TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

UZAKTAN ALGILAMA GÖRÜNTÜLERİNDE GÖRÜNTÜ ÖZETLEME

18011103 — Ömer Talha BAYSAN
18011064 — Tolga SAĞLAM

BİLGİSAYAR PROJESİ

Danışman
Doç. Dr. Ali Can KARACA

Haziran, 2023

TEŞEKKÜR

Sayın Doç. Dr. Ali Can KARACA,

Öncelikle size ve Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği bölümüne en içten teşekkürlerimizi sunmak istiyoruz. Sizlerin ve bölümünüzün verdiği destek ve eğitimle hayatımızın en önemli adımlarından birini atmış bulunmaktayız.

Bu süreçte bize gösterdiğiniz ilgi, sabır ve yol göstericiliğiniz için size müteşekkiriz. Her zaman güncel ve ileri düzeydeki bilgileri biz öğrencilerle paylaşarak bizlere büyük bir vizyon kazandırdınız. Öğrencilerin gelişimine verdiğiniz önem ve destek için ayrıca teşekkür etmek istiyoruz.

Ailelerimize de minnettarlığımızı ifade etmek istiyoruz. Bize sağladıkları sürekli destek ve motivasyon, başarımızın arkasındaki en büyük güç kaynaklarından biridir. Ailelerimizin emekleri ve fedakarlıkları olmadan bu süreci başarıyla tamamlayamazdık. Bu yüzden, ailelerimize sonsuz teşekkürlerimizi sunuyoruz.

Saygılarımızla,

Ömer Talha BAYSAN
Tolga SAĞLAM

İÇİNDEKİLER

SİMGE LİSTESİ

β	Genellikle Çok Büyük Bir Sayı
x_{ij}^{I}	Yapay Sinir Ağına Giriş Bilgisi
y_{ij}^{l}	Yapay Sinir Ağı Çıkışı
σ	Aktivasyon Fonksiyonu
h_t	t Anındaki Gizli Hal
m	İki cümle Arasındaki Ortak Unigram Sayısı
u_m	Eşleştirilmiş Olan Unigram Sayısı
p_n	Modifiye Edilmiş n-gram Hassasiyeti
w_n	Her Bir Modifiye Edilmiş n-gram Hassasiyetlerinin Ağırlıkları

KISALTMA LİSTESİ

BLEU	BiLingual Linguistik Evolutional Understudy
CNN	Convolutional Neural Network
CPU	Central Process Unit
GRU	Gate Recurrent Unit
GPU	Graphical Process Unit
LSTM	Long Short Term Memory
METEOR	Metric for Evaluation of Translation with Explicit ORdering
RNN	Recurrent Neural Network
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
LCS	Longest Common Subsequence

ŞEKİL LİSTESİ

TABLO LİSTESİ

ÖZET

UZAKTAN ALGILAMA GÖRÜNTÜLERİNDE GÖRÜNTÜ ÖZETLEME

Ömer Talha BAYSAN
Tolga SAĞLAM

Bilgisayar Mühendisliği Bölümü
Bilgisayar Projesi

Danışman: Doç. Dr. Ali Can KARACA

Uzaktan algılama görüntülerinde görüntü özetleme, görüntünün özelliklerini belirleyerek seçilen görüntü için açıklama üreten derin ögrenme tekniğidir. Bu teknik derin ögrenme için önemli araştırma konuları arasındadır. Görüntü özetlemede doğru açıklamalar çıkarmak için bir çok farklı yöntem kullanılmaktadır. Son dönemlede çıkan yeni metodlar görüntü özetleme alanında önemli iyileştirmelere neden olmuştur. Dikkat mekanizması, bu metodalar arasındaki en önemlilerindendir. Projede öncelikle dikkat mekanizmasını ve diğer metodları birbirleri ile kombinleyerek görüntüden açıklama üretmek için kullanılacaktır.

Anahtar Kelimeler: Görüntü özetleme, derin öğrenme, açıklama üretme, dikkat mekanizması, doğal dil işleme

ABSTRACT

REMOTE SENSING IMAGE CAPTIONING

Ömer Talha BAYSAN
Tolga SAĞLAM

Department of Computer Engineering
Computer Project

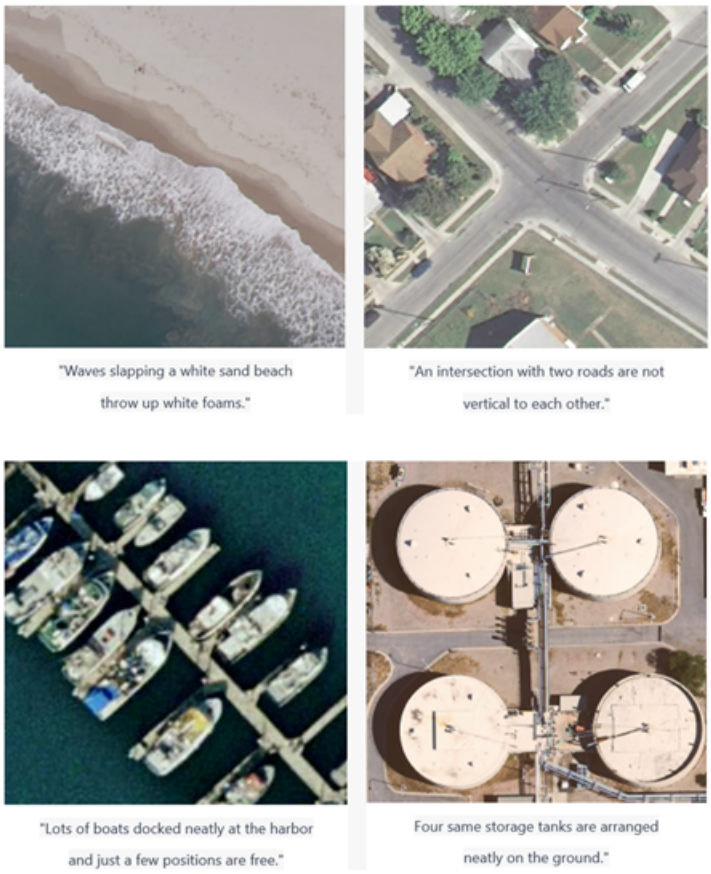
Advisor: Doç. Dr. Ali Can KARACA

Image captioning in remote sensing images is a deep learning technique that produces annotation for the selected image by determining the features of the image. This technique is among the important research topics for deep learning. Many different methods are used to extract accurate descriptions in image captioning. Recent new methods have led to significant improvements in the field of image captioning. The attention mechanism is one of the most important of these methods. In the project, it will primarily be used to produce an explanation from the image by combining the attention mechanism and other methods with each other.

Keywords: Image captioning, deep learning, annotation generation, attention mechanism, natural language processing

1
Giriş

Dijital görüntülerin özelliklerine bakarak altyazı oluşturma yöntemi olan görüntü özetleme, doğal dil işleme ve yapay sinir ağları gibi gelişmiş teknolojileri kullanmaktadır. Uzaktan algılamalı görüntü özetleme kavramı ise dron ve uçaklar gibi hava araçlarından elde edilen yeryüzü görüntülerin işlenmesidir. Yeryüzü görüntülerinden elde edilen bilgiler ile arama-kurtarma, keşif, trafik kontrolü ve daha pek çok alanda kullanılabilecektir.



Şekil 1.1 Bazı Uzaktan Algılamalı Görüntü Örnekleri

Şekil 1.1'de gösterilen görüntüler gibi tepeden çekilen, "god-view" olarak adlandırılmaktadır. Çalışma mantığı özetlenmek gerekirse veri setindeki

görüntülerin özellikleri çıkarılır. Bu işlemde Res-Net gibi CNN mimarileri kullanılmaktadır. Görüntülerin özellikleri çıkarıldıktan sonra bu bilgiler RNN mimarisine gönderilmektedir. İlk kelime "start token" verilir. Daha sonra, dikkat mekanizması ile önceki kelimeye ve çıkarılan özelliklere göre sıradaki kelime belirlenmektedir[1].

2
Ön İnceleme

Derin öğrenme teknolojisi, getirdiği gelişmeler ile Görüntü Özetleme teknikleri de gelişmeler göstermiştir. 2014 ve 2015 yıllarındaki Uzaktan Algılama Görüntü Özetleme yönteminin kullanımı ile derin öğrenme teknolojilerindeki yaygın kullanımın aynı yıllara denk gelmektedir. Bu alandaki literatür çalışmalarında (yaklaşık 30 farklı çalışmada) bu tekniklere üzerindeki farklı yaklaşımlar yer almaktadır. Bu yaklaşımlar arasında ön plana çıkan yaklaşım dikkat mekanızması yaklaşımıdır.Projede model hazırlanılırken dikkat mekanizması tekniği de denenecektir.

Proje başlamadan önce derin öğrenme ve makine öğrenmesi konularında yeterli deneyime veya bilgiye sahip olunmadığından dolayı projeye ayırılan zamanın çoğu projeyle ilgili konuların araştırılmasıyle ve öğrenilmesiyle geçirilmiştir.

Proje öncesinde, makine öğrenmesi konuş ile alakalı Udemy sitesinden Mustafa Vahit Keskin'in Python ile Veri Bilimi ve Makine Öğrenmesi kursu[2] alınmıştır. Bu kursta makine öğrenmesini ve kullanılan algoritmalara özel optimizasyon teknikleri öğrenilmiştir. Veri ön işleme ile modelleme öncesi verinin nasıl hazır hale getirileceği öğrenilmiştir.

Derin öğrenme konusu ile alakalı Udemy sitesinden DATAI TEAM'ın Deep Learning ve Python: A'dan Z'ye Derin Öğrenme kursu[3] alınmıştır. Bu kursta Logistic Regression, Artificial Neural Network (ANN), Convolutional Neural Network (CNN) ve Recurrent Neural Network (RNN) konuları hakkında bilgi edinilmiştir.

