



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



DERİNLEMESİNE AYRILABİLİR EVRİŞİM
VE LSTM AĞLARI İLE GÖRÜNTÜLERDEN
ANLAMSAL İFADE ÇIKARMA

Ezgisu ŞENEL

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Ekim-2023
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Ezgisu ŞENEL tarafından hazırlanan “Derinlemesine Ayrılabilir Evrişim ve LSTM Ağları ile Görüntülerden Anlamsal İfade Çıkarma” adlı tez çalışması 02/10/2023 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof. Dr. İsmail BABAOĞLU

.....

Danışman

Dr. Öğr. Üyesi Hazim İŞCAN

.....

Üye

Doç. Dr. Mehmet Akif ŞAHMAN

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mevlüt UYAN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Ezgisu ŞENEL

Tarih: 02.10.2023

ÖZET

YÜKSEK LİSANS TEZİ

DERİNLEMESİNE AYRILABİLİR EVRİŞİM VE LSTM AĞLARI İLE GÖRÜNTÜLERDEN ANLAMSAK İFADE ÇIKARMA

Ezgisu ŞENEL

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Hazim İŞCAN

2023, 55 Sayfa

Jüri
Prof. Dr. İsmail BABAOĞLU
Doç. Dr. Mehmet Akif ŞAHMAN
Dr. Öğr. Üyesi Hazim İŞCAN

Bir görüntünün içeriğini, görüntünün bize neler anlattığını cümleler kurarak doğru bir şekilde ifade etmek insan beyni için her ne kadar kolay olsa da bir bilgisayar için bu işlemi yapmak oldukça zordur. Doğru ve iyi biçimlendirilmiş cümleler oluşturmak için, dilin hem sözdizimsel hem de anlamsal olarak anlaşılması gerekir. Bu konuda karşımıza çıkan en büyük zorluk sadece görüntülerde bulunan nesneleri değil, aynı zamanda bu nesnelerin birbirleriyle ilişkisini, nasıl bir ilişki içerisinde olduklarını ifade eden bir açıklama oluşturabilmektir. Derin öğrenme yaklaşımı ile ağlar, görüntülerdeki nesneleri, yüzleri, sahneleri ve diğer anlamsal bilgileri anlamak için büyük veri kümeleri üzerinde eğitilir. Görüntülerin anlamsal analizi, otomotiv, güvenlik, video gözetimi ve tıbbi görüntüleme gibi birçok alanda uygulanabilmektedir. Bu alan, daha doğru ve karmaşık analizler sağlayan yeni derin öğrenme modelleri ve büyük veri kümeleriyle sürekli olarak gelişmekte ve ilerlemektedir. Bu çalışmada, Flickr_8k veri setinde bulunan 8000 görüntünün, Xception modeli ile özellik çıkarımı yapılmıştır. Diğer bir yandan Flickr_8k'da bulunan görüntülere ait 5 açıklamadan, LSTM ile benzersiz sözlük yapısı ortaya çıkarılmıştır. Elde edilen bu iki veri transfer öğrenme yapılan modele verilerek görüntülerin doğal cümlelere çevrilmesi sağlanmıştır.

Anahtar Kelimeler: Altyazı oluşturma, Anlamsal analiz, Bilgisayarlı görü, CNN-LSTM, Görüntüler, Keras, NLP

ABSTRACT

MS THESIS

SEMANTIC EXPRESSION EXTRACTION FROM IMAGES WITH DEPTHWISE SEPARABLE CONVOLUTION AND LSTM NETWORKS

Ezgisu ŞENEL

Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering

Advisor: Asst. Prof. Dr. Hazim İŞCAN

2023, 55 Pages

Jury
Prof. Dr. İsmail BABAOĞLU
Assoc. Prof. Dr. Mehmet Akif ŞAHMAN
Asst. Prof. Dr. Hazim İŞCAN

