak %20 ye %80 olarak ayrım yaptım. Modellerimin algoritmaları için Colab Notebooks içinden Jupyter Notebook'u kullanacağım dosyalarımı oluşturmaya başladım. Google Drive' a attığım verileri işleyebilmek için gerekli bağlantıyı sağladım ve Colab ortamında verilerimi gördüm. Bu adımdan sonra verilerimin doğruluk seviyeleri üzerinde denemeler yapmaya başladım. Verilerimi nasıl topladığım sorusunun cevabı ise Web Crawling yaparak Google Görseller üzerinden otomatik olarak Python kodu kullanarak çektim bunun için öncelikle Google için crawler yapabilmek için eklenti ekledim sonrasında ise Selenium kütüphanesini kullanarak otomatik veri çekebileceğim bir kod yazdım. Çalışabilmesi için Chrome sürümünün güncelliğinin uygun olması gerekiyor. Verilerimi topladıktan sonra 4 ayrı sınıfa ayırdım. Konum zemin etüdü üzerine olduğu için zemin kategorileri oluşturdum. Kumlu Zemin, Karstik Zemin, Çamurlu Zemin, Sulak Arazi şeklinde verileri sınıflandırdım daha sonrasında ise verilerimi yukarıda bahsettiğim şekilde çoğaltma yoluna gittim.

# II. MICROSOFT VIT

Microsoft VIT ilk deneme yaptığım model oldu. Model için gerekli drive bağlantısını sağladıktan sonra eğitim sürecini başlattım. Modeli, verileri işlemek için kullandığım Colab ortamında eğittim. Eğitim sırasında modelin performansını değerlendirmek için epoch başına kayıp (loss) ve doğruluk (accuracy) değerlerini takip ettim. Bu takibi yapmadan önce

verilerimi işlemeye başlamadan önce bana verilen bellek alanı kısıtlı olduğu için ve daha verimli bir ortam oluşturmak için mixed precision kullandım bu GPU'yu verimli şekilde kullanmamın yanı sıra eğitim süremi hızlandırmak istememdi. Hem belleği verimli kullandım hem de eğitim süremi önemli ölçüde kısaltmış oldum. Bu yöntemi kullanmanın dezavantajları da olabilir modeline ve verinizin boyutuna göre. Bu asamayı gectikten sonra model eğitimi yapabilmek için gerekli yapıları oluşturmaya başladım öncelikle verileri test ve train klasör yollarını verdim daha sonrasında kategori listesini belirttim. Sonrasında gerekli parametreleri verebilmek için ImageDataGeneration nesnesi kullandım rescale=1./255 parametresi, piksel değerlerini [0,1] aralığına ölçekler. rotation\_range=20, width\_shift\_range=0.2, height\_shift\_range=0.2 ve horizontal\_flip=True parametreleri, resimlere uygulanacak dönüş, kaydırma ve yatay çevirme artırmalarını belirtir. validation\_split=0.2 parametresi, veri setini eğitim ve doğrulama setlerine ayırırken kullandım

train\_generator ve validation\_generator, flow\_from\_directory yöntemini kullanarak eğitim ve doğrulama veri setlerini oluşturur. Bu yöntem, veri setini belirtilen dizinden yükler ve artırır. target\_size=(224, 224) parametresi, yüklenen resimlerin boyutunu belirtir. batch\_size=32 parametresi, her adımda işlenecek resim sayısını belirtir. class\_mode='sparse' parametresi, sınıf dizinlerini (0'dan başlayarak) döndürür. subset='training' ve subset='validation' parametreleri, veri setinin eğitim ve doğrulama altkümelerini belirtir.test\_datagen ve test\_generator, test veri seti için benzer bir işlemi gerçekleştirir. Ancak, burada sadece rescale işlemi yapılır ve veri artırma yapılmaz. shuffle=False parametresi, veri setinin karıştırılmayacağını belirtir. Bu işlemleri yaptıktan sonra modeli yükleyip derleme ve eğitim aşamasına geçtim.



Şekil-1

Kodumun bu kısmında model özellikleri belirtilir önceden eğitilmiş üst katmanların kullanılıp kullanılamayacağını resimlerin boyutları ve sınıf sayısı belirttim daha sonrasında ise model derledim. Eğitim sırasında kullanılan geri çağırıcılar ayarlamasını yaptıktan sonra eğitimde

kullanacağım grup boyutunu ve epoch sayısını ayarladım bu kısım ekstra olarak modelin eğitim ve sonuç alma süresini etkilediğinden ekstra olarak düzenleme yaptı

history vit - vit model. (it(	
· · · train generator.	
····validation data=validation generator.	
· · · epochs-epochs.	
··· verbose-1,	
··· callbacks-callbacks.	
· · · batch_size=batch_size	
)	
<ul> <li>Downloading data from <a href="https://github.com/faustomorales/vit-keras/releases/down">https://github.com/faustomorales/vit-keras/releases/down</a></li> </ul>	load/dl/ViT-B_32_imagenet21k+imagenet2012.noz
353253686/353253686 [] - 6s @us/step	
/usr/local/lib/python8.10/dist-packages/vit_keras/utils.py:81: UserWarming: Re warnings.warm(	sizing position embeddings from 12, 12 to 7, 7
Epoch 1/3	
121/121 [] - 1352s 11s/step - loss: 0.9336 - acc	uracy: 0.6271 - val_loss: 0.8546 - val_accuracy: 0.6750
Epoch 2/3	
121/121 [	curacy: 0.8072 val loss: 0.8973 val accuracy: 0.7103
Epoch 3/3	
121/121 [	curacy: 0.8696 - val loss: 0.8891 - val accuracy: 0.7331