Projemizi konu alan ve Beigeng Zaho tarafından yazılmış olan "Remote Sensing Image Captioning" [1] makalesi incelenmiştir. Ek olarak TensorFlow'un resmi sitesindeki görüntü özetleme projesi incelenmiştir. Burdaki projede, kodlayıcı olarak InceptionV3 kullanılmaktadır ve Imagenet ağırlıkları yüklenmektedir. Kod çözücü kısım için GRU mimarisi kullanılmaktadır. Bahdanau Dikkat Mekanizması tercih edilmektedir. Projede MS-COCO veri seti kullanılmaktadır. Bu veri seti, 5 özet içeren yaklaşık 82.000

görüntü içermektedir. Proje, Google Collaboratory ile çalıştırılmıştır ve 20 epoch ile eğitililen model GPU hızlandırıcısı sayesinde eğitim işlemleri CPU'ya göre daha kısa sürmüştür.

İkinci bir inceleme olarak, " dabasajay" kullanıcısının Github'da paylaştığı bir görüntü özetleme projesi gözden geçirilmiştir[4]. Bu projede Tensorflow ve Keras derin öğrenme kütüphaneleri kullanılmaktadır ve Flickr8k veri seti kullanılmaktadır. InceptionV3 ve VGG16 modelleri, önceden eğitilmiş Imagenet ağırlıkları yüklenerek kodlayıcı olarak kullanılmaktadır. Kod çözücü olarak RNN ve LSTM (Uzun Kısa Süreli Bellek) yapıları tercih edilmektedir. Kod çözücüden çıkan ağırlıkların daha hızlı bir şekilde bulunması için açgözlü bir yaklaşım olan Işın Arama (Beam Search) ve Argmax teknikleri kullanılmaktadır. Projede 8GB memory GPU kullanıldığı belirtilmiştir. Eğitimler, farklı denemeler için farklı epoch değerleri ile tamamlanmaktadır.

3
Fizibilite

Bilgisayarlı görü ve doğal dil işleme konusu olan görüntü özetleme, hazırlanan modelin her bir görüntü için özet oluşturmayı amaçlamaktadır. Bu amacı gerçekleştirmek için modelin ilk olarak görüntünün içeriğini doğru anlaması ardından da tutarlı ve dilbilgisi kuralları doğru olan bir cümle oluşturmalıdır. Bu işlemin zorlu bir süreç olduğu bir gerçektir. Son yıllardaki derin öğrenme modellerindeki gelişmeler doğru ve tutarlı metinler oluşturabilen modellerin görüntü özetleme alanında hızlı bir gelişme sağlamıştır.

Projede, görüntü özetleme modeli için kodlayıcı-kod çözücü mimarisinden yararlanılacaktır. Projede belirlenen adımlar aşağıda listelenmiştir.

- Öznitelik çıkarmada kullanılan model, görüntünün özelliklerini belirlemek için CNN mimarisi kullanacaktır. Bu öznitelikler bir sonraki adımın girdisi olacaktır.
- Dil modelleme adımında ise öznitelik çıkarma aşamasından gelen girdiler kullanılarak görüntünün metinsel özetini çıkarılacaktır. Bu işlem, bir kelime dizisindeki sonraki kelimenin tahmin edilmesiyle gerçekleştirilecektir ve kodlayıcı-kod çözücü mimarisi kullanılacaktır.
- Dil kod çözme aşamasında, oluşturulan metinsel özetin dilbilgisi kurallarına göre doğru ve anlamlı olduğu incelenmek için özetin kodu çözülecektir. Bu işlemde RNN mimarisi kullanılarak gerçekleştirilecektir ve dilbilgisi açısından doğru ve anlamlı cümleler üretilebilecektir.

3.1 Teknik Fizibilite

Teknik Fizibilite içerisinde Yazılım, Donanım ve İletişim Fizibilitesi içermektedir.

3.1.1 Yazılım Fizibilitesi

Projede kullanılan programlama dili Python, makine öğrenimi ve yapay zeka alanlarında kullanılan yaygın bir dildir. Python dilinin basit, anlaşılabilir ve çok yönlü olması projede için tercih sebebidir.

Projenin gerçekleştirilmesinde bir makine öğrenimi ve derin öğrenme kütüphanesi olan TensorFlow kütüphanesi kullanılacaktır. TensorFlow, karmaşık makine öğrenimi modellerinin oluşturulmasını ve eğitilmesini sağlamaktadır. Açık kaynaklı olmasından dolayı dünya genelindeki geliştiriciler ile araştırmacılar tarafından geliştirilmektedir.

Modeller, Google Collaboratory kullanılarak ile eğitilecektir. Google Collaboratory, geliştiriciler için ücretsiz olaran bulut tabanlı bir makine öğrenimi platformudur. Anlaşılması kolay arayüzü ve güçlü GPU kaynakları sayesinde proje için ideal seçimlerden biridir.

3.1.2 Donanım Fizibilitesi

Bu proje, doğal dil işleme, görüntü işleme ve makine öğrenmesi alanlarını kapsamaktadır. Bu alanlar yüksek performansa ihtiyaç duydukalrı için, donanım bakımından yüksek özellikli bilgisayarlara ihtiyaç duyulmaktadır. Projede, ücretsiz olarak yüksek ram ve güçlü ekran kartlarına sahip olan Google Collaboratory kullanılacaktır.

Google Collaboratory Intel Xeon CPU @2.20 GHz işlemci, 13 GB RAM, Tesla K80 ekran kartı ve 12 GB GDDR5 VRAM sistemsel özellikleri ile çalışmaktadır.

Proje ekibinin sahip olduğu bilgisayar Intel i5 - 1135G7 işlemci, 16 GB RAM, 2GB Nvdia MX 2060 ekran kartı sistemsel özellikleri ile çalışmaktadır.

Veri seti, Google Collaboratory üzerinde eğitilirken bulut tabanlı veri depolama platformu olan Google Drive'da ve aksi durumlara karşı kişisel bilgisayarımızdaki depolama alanında da saklanacaktır.

3.1.3 İletişim Fizibilitesi

Proje ekibi, eğitim döneminin YÖK tarafından hibrit gerçekleştirilmesinden dolayı düzenli olarak gerçekleştirilecek çevrimiçi toplantıların Zoom ve Discord platformları aracılığıyla yapılacaktır. Bu toplantılarda, projedeki ilerleme, fikir paylaşımı ve çözüme ihtiyaç duyulan herhangi bir sorun ele alınacaktır. Ekip üyeleri bu yaklaşım ile iletişim halinde ve projeyi verimli bir şekilde gerçekleştirecektir.

3.2 İş gücü ve Zaman Planlaması

Projenin 3 ay süreceği öngörülmektedir. Ekip üyeleri proje geliştirme sürecini planlama çizelgesi Şekil 3.1 gösterildiği gibi modüler bir şekilde gerçekleştirecektir.

İP No	İş Paketinin Adı	HAFTALAR (27.02.2023 – 19.06.2023)														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Danışman Hoca ile Görüşme	x															
2	Proje Araştırması		x	x	x	x											
3	Veri Seti İnceleme					x	x										
4	Ara Rapor Hazırlanması						x	x									
5	Temel Modelin Oluşturulması						x	x	x								
6	Modelin Optimizasyonu									x	x						
7	Farklı Modellerin Oluşturulması										x	x					
8	Farklı Modellerni Optimizasyonu												x	x			
9	Final Raporu													x	x	x	x

Şekil 3.1 İş Gücü - Zaman Çizelgesi

3.3 Ekonomik Fizibilite

Projemiz, erişimi ücretsiz olan Google Collaboratory platformunu kullanacağından dolayı donanım için gereken maliyet azaltacaktır. Açık kaynak kodlu olan Tensorflow yazılımı ücretsiz kullanılabilmektedir. Ayrıca geliştiricilere ücretsiz olarak sunulan UCM Captions veri seti kullanılacağından dolayı veri toplamak için ek bir kaynak harcamasına gerek kalmayacaktır.

Tablo 3.1 Projede Çalışacak Ekibin Maliyeti

İsim	Haftalık Çalışma Süresi	Proje Süresi	Aylık Ücret	Toplam Maliyet
Ömer Talha BAYSAN	18 Saat	3 Ay	9000 TL	27000 TL
Tolga SAĞLAM	18 Saat	3 Ay	9000 TL	27000 TL

Tablo 3.1'deki maliyetler dikkate alındığında proje ekibine ödenmesi gereken tutar 54000 TL'dir.

Tablo 3.2 Projede Kullanılacak Donanımların ve Yazılımların Maliyeti

Ürün	Fiyat
Intel i5 - 1135G7 işlemci, 512GB SSD, 16 GB RAM 2 GB Nvdia MX450 ekran kartı ile Donatılmış Bilgisayar	18500 TL
Microsoft Windows 10	3500 TL

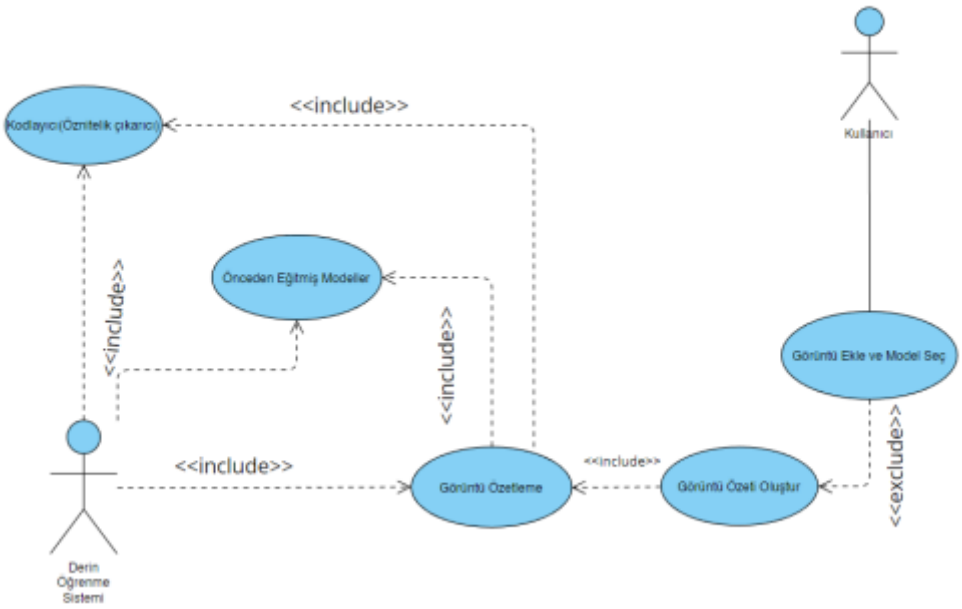
Projede kullanılacak donanımların ve yazılımların maliyeti Tablo 3.2'de gösterilmiştir. Tablodaki maliyetler dikkate alındığı zaman donanım ve yazılım gereksinimleri için ödenmesi gereken tutar 22000 TL'dir. Toplam proje maliyeti hesaplandığında projenin maliyeti 76000 TL'dir. Maliyetler incelendiğinde projenin lisanslanması halinde gelir elde edebileceği öngörülmektedir.