Although it is easy for the human brain to accurately express the content of an image and what the image tells us by using sentences, it is quite difficult for a computer to do this. To create correct and well-formed sentences, the language must be understood both syntactically and semantically. The biggest challenge we face in this regard is to create an explanation that not only expresses the objects in the images, but also the relationship of these objects to each other and how they are related. With a deep learning approach, networks are trained on large datasets to understand objects, faces, scenes and other semantic information in images. Semantic analysis of images can be applied in many fields such as automotive, security, video surveillance and medical imaging. This field is constantly evolving and advancing, with new deep learning models and large datasets providing more accurate and complex analytics. In this study, feature extraction was performed on 8000 images in the Flickr_8k dataset with the Xception model. On the other hand, a unique dictionary structure was revealed with LSTM from 5 descriptions of images found on Flickr_8k. These two data obtained were given to the transfer learning model and the images were converted into natural sentences.

Keywords: Captioning, Computer Vision, CNN-LSTM, Images, Keras, NLP, Semantic Analysis

ÖNSÖZ

Bu tez çalışmasında derin öğrenme ile görsel içeriklere ait altyazılar oluşturulmuştur.

Bu çalışma sürecinde karşılaştığım her türlü sorunun aşılmasında bana yardımcı olan, hoşgörüsü, desteği ve bilgisiyle bana yol gösteren değerli hocam ve danışmanım Dr. Öğr. Üyesi Hazim İŞCAN'a ve yüksek lisans eğitimim boyunca ve her an benden desteklerini esirgemeyen sevgili eşim Ümit ŞENEL'e teşekkürlerimi sunarım.

Ezgisu ŞENEL
KONYA-2023



İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	ix
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	3
3. MATERYAL VE YÖNTEM.....	6
3.1. Yapay Zeka (Artificial Intelligence, AI).....	6
3.1.1. Yapay Zeka Türleri	6
3.1.1.1. Dar Yapay Zeka (Artificial Narrow Intelligence, ANI)	6
3.1.1.2. Genel Yapay Zeka (Artificial General Intelligence, AGI)	6
3.1.1.3. Süper Yapay Zeka (Artificial Super Intelligence, ASI).....	7
3.1.2. Yapay Zeka Bileşenleri.....	7
3.1.2.1. Makine Öğrenmesi (Machine Learning, ML).....	7
3.1.2.2. Derin Öğrenme (Deep Learning, DL).....	8
3.1.2.3. Yapay Sinir Ağları (Artificial Neural Network, ANN)	8
3.1.2.4. Doğal Dil İşleme (Natural Language Processing, NLP).....	11
3.1.2.5. Bilgisayarlı Görü (Computer Vision, CV).....	12
3.1.3. Yapay Zeka Nasıl Çalışır?	13
3.2. Makine Öğrenmesi.....	14
3.2.1. Makine Öğrenmesi Algoritmaları	14
3.2.1.1. Gözetimli Öğrenme (Supervised Learning).....	14
3.2.1.2. Gözetimsiz Öğrenme (Unsupervised Learning)	15
3.2.1.4. Takviyeli Öğrenme	15
3.3. Derin Öğrenme	15
3.3.1. Evrişimli Sinir Ağı (Convolutional Neural Network, CNN).....	16
3.3.1.1. CNN Katmanları	17
3.3.1.2. CNN Mimarileri.....	20
3.3.2. Özyinelemeli Sinir Ağı (Recurrent Neural Network, RNN)	22
3.3.3. Uzun Kısa Süreli Bellek (Long Short Term Memory, LSTM).....	23
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	25
4.1. Kütüphane Hazırlığı.....	27
4.2. Veri Temizleme	27
4.3. Özellik Çıkarma	29
4.4. Veri Seti Yükleme	30
4.5. Kelime Dağılımı	30
4.6. Veri Oluşturma	31