Şekil 2

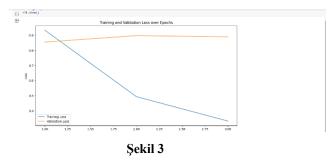
Bu kısımda ise modeli eğittim ve eğitim sırasında loss ve accurancy değerlerini epoch değerleri üzerinden aldım.

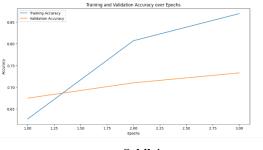
**Epoch 1:** Eğitim sürecinde, kayıp (loss) değeri 0.9336, doğruluk (accuracy) değeri ise 0.6271 olarak hesapladı. Bu, modelin eğitim veri seti üzerindeki performansını gösterir. Doğrulama sürecinde ise, kayıp değeri 0.8546, doğruluk değeri ise 0.6750 olarak hesaplandı. Bu, modelin genelleme yeteneğini ve aşırı uyuma (overfitting) karşı direncini gösterir.

**Epoch 2:** Eğitim sürecinde, kayıp değeri 0.4928, doğruluk değeri ise 0.8072 olarak hesaplandı. Modelin eğitim veri seti üzerindeki performansının arttığını gösterir. Doğrulama sürecinde ise, kayıp değeri 0.8973, doğruluk değeri ise 0.7103 olarak hesaplandı. Modelin genelleme yeteneğinde azalma olabileceğini gösterir.

**Epoch 3:** Eğitim sürecinde, kayıp değeri 0.3307, doğruluk değeri ise 0.8696 olarak hesaplandı. Modelin eğitim veri seti üzerindeki performansının daha da arttığını gösterir. Doğrulama sürecinde ise, kayıp değeri 0.8891, doğruluk değeri ise 0.7331 olarak hesaplandı. Modelin genelleme yeteneğinde azalma olabileceğini gösterir.

Genel olarak baktığımda en son ezberleyen bir model elde ettim. Bunun önüne geçmek için daha çok veri toplayıp optimizasyon yapabilir ve farklı düzenleme teknikleri model üzerinde kullanabilirdim.





Şekil 4

### Levit Modeli

Levit modelini bir önceki modelim ezber yapınca ve validation loss değerleri ve accurency değerleri arasında oldukça fazla fark olması sebebi ile modeli eğitme amacına uyacağını düsündüm cünkü Levit modeli Vit modele göre LeViT, farklı boyutlardaki ve şekillerdeki görüntülere daha iyi adapte olabilir. Bu, LeViT'in geniş bir görüntü veri kümesiyle daha iyi çalışmasını sağlar. Aynı zamanda LeViT, ViT'e göre daha az bellek tüketir. Bu, LeViT'in daha düşük bellek kapasitesine sahip cihazlarda veya sınırlı bellek kullanımı gerektiren uygulamalarda daha etkili olmasını sağlar. Bu özellik benim için ekstra önemliydi çünkü Google Colab'ın ücretsiz sürümü için sağladığı sınırlı bir kaynak kapasitesi var ve hem çok uzun saatler eğitilmesi gerekiyor hem de uygulamanızın sınırlı miktardaki ram kapasitesinden fazla bir miktar kullanmak istediğinizde oturumunuz kendiliğinden sonlanıyor. Bu sebeple Levit modeli Vit modeline göre çok daha avantajlı. Daha iyi sonuç veren bir model Levit.. Modeli kullanmaya başlamadan önce test ve train veri yollarımı verdim daha sonrasında CustomDataSet ile PyTorch'un dataset sınıfını genişleterek daha sonrasında resim için gerekli yolları ve etiketleri elde ettim. Daha sonra görüntülerimi işleyebilmek için onları girdi haline dönüştürmem gerekti. Görüntüleri isleyebilmek için onları tensör yapısına dönüştürerek işlenmesini kolaylaştırır. Aynı zamanda kullandığım RandomRotation ve RandomHorizantalFlip fonksiyonları ile resmin verdiğim açıyla dönmesini ve RandomHorizantalFlip görüntünün yatay eksende dönmesini sağlar. Daha sonra modelimin ezberlemesini önlemek için cross validation yaptım. Kfold nesnesi ile veri seti doğrulama yapmak için uygun hale getirir. Create model ile veri setini eğitmek için uygun yapıyı oluşturdum.