3.4 Yasal Fizibilite

UCM Captions veri seti araştırma amaçlı kullanım için ücretsiz kullanıma sunulmaktadır. Açık kaynak kodlu olan Tensorflow yazılımı ücretsiz kullanılabilmektedir. Google Collaboratory ise Google tarafından sunulan bulut tabanlı bir platformdur. Her iki yazılım da yasalara uygun bir şekilde proje için kullanılmaktadır. Bununla birlikte, veri koruma yasaları dikkate alınmıştır ve herhangi bir yasal sorunla karşılaşmadan başarılı bir şekilde projenin tamamlanması öngörülmektedir.

4
Sistem Analizi

Kullanıcının bir görüntü eklemesiyle sistemin işleyişi başlamaktadır. Daha sonra, eğitilmiş 16 modelden biri seçilip seçilen görüntü, CNN tarafından işlenmektedir.Kodlayıcı ise görüntünün öznitelikleri belirlemektedir. Bu öznitelikler modele gönderilip kod çözücü tarafından özelliklere göre görüntünün özeti oluşturulmaktadır. Ardından tahmin edilen özet kullanıcıya verilir.



Şekil 4.1 Use Case Diyagramı

4.1 Sistem Başarısının Ölçülmesi

Modellerimizin performansını değerlendirmek için cümleler arasındaki benzerlikleri tespit etmek için hazırlanan metrikler kullanılmıştır.

- BLEU-1

- BLEU-2
- BLEU-3
- BLEU-4
- METEOR

4.1.1 BLEU

Benzerlikleri tespit etmek için yaygın yöntemlerden olan BLEU (Bilingual Evaluation Understudy), görüntü özetleme modellerinde de kullanılmaktadır. Bu metot, doğru cümle ile tahmin edilen cümledeki ardışık kelimeleri (n-gram) hesaplamaktadır. BLEU skoru için hazırlanan formül bulunmaktadır.

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n log p_n) \tag{4.1}$$

BP (Brevity Penalty), tahmin edilen cümle ile doğru cümle arasındaki uzunluk farkını hesaba katmak için kullanılmaktadır. BP değişkeni için hazırlanan formül bulunmaktadır..

$$BP \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c \leq r \end{cases} \tag{4.2}$$

Ardışık kelimeler için N değeri kullanılmaktadır. p_n, değiştirilmiş n-gram hassasiyeti için kullanılan değerdir. w_n ise değiştirilmiş her bir n-gram hassasiyetinin ağırlığı için kullanılan değerdir. N = n olmasıyla hesaplanan değer BLEU-n değerini oluşturmaktadır.

4.1.2 METEOR

METEOR skoru, BLEU skoruna benzer olarak bir Benzerlikleri tespit ölçütüdür. BLEU skordan fakrlı olarak METEOR skoru, anlamdaş kelimeler ve kelimelerin köklerini de dikkate alarak doğru cümle ile tahmin edilen cümle arasında hizalama hesaplanmaktadır. Hesaplanan hizalama değeri sonucunda precision(doğruluk) ve recall(doluluk) değerleri elde edilmektedir. Bu yöntem ile cümlelerin anlamı korunup eş anlamlı kelimeler kullanılarak METEOR skoru hesaplanmaktadır.

$$P = \frac{m}{w_c} \tag{4.3}$$

$$R = \frac{m}{w_R} \tag{4.4}$$

Formülde kullanılan m değeri, tahmin edilen cümle ile doğru cümle arasındaki ortak unigram sayısı için kullanılmaktadır. w_c ve w_R ise tahmin edilen cümle ve doğru cümledeki toplam unigram sayısı için kullanılmaktadır. Doğruluk ve doluluk değerlerinden harmonik ortalama elde edilmektedir.

$$F_{mean} = \frac{10PR}{R + 9P} \tag{4.5}$$

Doluluk değeri ile doğruluk değeri arasında 9 kat ağırlık bulunmaktadır. Doğru cümle ile tahmin edilen cümlenin kelimelerindeki sıralamalar farklılık gösterse bile aynı kelimeler içermesi değeri 1 olarak kabul edilmektedir. Bu durumun oluşması istenmediği için aşağıdaki formül kullanılmaktadır.

$$p = 0.5\left(\frac{c}{u_m}\right)^3 \tag{4.6}$$

Formüldeki p değişkeni yığın hatası (chunk penalty) olarak adlandırılan düzeltme faktörüdür. Eşleştirilmiş unigram sayısını u_m değeri, c ise ardışık eşleştirilen unigram sayısı için kullanılmaktadır. Yığın terimi, ardışık eşleştirilen unigram sayısını ifade etmektedir. METEOR skorunu hesaplamak için aşağıdaki formül kullanılmaktadır.

$$M = F_{mean}(1 - p) \tag{4.7}$$

5
Sistem Tasarımı

Projede, CNN için VGG-16 ve ResNET mimarileri ve RNN için standart RNN, LSTM, BiLSTM ve GRU mimarileri kombinasyonları dikkat mekanizması kullanılarak gerçekleştirilmiştir.

CNN için VGG-16 ve ResNET mimarileri öznitelik çıkarma konusunda başarılı olduklarından dolayı tercih edilmiştir. Bu mimarileri proje için uygun hale getirmeden önce UCM Captions veri setine veri temizleme gibi ön işlemler uygulanmıştır.

RNN için standart RNN, LSTM, BiLSTM ve GRU mimarileri tercih edilmiştir. Bu mimariler, metin verilerinin işlenmesinde ve uzun vadeli bağımlılıkları öğrenebilen başarılı mimarilerdir. Standart RNN yapısı, adım adımda güncellenen gizli bir durum kullanırken, LSTM ve GRU yapıları ek mekanizmalara sahip olduklarından dolayı uzun vadeli bağımlılıkları işleyebilme kapasiteleri yüksektir.

Modelde, optimizasyon işlemi için Adam optimizasyon algoritması tercih edilmiştir. Adam algoritması, öğrenme hızını verilen parametreye göre uyarlayarak ağırlıkların uygun şekilde ayarlanmasını sağlamaktadır.

Modelin başarısı, Accuracy metriği epoch başına kullanılarak ölçülmüştür. Accuracy metriği, doğru oluşturulan tahmin cümlesinin tüm tahmin cimleler içindeki oranını ölçmektedir. Ancak modelin genel başarısını ölçmek için daha gelişmiş metrikler olan BLEU ve METEOR metrikleri kullanılmıştır.

5.1 Veri Seti Hazırlama

Projede uzaktan algılama görüntüleri içeren UCM-Captions[5] veri setini kullanılmıştır. UCM-Captions veri seti, her biri 5 adet özet cümleye sahip olan toplamda 2100 görüntü içermektedir.

Görüntülerin boyutları 224x224 piksel olacak şekilde işlenmiştir. Veri setinin yüzde 70'i eğitim, yüzde 20'si validation için yüzde 10'u test için ayrılmıştır.

5.2 CNN

CNN mimarisi görüntü işleme, yüz tanıma, nesne tanıma, doğal dil işleme, görüntü özetleme ve daha pek çok alanda kullanılmaktadır. CNN mimarisi havuzlama katmanları, konvolüsyonel katmanlar ve tam bağlı katmanlarınan oluşmaktadır.