4.7. Model Tanımlama.....	32
4.8. Model Eğitimi.....	32
4.8. Modelin Test Edilmesi.....	33
5. SONUÇLAR VE ÖNERİLER.....	35
5.1. Sonuçlar.....	35
5.2. Öneriler.....	42
KAYNAKLAR.....	43



SİMGELER VE KISALTMALAR

Kısaltmalar

AGI	: Genel Yapay Zeka
AI	: Yapay Zeka
ANN	: Yapay Sinir Ağları
ANI	: Dar Yapay Zeka
ASI	: Süper Yapay Zeka
BLEU	: İki Dilli Değerlendirme Alt Çalışması
CIDeR	: Konsensüse Dayalı Görüntü Açıklaması Değerlendirmesi
CNN	: Konvolüsyonel Sinir Ağı
CV	: Bilgisayarlı Görü
DL	: Derin Öğrenme
GAN	: Üretken Düşman Ağı
HAF	: Hiyerarşik Dikkat Füzyon
LSTM	: Uzun Kısa Süreli Bellek
LTM	: Uzun Süreli Bellek
ML	: Makine Öğrenmesi
NLP	: Doğal Dil İşleme
ReLu	: Rektifiye Lineer Birim
REN	: Yeniden Değerleme Ağı
RNN	: Özyinelemeli Sinir Ağı
RODe	: Bölgesel Nesne Dedektörü
RSIC	: Uzaktan Algılama Görüntü Yazısı
STM	: Kısa Süreli Bellek

1. GİRİŞ

Derin öğrenme bir makine öğrenme tekniğidir. Bu, çıktıyı tahmin etmek için belirli bir veri kümesini kullanmak üzere bir sinir ağının eğitilmesini sağlar. Hem denetimli hem de denetimsiz öğrenme kullanılarak sinir ağları eğitilebilir. Derin öğrenmede, makine öğreniminde kullanılan hesaplamalar, her seferinde bir katman yerine birden çok katmanda gerçekleştirilir. Bu, makine öğreniminin tanımlaması gereken parametreleri de bilen ve potansiyel olarak daha iyi parametrelerle değerlendirmeler yapabilen bir tekniktir. Derin öğrenme sisteminde bulunan yapay sinir ağları, beynin yapısı, şekli ve buna benzer özellikler kullanılarak oluşturulan algoritmalarlardır.

Derin öğrenmenin en zor kısımlarından biri sinir ağını eğitmektir. Neden?

- Büyük veri kümeleri gerektirir.
- Büyük miktarda bilgi işleme ve hesaplama gücü gerektirir.

Örneğin; muz ve portakal görsellerinin birbirinden ayrılması. Makine öğrenimi, insan deneyimini parametreler aracılığıyla makinelere aktarmaya çalışır. Turuncu portakal, sarı muz benzeridir. Yuvarlaksa muhtemelen portakal, kavisliyse muhtemelen muz vs. gibi pek çok parametrenin tanımlanması gerekir, ancak derin öğrenme bunu kendi başına öğrenebilir. Portakal ve muz resimleri derin öğrenme sistemine sunulurken, renk ve şeklin farklılıklarını ortaya çıkarmak için temel ayırt edici unsurlar olduğunu kabul ederek kendi kurallarını oluşturulur. Bu nedenle, temel insan becerileri gerektirmez ve kendi ayırtırma yeteneğini inşa ederek eylemlerini gerçekleştirebilir.

Bir görüntünün içeriğini, görüntünün bize neler anlattığını cümleler kurarak doğru bir şekilde ifade etmek insan beyni için her ne kadar kolay olsa da bir bilgisayar için bu işlemi yapmak oldukça zordur. Doğru ve iyi biçimlendirilmiş cümleler oluşturmak için, dilin hem sözdizimsel hem de anlamsal olarak anlaşılması gerekir.

Görme engelli kişilerin, etraflarında olan olayları, görüntüleri daha iyi anlamalarına yardımcı olması için görüntülere altıyazı oluşturmanın büyük bir etkisi olabilir.