### • Fold 1:

- Epoch 1: Eğitim kaybı yüksek, doğrulama kaybı da yüksek ancak doğruluk makul düzeyde.
- Epoch 2: Eğitim ve doğrulama kayıpları düşmüş, doğruluk artmış.
- Epoch 3: Eğitim ve doğrulama kayıpları daha da düşmüş, doğruluk artmış.
   Modelin eğitimi iyi gidiyor gibi görünüyor.

# • Fold 2:

- Epoch 1: Eğitim ve doğrulama kayıpları yüksek, doğruluk düşük.
- Epoch 2: Eğitim ve doğrulama kayıpları düşmüş, doğruluk artmış.
- Epoch 3: Eğitim ve doğrulama kayıpları daha da düşmüş, doğruluk artmış.

Modelin eğitimi iyi gidiyor gibi görünüyor.

## • Fold 3:

- Epoch 1: Eğitim ve doğrulama kayıpları yüksek, doğruluk düşük.
- Epoch 2: Eğitim ve doğrulama kayıpları düşmüş, doğruluk artmış.
- Epoch 3: Eğitim ve doğrulama kayıpları biraz artmış, doğruluk hala yüksek.
   Modelin overfitting eğilimi gösterdiği görülüyor.

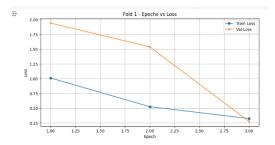
### • Fold 4:

- Epoch 1: Eğitim ve doğrulama kayıpları yüksek, doğruluk düşük.
- Epoch 2: Eğitim ve doğrulama kayıpları düşmüş, doğruluk artmış.
- Epoch 3: Eğitim ve doğrulama kayıpları biraz artmış, doğruluk hala yüksek.
   Modelin overfitting eğilimi gösterdiği görülüyor.

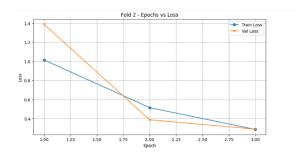
#### • Fold 5:

- Epoch 1: Eğitim ve doğrulama kayıpları yüksek, doğruluk düşük.
- o Epoch 2: Eğitim ve doğrulama kayıpları düşmüş, doğruluk artmış.
- Epoch 3: Eğitim ve doğrulama kayıpları biraz artmış, doğruluk hala yüksek.
   Modelin overfitting eğilimi gösterdiği görülüyor.

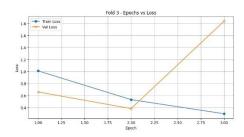
Genel olarak, modelin eğitimi iyi gidiyor gibi görünüyor ancak bazı katmanlarda overfitting eğilimi gözlenebiliyor. Bu durumu önlemek için düzenleme (regularization) teknikleri veya veri artırma (data augmentation) gibi yöntemler kullanılabilir. Ayrıca, eğitim süresince öğrenme oranının (learning rate) ve diğer hiperparametrelerin ayarlanması da modelin performansını iyileştirebilir.



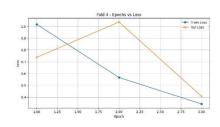
Şekil 5



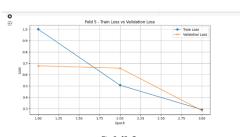
Şekil 6



Şekil 7



Şekil 8



Şekil 9

# KARŞILAŞTIĞIM ZORLUKLAR

Projeyi yaparken karşılaştığım en büyük zorluk modelleri eğitme konusunda grafik sonuçlarını almaktı. Google Colab üzerinden gerçekleştirdiğim süreçte grafiklerin sonuç olarak çıkması ve modelin ezber yapıp yapımadığı üzerindeki sonucu görmem saatlerimi aldı. Ayrıca Google Colab'ın ücretsiz versiyonu da oldukça yavaş çalıştığı için hem çalışan kodumdaki hataları görmem çok zamanımı aldı. Hem de diğer modeller üzerinde çalışmamı engelledi.

# KAYNAKÇA

 $\frac{https://huggingface.co/docs/transformers/model\_doc/vit\#tra}{nsformers.ViTModel}$ 

https://www.youtube.com/watch?v=XFFLvl1eySY

https://huggingface.co/docs/transformers/model\_doc/levit

https://sh-tsang.medium.com/review-levit-a-visiontransformer-in-convnets-clothing-for-faster-inference-3f8ea3eba63a PROJE LINKLERI

(Veri Toplama ve Crawling)
<a href="https://drive.google.com/drive/folders/1sFtike\_Gq7IEgjP">https://drive.google.com/drive/folders/1sFtike\_Gq7IEgjP</a>
<a href="https://drive.google.com/drive/folders/1sFtike\_Gq7IEgjP">Jnrp IBHg9G6v GAV?usp=sharing</a>

(Verilerin Tamamı Link: https://drive.google.com/drive/folders/1UVj3DlQzJZcG8IKDPxJu9I502OPbWApX?usp=sharing)

(Colab Notebook Linki: <a href="https://drive.google.com/drive/folders/15WAjxaRxPnD\_IcUv3qwfyfn2OMgUcWEG?usp=sharing">https://drive.google.com/drive/folders/15WAjxaRxPnD\_IcUv3qwfyfn2OMgUcWEG?usp=sharing</a>)