CNNmimarisi, veri öğrenmek amacıyla farklı katmanlar ile yapılandırılmıştır. Konvolüsyonel katman, girdiyi almaktadır ve özellik haritası çıkarmak için bir dizi filtre uygulamaktadır. Bu haritalar bir sonraki katmana aktarılmaktadır. Bu işlem birkaç kez tekrarlanıp verilerin yüksek seviyeli özelliklerini öğrenmektedir. Bu filtreler, 3x3 ya da 5x5 boyutlarında olmaktadır. Görüntüdeki kenarlar, dokular gibi belirli kalıpları tanımlamak için tasarlanmıştır. *N x N*'lik bir nöron katmanına *m x m*'lik bir w filtreli konvolüsyon katmanı uygulanırsa formül aşağıdaki gibi olmaktadır [6].

$$x_{ij}^{l} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} w_{ab} y_{(i+a)(j+b)}^{l-1} \tag{5.1}$$

Projedeki konvolüsyonel katmanın çıktısı, ReLU olarak adlandırılan belirli bir doğrusal olmayan aktivasyon fonksiyonundan geçirilmiştir.

$$y_{ij}^{l} = \sigma\left(x_{ij}^{l}\right) \tag{5.2}$$

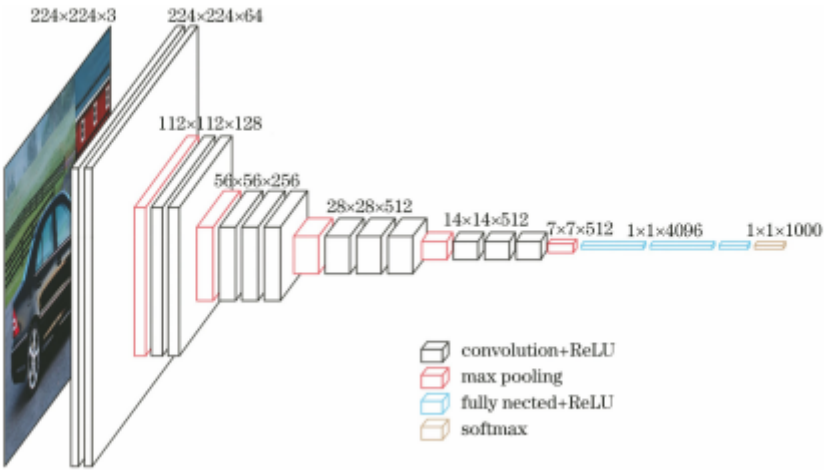
CNN mimarisinde kullanılan havuzlama katmanları, verilerin boyutunu azaltmak ve aşırı öğrenmeyi engel olmak için ideal bir yöntemdir. Özellik haritalarındaki değerleri küçülterek ve çıktıların boyutunu azaltarak hesaplama yükünün azalmasını sağlamaktadır. En yaygın kullanılan havuzlama katmanlarından biri olan maksimum havuzlama katmanı belirli bir çerçeve içinde en yüksek değeri almaktadır. Bu sayede hesaplama maliyetini azaltmaktadır ve modelin güvenilirliğini arttırmaktadır. Buna alternatif olarak ortalama havuzlama katmanı bulunmaktadır. Bu katman ise bitişik piksellerin ortalaması alaarak boyut küçültülmektedir.

Bir *k x k* bölgesi için maksimum havuzlama işlemi yapıldığı zaman, tek bir sonuç elde edilmektedir. Örneğin, giriş katmanının boyutu *N* x *N* ise, çıkış katmanının boyutu $\frac{N}{K}$ x $\frac{N}{K}$ şeklinde oluşturulmaktadır.

CNN mimarilerinde tam bağlantılı katmanlar genellikle son katmanda kullanılmaktadır. Tam bağlantılı katmanlar önceki katmanlardan gelen öznitelik vektörlerini alarak sınıflandırma işlemi için bir nihai çıktı üretmektedir. Bu çıktı, farklı sınıflar için olasılık dağılımı şeklinde ifade edilmektedir. Bu katmanlar, standart

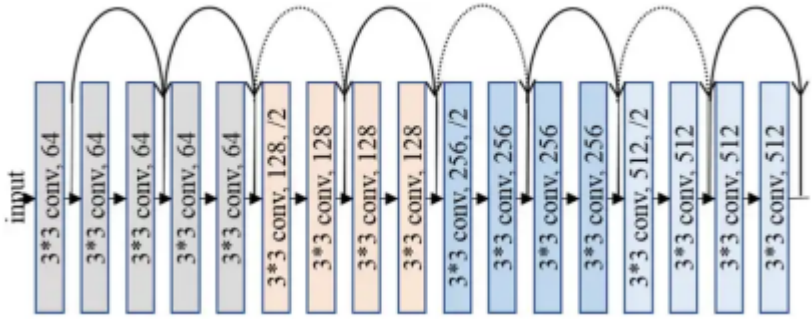
bir ileri beslemeli sinir ağındaki katmanlara benzemektedir. Konvolüsyonel ve havuzlama katmanları tarafından oluşturulan özellik haritalarını temsil eden verilerin üst düzey soyutlamalarını öğrenmek için kullanılmaktadır. Tam bağlantılı katmanlar, modelin sonucunu üretip bu soyutlamaları kullanarak verileri sınıflandırmak için tasarlanmıştır.

Projede, iki özel CNN mimarisi olan VGG-16 ve ResNet kullanılmıştır. Oxford Üniversitesi'ndeki Görsel Geometri Grubu tarafından geliştirilen VGG-16, büyük bir görüntü veri seti olan ImageNet'te 1000 sınıflı sınıflandırma görevinde çok iyi sonuçlar elde eden derin bir konvolüsyonel ağdır[7]. VGG-16, 3x3'lük küçük konvolüsyonel filtreler ve 16 katmanlı, derin bir mimari kullanmasıyla tanınmaktadır. VGG-16'nın mimarisi, Şekil 5.1'deki gibi gösterilmektedir.



Şekil 5.1 VGG_16 Mimarisi

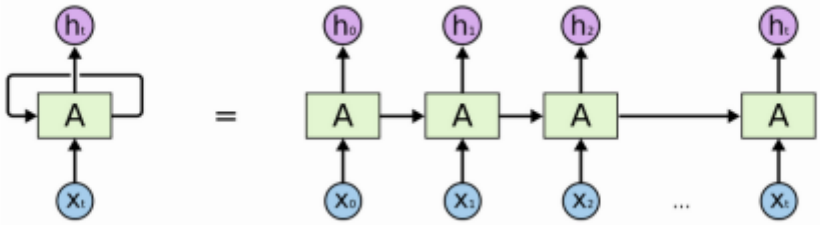
Resnet(Artık Ağ), derin sinir ağlarının daha kolay eğitilebilmesini sağlamak için geliştirilen bir yapısal yenilik olan "artık ağlar" kullanılarak tasarlanmış popüler bir CNN mimarisidir. Bu yapısal yenilik sayesinde ResNet daha derin ağların daha kolay eğitilmesine olanak tanımaktadır. Genellikle çeşitli görüntü sınıflandırma ve nesne algılama görevlerinde önde gelen modeller arasında yer almaktadır. 2015 yılında geliştirilmiş ve yüzde 3,57'lik hata oranıyla ImageNet yarışmasında ilk 5'e girmeyi başarmıştır. [8] ResNet sayesinde daha derin mimariler kullanılmasına olanak sağlanmıştır. Bu ağlar, kaybolan gradyanlar sorununu çözmüştür. ResNet mimarisi, her bloğun birden fazla katman ve artık bağlantılar içerdiği artık bloklardan oluşmaktadır ve bu mimari Şekil 5.2'de gösterilmiştir.



Şekil 5.2 ResNet Mimarisi

5.3 RNN

RNN (Tekrarlayan Sinir Ağı), zaman serileri gibi ardışık verilerin işlenmesinde kullanılan bir yapay sinir ağı modelidir. RNN, her veri girişi için aynı işlemleri yaparken, dahili belleklerini kullanarak geçmiş bilgileri hatırlayabilmektedir ve gelecekteki çıktıları bu bilgilerle şekillendirebilmektedir. Bu sayede RNN, girdiler arasındaki zaman, sıra veya diğer ilişkileri modelleyebilmektedir ve özellikle sıralı veriler gibi yapısal olarak karmaşık verilerde başarılı sonuçlar vermektedir. İleri beslemeli sinir ağlarında olduğu gibi, RNN de birçok farklı uygulama alanında kullanılmaktadır. Örneğin dil işleme, konuşma tanıma, müzik oluşturma, video analizi ve makine çevirisi gibi birçok alanda başarılı bir şekilde kullanılmaktadır.



Şekil 5.3 RNN Yapısı

İlk adımda, girdi dizisinden X(0) alınmaktadır. Bir sonraki adımın girdisi olan h(0)'ı X(1) ile birlikte vermektedir. Bu süreç, her adımda bir önceki çıktının bir sonraki girdiye eklenmesiyle devam etmektedir. Dolayısıyla, her adımda h(x) ve X(x+1), bir sonraki adımın girdisini oluşturmaktadır. Bu işlem, eğitim sırasında bağlamı korumak için kullanılmaktadır. Hesaplama, Current State (Mevcut Durum) formülüyle belirtilmektedir:

$$h_t = f(h_{t-1}, x_t) \tag{5.3}$$

Ardından aktivasyon Fonksiyonları Uygulanmaktadır:

$$h_t = tanh(W_{hh}h_{t-1} + W_{xh}X_t) \tag{5.4}$$

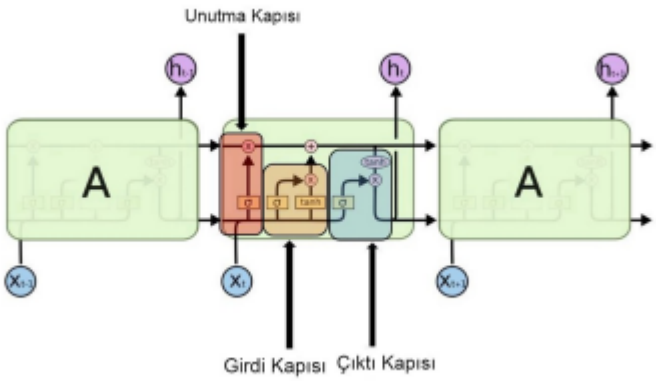
W ağırlık, h tek gizli vektör, W_{hh} önceki gizli haldeki ağırlık, tanh ise aktivasyonları [-1,1] aralığına sıkıştıran doğrusal olmayan bir aktivasyon fonksiyonudur. Çıktı:

$$y_t = W_{hy}h_t \tag{5.5}$$

y_t çıktı durumudur, W_{hy} ise çıktı durumundaki ağırlıktır.

5.4 LSTM

LSTM(Uzun-Kısa Süreli Hafıza), RNN (Recurrent Neural Network) mimarisinin geliştirilmiş bir versiyonudur. RNN mimarisinin hatırlama kapasitesini ile geliştirilmiş versiyonudur. RNN mimarisinde kaybolan gradyan sorunu ise LSTM yapısında çözülebilmektedir. LSTM yapısı, değişken gecikmeli zaman serilerini işlemek, sınıflandırmak ve tahmin etmek için uygun bir mimaridir. Model geriye doğru yayılım algoritmasını kullanarak eğitilmektedir.