Bu konuda karşımıza çıkan en büyük zorluk sadece görüntülerde bulunan nesneleri değil, aynı zamanda bu nesnelerin birbirleriyle ilişkisini, nasıl bir ilişki içerisinde olduklarını ifade eden bir açıklama oluşturabilmektir.

Görüntülerin anlamsal analizi, günlük hayatta birçok kullanım alanına sahiptir ve önemli bir özelliği görme engelli bireylerin görsel içeriklere erişimini artırmaktır. Günlük hayattan kullanım alanlarına örneklerden bazıları:

Görme engellilere yardımcı olma: Altyazı oluşturma, görme engelli bireylerin görsel içeriklere erişimini kolaylaştırır. Görme engelli bir kişi, sosyal medya platformlarında veya web sitelerinde paylaşılan fotoğrafların veya videoların içeriğini anlamak için otomatik olarak üretilen altyazıları kullanarak sesli olarak dinleme yapabilir. Bu, onların dijital içeriklere daha fazla katılımını ve sosyal etkileşimlerini artırır.

Görsel içeriğin arama motorlarında erişilebilir hale gelmesi: Altyazılar, görsel içeriği tarif eden metinsel bilgiler içerir. Bu, arama motorlarının görsel içeriği anlamalarını ve kullanıcılara daha doğru sonuçlar sunmalarını sağlar. Örneğin, bir arama motorunda "plajda güneşlenen insanlar" şeklinde bir arama yapıldığında, görüntülerin analizi ile oluşturulan altyazılara dayanan görseller bulunabilir.

E-ticarete ürün tanıtımı: E-ticaret platformları, ürünlerin görsellerini kullanarak satış yaparlar. Görüntülere altyazı oluşturulması, bu görsellerin daha etkili bir şekilde tanıtılmasını sağlar. Altyazılar, ürünün özelliklerini veya kullanım senaryolarını açıklayabilir, potansiyel müşterilere daha fazla bilgi sağlayabilir ve satın alma kararlarını kolaylaştırabilir.

Haberlerde görsel açıklamalar: Haber siteleri veya dergiler, haberlerin yanında görseller kullanır. Anlamsal analiz ile oluşturulan altyazılar, haberlerin içeriğini görsellerle daha iyi bir şekilde destekler ve okuyuculara daha fazla bilgi sağlar. Bu, haberleri daha anlaşılır ve ilgi çekici hale getirebilir.

Sosyal medyada görsel içerik paylaşımı: Sosyal medya platformları, kullanıcıların fotoğraf veya video paylaşmasına olanak sağlar. Altyazı oluşturma ile otomatik olarak oluşturulan altyazılar, kullanıcıların paylaşımlarını açıklamalarla destekleyerek görsel içeriğin anlamını ve etkileşimini artırır.

Bilgisayarlı görü ve doğal dil işlemenin bulunduğu bir noktada yapılan bu çalışma, resimlerin insanlar tarafından anlaşıldığı gibi yorumlanması ve bilgisayar tarafından bu görüntüye ait altyazının oluşturulmasını amaçlamaktadır.

2. KAYNAK ARAŞTIRMASI

Zakir Hossain ve ark. (2021), çalışmasında derin öğrenmeye dayalı birtakım teknikler önermiştir. Görüntülerin metinsel tanımlarını oluşturmak, bilgisayarla görme ve doğal dil işlemede önemli bir konu olmuştur. Bu teknikler, modelleri eğitmek ve test etmek için insan açıklamalı görüntüleri kullanır. Modeli eğitmek ve test etmek için hem gerçek hem de sentetik verileri kullanan bir resim yazısı yöntemi önermiştir. Sentetik görüntüler oluşturmak için Üretken Düşman Ağ (GAN) tabanlı metinden görüntüye üretici kullanmıştır.