Şekil 5.4 LSTM Yapısı

Bir LSTM ağı girdileri, çıktıları ve hafıza hücreleri arasındaki akışı sağlayan üç kapıdan oluşmaktadır.

Girdi kapısı: Hafıza güncellemesi için hangi girdi değerlerinin kullanılacağını belirlemektedir. Bu kapı girdi vektörünü ve önceki hücrenin çıktısını bir araya getirerek Sigmoid fonksiyonunu uygulamaktadır. Bu sayede, her bir girdi ögesinin ne kadarının hücredeki bilgiye katkıda bulunacağına karar verilmektedir. Sigmoid fonksiyonu, hangi girdi değerlerinin 0-1 aralığından geçeceğine karar verirken, tanh

fonksiyonu bu geçen değerlere ağırlıklar vererek önem seviyelerine göre -1 ile 1 arasında bir çıktı üretmektedir.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5.6}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5.7}$$

Unutma Kapısı: Hücrenin önceki durumdan hangi bilgilerin silineceğini belirlemektedir. Bu kararı vermek için önceki hücrenin çıktısını ve mevcut girdi vektörünü bir araya getirerek sigmoid aktivasyon fonksiyonunu uygulamaktadır. Önceki durum (h_{t-1}) ve içerik girdisi (X_t) dikkate alınarak, *Ct-1* hücre durumundaki her değer için 0 (at) ile 1 (tut) arasında bir sayı belirlenmektedir.

$$f\,t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{5.8}$$

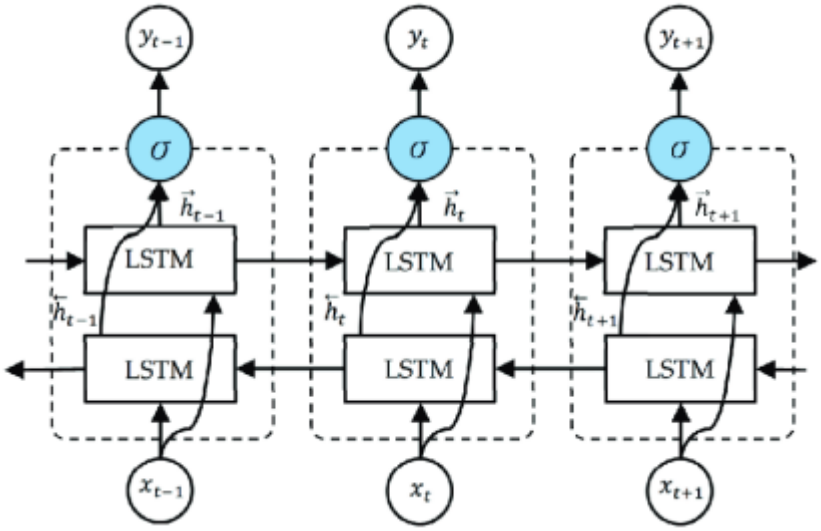
Çıktı kapısı: Verilerdeki zamansal bağımlılıkları modellemek için kullanılmaktadır. Girdi ve hafıza bloğunun çıktısını kullanarak tahmin veya sonuç üretmektedir. Bu çıktı kapısı, sigmoid aktivasyon fonksiyonu ve tanh aktivasyon fonksiyonunun bir kombinasyonunu kullanmaktadır. Sigmoid fonksiyonu, hangi girdi değerlerinin 0-1 aralığına sıkıştırılacağına karar verirken, tanh fonksiyonu, geçen değerlerin önem seviyelerine göre -1 ve 1 arasında ağırlıklar vermektedir. Bu ağırlıklar, sigmoid fonksiyonunun çıktısıyla çarpılarak nihai çıktı elde edilmektedir. Bu işlem, LSTM modelinin öğrenme sürecindeki hafıza bloklarını kullanarak geçmiş verilerdeki kalıpları tanımlamak ve gelecekteki verileri tahmin etmek için son derece etkili bir yöntemdir.

BiLSTM (çift yönlü LSTM), standart LSTM ağ mimarisinin geliştirilmiş bir versiyonudur. Verilerin hem ileri hem de geri yönde işlenmesine imkan tanmaktadır. Bu sayede, model hem geçmiş hem de gelecek durumlardaki bağlamı öğrenebilmektedir. Özellikle kelimenin anlamının kendisinden önce ve sonra gelen kelimelere bağlı olduğu dil modelleme gibi görevlerde yararlıdır. BiLSTM yapısı, bir kelimenin anlamını anlamak için önceki ve sonraki kelimelerin bağlamını dikkate almaktadır. Bu, doğal dil işleme gibi birçok uygulama için oldukça önemlidir. Ayrıca, BiLSTM modelleri, dil modelleme dışında ses tanıma, görüntü sınıflandırma ve biyomedikal veri analizi gibi birçok alanda da başarılı bir şekilde kullanılmaktadır.

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{5.9}$$

LSTM'nin gizli durumlarını temsil eden $\vec{h_t}$ ve $\overleftarrow{h_t}$, sırasıyla ileri ve geri yönde

işleyen girdileri işlemektedir. Nihai çıktı, her adımda her iki LSTM'nin çıktılarının birleştirilmesiyle hesaplanmaktadır.



Şekil 5.5 BiLSTM Yapısı

5.5 Dikkat Mekanizması

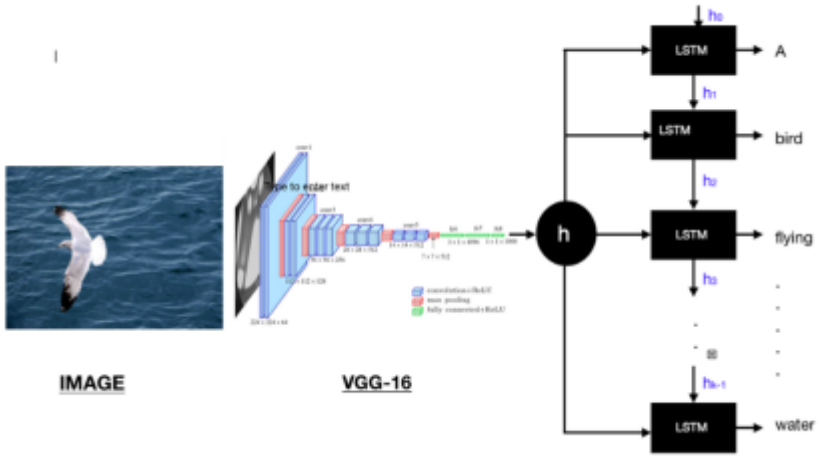
Dikkat mekanizması, son yıllarda makine öğrenmesinin ilerlemesindeki en önemli buluşlardan biri olarak kabul edilmektedir. Derin öğrenmenin uygulanış şeklini değiştirmiştir. Dikkat mekanizması, bir makine öğrenmesi modelinin dikkatini belirli bir özellik veya bölgeye yoğunlaştırmasını sağlayan insanların dikkat süreçlerini taklit ederek makine modellerinin önemli kısımlarına odaklanmasını ve ilgisiz bilgileri filtrelemesini sağlayan bir tekniktir. Bu mekanizmalar girdi olarak verilen büyük karmaşık veri kümelerindeki önemli bilgileri tanımlamak ve ayırt etmek için kullanılmaktadır. Örneğin, bir görüntü sınıflandırma modeli, her bir nesnenin özelliklerini öğrenerek nesneleri doğru bir şekilde sınıflandırmak için dikkat mekanizmasını kullanabilmektedir. Bu mekanizma, özellikle önemli olduğu düşünülen pikselleri veya özellikleri belirleyerek modelin yalnızca bu bölümlere odaklanmasını sağlamaktadır.

Dikkat mekanizmaları, makine çevirisi, konuşma tanıma, görüntü sınıflandırma ve nesne tespiti gibi birçok yapay zeka uygulamasında ve görüntü özetleme alanında oldukça önemlidir.

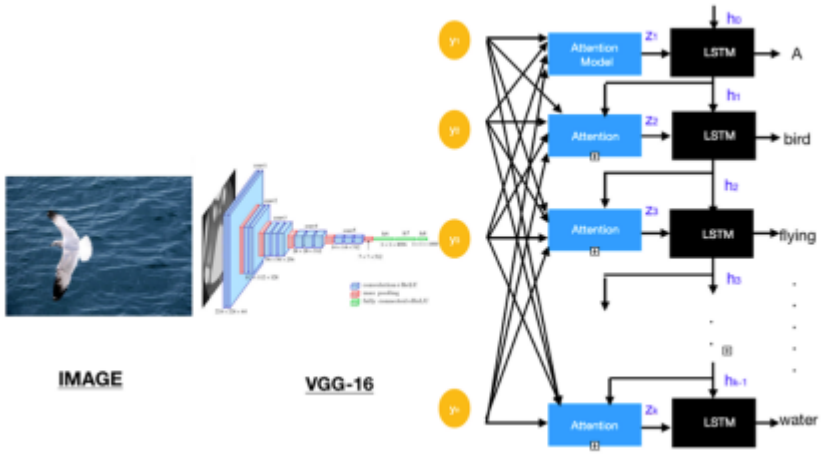
Klasik görüntü özetleme yöntemlerinde, resim önceden eğitilmiş bir kodlayıcı tarafından işlenmektedir ve ardından kod çözücü, özetlemenin her bir kelimesini üretmek için kullanılmaktadır. Ancak bu yöntem, genellikle resmin sadece belirli

bir kısmına odaklanmaktadır ve resmin bütünlüğündeki anlamı kaybetme riski taşımaktadır.

Dikkat mekanizması, resmi parçalara ayırmaktadır. Her bir parçayı kodlayıcıya (CNN) göndermektedir. Kod çözücü (LSTM, GRU, RNN) , yeni bir kelime oluştururken dikkat mekanizması resmin ilgili parçasına odaklanmaktadır. Kod çözücünün yalnızca bu parçasını kullanmaktadır. Bu sayede, özetlemeler daha doğru ve bütünsel hale gelmektedir.



Şekil 5.6 Klasik Görüntü Özetleme

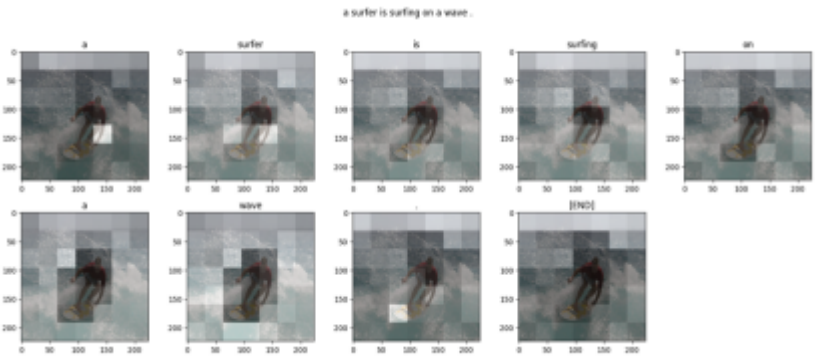


Şekil 5.7 Dikkat Mekanizmalı Görüntü Özetleme

Şekil 5.7'de, geleneksel yöntemin yanı sıra dikkat mekanizmasının da kullanıldığı görülmektedir. Bu yöntemde, bir sonraki kelime tahmin edilmeye çalışılırken, gizli durum (h_i) oluşmaktadır. Bu gizli durumu kullanarak ilgili kısmı seçen dikkat mekanizması devreye girmektedir. Dikkat mekanizmasının çıktısı olan (z_i), resmin ilgili kısmının filtrelenmiş halidir ve kod çözücü için girdi olarak kullanılmaktadır.

Sonrasında, kod çözücü yeni kelimeyi tahmin etmektedir ve gizli durum (h_{i+1}) oluşturulmaktadır.

Bu yöntem, görüntü özetleme gibi uygulamalarda oldukça başarılı sonuçlar vermektedir. Dikkat mekanizması sayesinde, model resmin sadece ilgili kısımlarını kullanarak daha bütünsel ve doğru bir özetleme yapabilmektedir. Bu da, özetlemenin daha anlamlı ve etkili olmasını sağlamaktadır.Her kelime için dikkat mekanizmasının odaklandığı bölgeler Şekil 5.8'de görülmektedir:



Şekil 5.8 Her Kelime İçin Dikkat Mekanizmasının Odaklandığı Bölgeler

5.6 Öznitelik Çıkarma

Görüntü özetleme işlemi, kelimelerin doğru sırayla tahmin edilebilmesi için temelinde derin öğrenme bulunan öznitelik çıkarma işlemine dayanmaktadır. Bu işlem, görüntüyü yapay sinir ağlarından geçirerek, program için işlemesi kolay ve daha verimli hale getirilen veriye dönüştürmektedir. Günümüzde, bu işlemi gerçekleştirmek için birçok mimari bulunmaktadır ve her biri farklı katmanlar ve boyutlardaki çıktıları kullanmaktadır. Örneğin, InceptionV3 mimarisi minimum 150x150x3 boyutunda bir girdi almaktadır ve bunları işleyerek 8x8x2048 boyutunda bir tensor vermektedir. Bu çıktı, görüntüyü 64 eş kare alana böler ve her bir karede 2048 farklı özellik bulunmasını sağlamaktadır. Şekil 5.4'te ise örnek bir görüntüye bu işlem uygulanmıştır.

Girilen görüntüyü programın anlayabileceği şekilde düzenlemek için yapılan öznitelik çıkarma işlemi oldukça önemlidir. Şekil Şekil 5.5'te niteliklerin çıkarılmasıyla elde edilen sonuçlar gösterilmiştir. Bu işlem sayesinde görüntü, derin öğrenme tabanlı bir yapıdan geçirilerek çeşitli yapay sinir ağları ile işlenebilir hale getirilmektedir. İşlenmiş görüntü, her pikselinin belirli özelliklerle donatılmış bir tensör halinde çıktı olarak verilmektedir. Bu tensör, eğitim sürecinde kullanılmak üzere saklanmaktadır. Özetle, öznitelik çıkarma işlemi, görüntülerin program için anlamlı hale getirilmesi ve

derin öğrenme modellerinin bu görüntüleri işleyebilmesi için gerekli olan özelliklerin çıkarılması işlemidir.

5.7 GRU

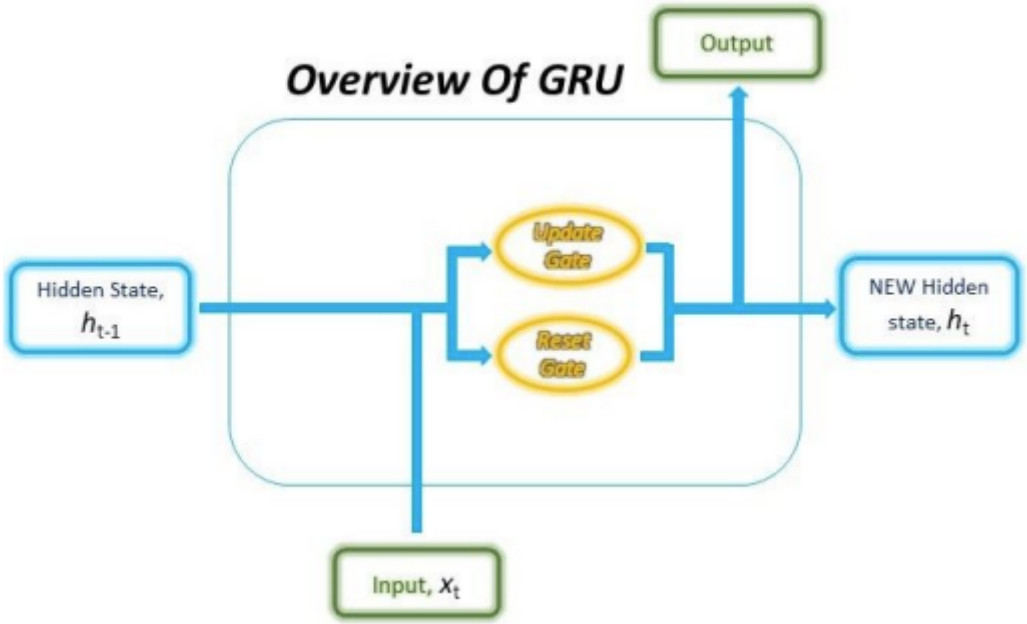
GRU(Kapılı Tekrarlayan Birim), sıralı verilerdeki uzun vadeli bağımlılıkları başarılı bir şekilde modelleyebilen, özellikle doğal dil işleme (NLP) gibi sıralı verilerin işlenmesinde kullanılan, geriye dönük bir tekrarlayan sinir ağı (RNN) mimarisidir. GRU (Kapılı Tekrarlayan Birim), 2014 yılında Cho ve 1997 yılında Hochreiter ve Schmidhuber tarafından önerilmiştir. Bu yapı, LSTM (Uzun Kısa Vadeli Hafıza) ağının varyasyonudur ve daha az parametreye sahip olması nedeniyle daha hızlı çalışmaktadır.

GRU'lar, LSTM'ler gibi uzun vadeli bağımlılıkları takip edebilme özelliğine sahiptir. Fakat LSTM'lerde üç kapı (giriş, unutma ve çıkış) bulunurken, GRU'lar yalnızca sıfırlama ve güncelleme kapılarına sahiptir. Bu kapılar önceki durumdan atılacak bilginin ne kadarının korunacağı ve o anki durumda hangi bilginin korunacağını belirlemek için kullanılmaktadır.

GRU modelinde bulunan sıfırlama kapısı, önceki önceki zaman adımından gelen bilginin ne kadarının unutulacağını kontrol etmek için kullanılmaktadır ve hesaplaması önceki gizli durum ile mevcut girdi kullanılarak yapılmaktadır. Sıfırlama kapısı, girdilerin ağırlıklı toplamı ile önceki zaman adımındaki girdilerin ağırlıklı toplamı arasındaki farkı alarak çalışmaktadır. Bu fark, geçmiş hafızanın ne kadarının kullanılacağını kontrol eder ve modelin öğrenme sürecindeki ağırlıkların güncellenmesine yardımcı olmaktadır. Güncelleme kapısı ise geçmiş durumdan korunacak bilginin miktarını belirlemek için kullanılmaktadır ve hesaplaması da önceki gizli durum ile mevcut girdi kullanılarak yapılmaktadır. Bu şekilde, hem sıfırlama hem de güncelleme kapıları, modelin öğrenme sürecindeki ağırlıkların güncellenmesine yardımcı olmaktadır ve ezberleme sorunlarını önlemektedir.

Önceki durum ve şimdiki girdinin bir kombinasyonu olan son gizli durum veya hafıza hücresi, sıfırlama ve güncelleme kapıları tarafından belirlenmektedir.

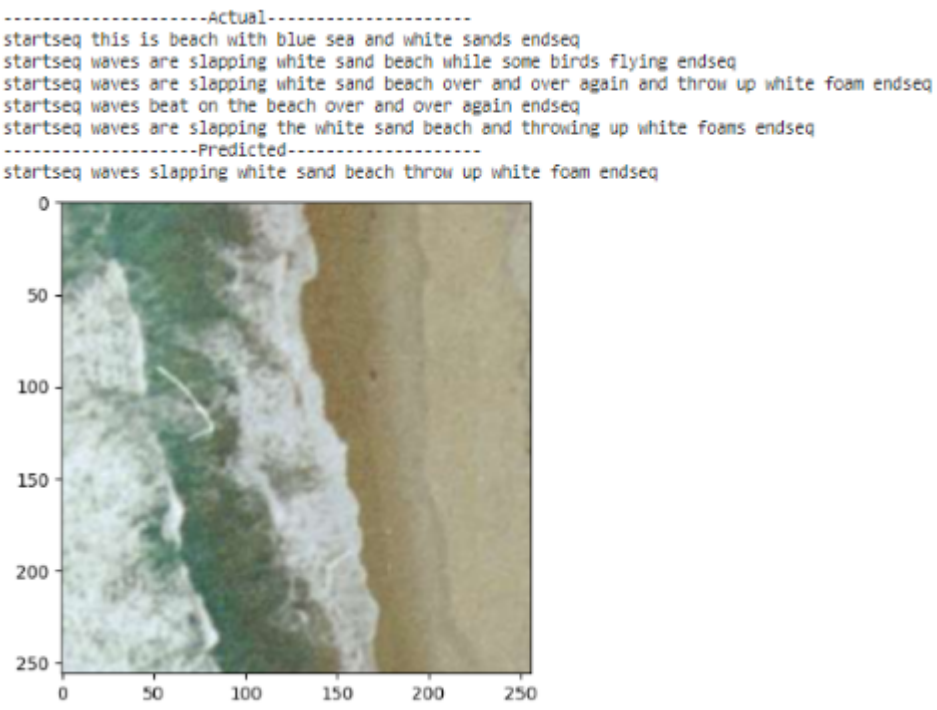
LSTM'e kıyasla daha az hafıza kapasitesi ve daha hızlı işlem yapabilme özelliğine sahip olan GRU, ancak daha uzun sekanslar içeren veri setlerinde LSTM'in daha doğru sonuçlar elde ettiği gözlemlenmiştir.[9]



Şekil 5.9 GRU Yapısı

6
Uygulama

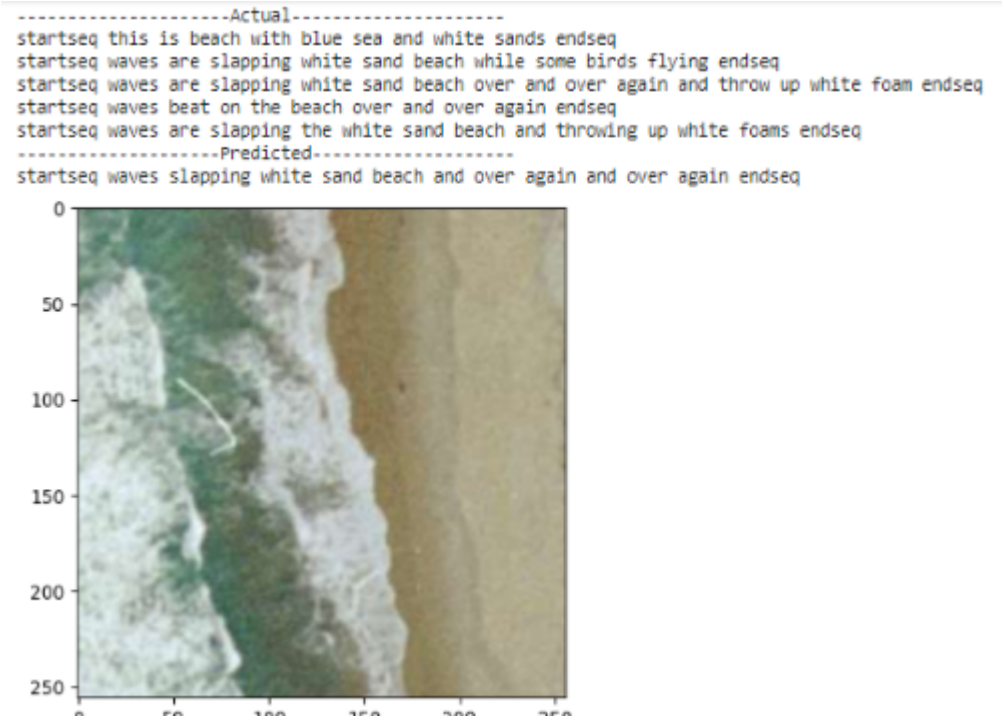
Şekil 6.1 ve Şekil 6.2'deki örnek görseller için dikkat mekanızması olan ve olmayan, VGG16 ve GRU ile oluşturulmuş model çıktıları görülmektedir.



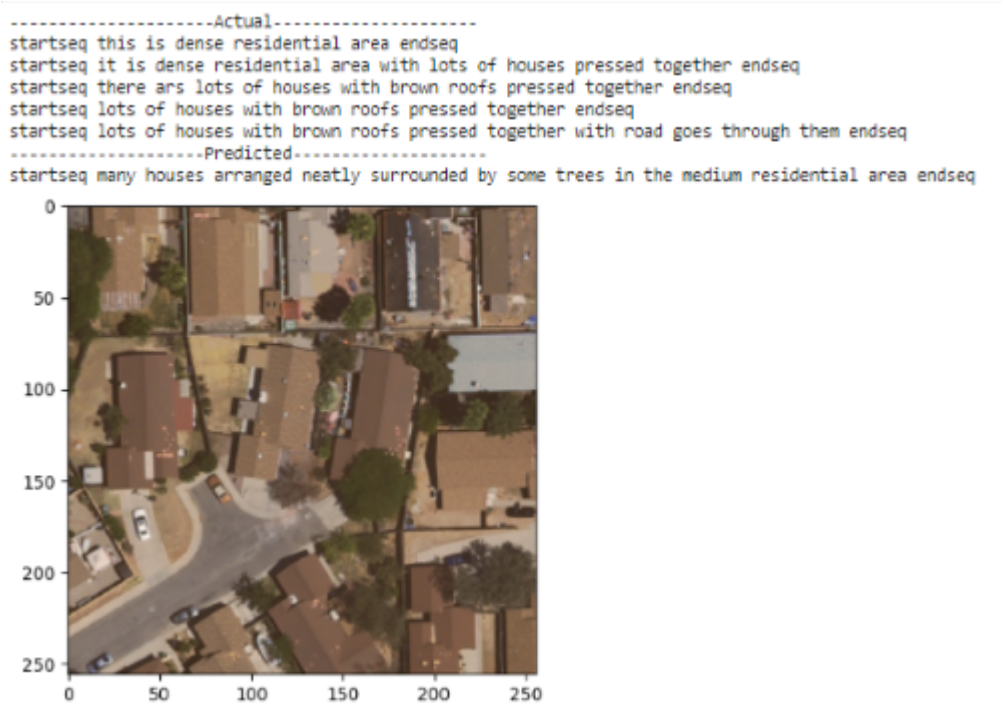
Şekil 6.1 VGG16 GRU Modeli

Görsellerde mekanın bir sahil olduğu görülmektedir. Bir tarafta sahil kumları diğer tarafta ise dalgalı bir deniz görülmektedir. Dikkat mekanızmasız modelde dalgaları ve kumları görürken dikkat mekanizmalı modelde dalgaların terkrarlı olarak çarptığını görmektedir. Modeller benzer özelliklere odaklanmalarına rağmen dikkat mekanızmalı modelin daha fazla doğru detay verdiği görülmektedir.

Şekil 6.3 ve Şekil 6.4'deki örnek görseller için dikkat mekanızması olan ve olmayan, ResNET50 ve LSTM ile oluşturulmuş model çıktıları görülmektedir.



Şekil 6.2 Dikkat Mekanizmalı VGG16 GRU Modeli



Şekil 6.3 ResNET50 LSTM Modeli

Görsellerde bir yerleşim yeri görülmektedir. Her iki model de yerleşim yeri olduğunu fark etmekte ancak "orta" kelimesiyle ifade etmektedir. Ancak böyle bir durum görülmektedir. Görselde ek olarak evler ve bitkiler görülmektedir. Her iki model de bu iki özelliği fark ettiği görülmektedir.

Şekil 6.4 Dikkat Mekanizmalı ResNET50 LSTM Modeli

7
Deneysel Sonuçlar

Aşağıdaki şekillerde 64 epoch ve 128 batch size hiperparametreleri ile hazırlanan VGG-16+LSTM+Dikkat mekanizmasının Loss ve Validation Loss değerleri gösterilmektedir. Loss ve Validation Loss değerleri şekillerde görüldüğü üzere

```
· loss: 5.6675 - val_loss: 5.4350
loss: 5.2102 - val_loss: 4.9923
loss: 4.7992 - val_loss: 4.6521
loss: 4.4710 - val_loss: 4.3425
loss: 4.1777 - val_loss: 4.0914
loss: 3.9438 - val_loss: 3.8768
loss: 3.7394 - val_loss: 3.6883
loss: 3.5539 - val_loss: 3.5166
loss: 3.3805 - val_loss: 3.3492
loss: 3.2132 - val_loss: 3.1879
loss: 3.0481 - val_loss: 3.0296
loss: 2.8863 - val_loss: 2.8724
loss: 2.7278 - val_loss: 2.7192
loss: 2.5721 - val_loss: 2.5696
loss: 2.4216 - val_loss: 2.4260
loss: 2.2766 - val_loss: 2.2924
loss: 2.1480 - val_loss: 2.1650
loss: 2.0262 - val_loss: 2.0497
loss: 1.9153 - val_loss: 1.9453
loss: 1.8075 - val_loss: 1.8479
loss: 1.7160 - val_loss: 1.7592
loss: 1.6284 - val_loss: 1.6771
loss: 1.5465 - val_loss: 1.6014
loss: 1.4724 - val_loss: 1.5330
loss: 1.4041 - val_loss: 1.4701
loss: 1.3420 - val_loss: 1.4136
loss: 1.2831 - val_loss: 1.3626
loss: 1.2330 - val_loss: 1.3152
loss: 1.1830 - val_loss: 1.2725
loss: 1.1389 - val_loss: 1.2336
loss: 1.0977 - val_loss: 1.1982
loss: 1.0619 - val_loss: 1.1677
loss: 1.0285 - val_loss: 1.1378
loss: 0.9978 - val_loss: 1.1122
loss: 0.9713 - val_loss: 1.0863
loss: 0.9453 - val_loss: 1.0628
loss: 0.9208 - val_loss: 1.0421
loss: 0.8978 - val_loss: 1.0264
loss: 0.8725 - val_loss: 1.0082
```

Şekil 7.1 Loss Val-Loss 1. Kısım

düşmektedir. Loss değerinin düşmesi modelin öğrendiğini gösterirken Validation Loss değerinin düşmesi modelin ezberleme yapmadığını göstermektedir. Model ilk başlarda 0.001 learning ile çalıştırılmıştır. Bu çalıştırmada modelin Loss değeri düşerken Validation Loss değerinin dalgalanma yaşadığı gözlenmiştir. Bu modelin ezberlediği anlamına gelir. Bu yüzden sonuçlar karmaşık çıkmıştır. Learning Rate 0.0001'e azaltılarak bu sorun ortadan kaldırılmıştır.

```
 loss: 5.6675 - val_loss: 5.4350
loss: 5.2102 - val_loss: 4.9923
loss: 4.7992 - val_loss: 4.6521
loss: 4.4710 - val_loss: 4.3425
loss: 4.1777 - val_loss: 4.0914
loss: 3.9438 - val_loss: 3.8768
loss: 3.7394 - val_loss: 3.6883
loss: 3.5539 - val_loss: 3.5166
loss: 3.3805 - val_loss: 3.3492
loss: 3.2132 - val_loss: 3.1879
loss: 3.0481 - val_loss: 3.0296
loss: 2.8863 - val_loss: 2.8724
loss: 2.7278 - val_loss: 2.7192
loss: 2.5721 - val_loss: 2.5696
loss: 2.4216 - val_loss: 2.4260
loss: 2.2766 - val_loss: 2.2924
loss: 2.1480 - val_loss: 2.1650
loss: 2.0262 - val_loss: 2.0497
loss: 1.9153 - val_loss: 1.9453
loss: 1.8075 - val_loss: 1.8479
loss: 1.7160 - val_loss: 1.7592
loss: 1.6284 - val_loss: 1.6771
loss: 1.5465 - val_loss: 1.6014
loss: 1.4724 - val_loss: 1.5330
loss: 1.4041 - val_loss: 1.4701
loss: 1.3420 - val_loss: 1.4136
loss: 1.2831 - val_loss: 1.3626
loss: 1.2330 - val_loss: 1.3152
loss: 1.1830 - val_loss: 1.2725
loss: 1.1389 - val_loss: 1.2336
loss: 1.0977 - val_loss: 1.1982
loss: 1.0619 - val_loss: 1.1677
loss: 1.0285 - val_loss: 1.1378
loss: 0.9978 - val_loss: 1.1122
loss: 0.9713 - val_loss: 1.0863
loss: 0.9453 - val_loss: 1.0628
loss: 0.9208 - val_loss: 1.0421
loss: 0.8978 - val_loss: 1.0264
loss: 0.8725 - val_loss: 1.0082
```

Şekil 7.2 Loss Val-Loss 2. Kısım

8
Performans Analizi

Aşağıdaki tabloda hazırlanan modellerin başarı sonuçları gösterilmektedir.

Tablo 8.1 Hazırlanan Modeller ve Metrik Skorları

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
VGG16+RNN	0.806135	0.701411	0.628995	0.562215	0.488393
VGG16+LSTM	0.758535	0.610692	0.485401	0.382900	0.411876
VGG16+BiLSTM	0.744734	0.607868	0.491425	0.392511	0.416078
VGG16+GRU	0.808041	0.726508	0.670042	0.615177	0.53417
ResNet50+RNN	0.830920	0.741311	0.673909	0.607746	0.530444
ResNet50+LSTM	0.794503	0.663478	0.558681	0.461408	0.465616
ResNet50+BiLSTM	0.785018	0.647081	0.542737	0.450362	0.456563
ResNet50+GRU	0.836279	0.748475	0.680604	0.615663	0.525191
VGG16+RNN+DM	0.805821	0.722590	0.665082	0.609683	0.512942
VGG16+LSTM+DM	0.827071	0.747207	0.689500	0.632323	0.538007
VGG16+BiLSTM+DM	0.833367	0.760134	0.704059	0.646394	0.499069
VGG16+GRU+DM	0.801298	0.720611	0.660276	0.600345	0.500892
ResNet50+RNN+DM	0.856392	0.786050	0.734362	0.683725	0.546313
ResNet50+LSTM+DM	0.830675	0.752609	0.696541	0.640177	0.535547
ResNet50+BiLSTM+DM	0.849758	0.772904	0.716644	0.661309	0.546588
ResNet50+GRU+DM	0.847073	0.770262	0.715349	0.660849	0.546474

Tabloyu incelediğimizde, dikkat mekanizması olan modellerin dikkat mekanizması olmayan modellere üstünlük kurduğunu çıkarımı yapılabilir. CNN modellerini karşılaştıracak olursak, en iyi Bidirectional LSTM ve GRU çıkarken onları LSTM takip etmektedir. RNN'in ise en kötü performansı veren CNN mimarisi olduğu çıkarılabilir. Modelleri karşılaştırdığımızda ResNet50 modelinin VGG16'ya göre daha iyi sonuçlar verdiği görülebilmektedir.

Yukarıdaki sonuçlardan ResNet50 ve VGG16 modelinin gayet başarılı modeller olduğunu söyleyebiliriz.

Tabloyu incelendiğinde bazı modellerin aykırı değerlerde olduğu görülüyor. Bunun sebebi ezberlemekten kaynaklanıyor olabilir.

Bakıldığında dikkat mekanizmasının gayet iyi bir performans verdiği kanısına varılabilmektedir.

Proje kapsamında, Uzaktan Algılamalı Görüntülü Özetleme için farklı modeller ve farklı mimariler denenmiştir. CNN mimarisindeki ResNet50 ve VGG16'nın, RNN mimarisinde simple RNN, LSTM, BiLSTM, GRU performansları karşılaştırıldı. Ayrıca dikkat mekanizmasının da başarıya katkısı gözlendi. 16 farklı modelin başarısı BLEU-1, BLEU-2, BLEU-3, BLEU-4, METEOR metrikleriyle ölçüldü. Bu projede bazı eklemeler ve geliştirmeler yapılarak doğal afet izleme, tarımsal verililik, şehir planlaması gibi alanlarda kullanılabilir.

9
Sonuç

Projede uzaktan algılama görüntülerinde görüntü özetleme farklı mimarilerle denenmiştir. Kodlayıcı (CNN) olarak VGG ve ResNet mimarisini kullanılmıştır. Kod çözücü kısmında ise RNN, LSTM, GRU ve BiLSTM yapıları birbirleri ile kombinlenerek modeller hazırlanmıştır. Hazırlanan modeller, dikkat mekanizması eklenen ve eklenmeyen modeller olarak ayrı bir şekilde hazırlanmıştır. Toplamda 16 model hazırlanmıştır. BLEU ve METEOR metrikleri ile değerlendirilen bu modeller farklılıkları ve güçlü oldukları yönleri ile incelenmiştir. Sonuç olarak projede, belirli geliştirilmeler ile meteorolojide, savunma sanayisinde ya da uzay endüstrisinde kullanılmak için uygundur.

Referanslar

[1] B. Zhao. "A systematic survey of remote sensing ımage captioning." (2021), [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9615088 (visited on 11/15/2021).

[2] V. Keskin. "(50 saat) python a-z™: Veri bilimi ve machine learning." (2023), [Online]. Available: https://www.udemy.com/course/python-egitimi/ (visited on 02/15/2023).

[3] D. TEAM. "Deep learning ve python: A'dan z'ye derin öğrenme (5)." (2023), [Online]. Available: https://www.udemy.com/course/deep-learning-ve-python-adan-zye-derin-ogrenme-5 (visited on 03/01/2023).

[4] A. Dabas. "Dabasajay/ımage-caption-generator." (2019), [Online]. Available: https://github.com/dabasajay/Image-Caption-Generator (visited on 11/22/2019).

[5] X. Lu, B. Wang, X. Zheng, and X. Li, "Exploring models and data for remote sensing ımage caption generation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 4, pp. 2183–2195, DOI: 10.1109/TGRS.2017.2776321.

[6] A. Gibiansky. "Convolutional neural networks." (2014), [Online]. Available: https://andrew.gibiansky.com/blog/%20machine-learning/convolutional-neural-networks/ (visited on 02/24/2014).

[7] A. Z. Karen Simonyan, "Very deep convolutional networks for large-scale ımage recognition," 2014.

[8] S. R. Kaiming He Xiangyu Zhang and J. Sun, "Deep residual learning for ımage recognition," 2015.

[9] B. Zhao, "A systematic survey of remote sensing ımage captioning," *IEEE Access 9*, pp. 154 086–154 111, 2021.

Özgeçmiş

BİRİNCİ ÜYE

İsim-Soyisim: Ömer Talha BAYSAN
Doğum Tarihi ve Yeri: 08.03.2000, Eskişehir
E-mail: talha.baysan@std.yildiz.edu.tr
Telefon: 0555 056 51 26
Staj Tecrübeleri: CRS Soft Şirketi Yazılım Departmanı

İKİNCİ ÜYE

İsim-Soyisim: Tolga SAĞLAM
Doğum Tarihi ve Yeri: 20.12.1999, İstanbul
E-mail: tolga.saglam@std.yildiz.edu.tr
Telefon: 0542 674 42 08
Staj Tecrübeleri: Kalyon Holding Yazılım Departmanı

Proje Sistem Bilgileri

Sistem ve Yazılım: Windows İşletim Sistemi, Python
Gerekli RAM: 30GB
Gerekli Disk: 6GB