Qi Wang ve ark. (2021), iyi oluşturulmuş bir cümle oluşturmayı amaçlayan uzaktan algılama görüntü yazısı (RSIC), son yıllarda daha fazla ilgi gördüğünden; RSIC’de, kodlayıcı ve kod çözücünün iki alt modelini içeren kodlayıcı-kod çözücü mimarisi önermiştir. Elde edilen performans iyi olmasına rağmen, kodlayıcı-kod çözücü mimarisi, açıklanabilirliği olmayan bir kara kutu modelidir. Bu dezavantajın üstesinden gelmek için, RSIC için yeni bir açıklanabilir sözcük-cümle çerçevesini önermişlerdir. Önerilen kelime-cümle çerçevesi iki bölümden oluşur: kelime çıkarıcı ve cümle oluşturucu. Kelime çıkarıcı verilen uzaktan algılama görüntüsündeki değerli kelimeleri çıkarırken, cümle oluşturucu bu kelimeleri iyi oluşturulmuş bir cümle halinde düzenler.

Komal Kumar ve ark.’nın (2019) önerdiği yöntem, mevcut resim yazısı oluşturucu sistemini daha da geliştirmek için derin öğrenme üzerinde durulmasıdır. Bölgesel Nesne Dedektörü (RODe) algılama, tanıma ve altyazı oluşturma için kullanılır. Görüntülerden altyazı elde etmek için genellikle derin öğrenme kullanılmıştır. Derin öğrenme ile gelen başarıların yanı sıra oluşan altyazıların doğruluğunda, duygularda bazı eksiklikler ortaya çıkmaktadır. Bu eksikliklerin giderilmesi için yeni ve kombinatoryal örnekler üretmek için Üretken Düşman modelleri önerilmiştir. Daha spesifik olarak, resimler için daha doğru ve anlamlı altyazılar oluşturmak için veri kodlamalarını denetimsiz bir şekilde öğrenen sinir ağları olan çeşitli otomatik kodlayıcılar önerilmiştir.

Amirian ve ark. (2019), görsel tabanlı metin üretme modelinin tanıtıldığı bir çalışmada görüntüler ve metinler bir araya getirilerek birbiri arasındaki ilişkinin daha iyi olması sağlanmıştır. Bu çalışmada modelin görüntü ve metin arasındaki ilişkiyi tam anlamıyla yakalayabilmesi için encoder-decoder mimarisi kullanılmıştır.

Xu ve ark. (2015), daha iyi ve daha odaklı altyazıların üretilmesini amaçlayan bir çalışmada görsel dikkat mekanizmalarını kullanmışlardır. Geliştirilen modelde kelime oluşturulurken hangi bölümün görüntüde önemli olduğu vurgulanmaktadır.

Lu ve ark. (2016), Visual Sentinel yani görsel bekçi olarak adlandırılan bir mekanizma kullandıkları çalışmalarında modelin ne zaman görsel bilgiye odaklanması gerektiği belirlenmektedir. Bu sayede bağlam yakalanılarak açıklamalar üretilir.

Gupta ve Malik (2015), daha ayrıntılı ve bağlamı zengin bir şekilde açıklama oluşturmaya odaklanan bir çalışmalarında Görsel Anlamsal Rol Etiketlemesi (Visual Semantic Role Labeling) konseptini tanıtmışlardır.

Wang ve ark. (2022), dil tabanlı Transformer modellerini altyazı oluşturma için uyarlanan bir çalışma yapmıştır. Görsel ve metin verilerinin daha iyi bir şekilde birleştirilmesi sağlanarak ve büyük metin verileri kullanılarak daha iyi açıklamalar elde edilmesini sağlamışlardır.

Vinyals ve ark. (2014), Altyazı oluşturma konusunda yaptıkları bir çalışmada nesne tespiti ve yerleştirme kullanılmışlardır. Model, görüntüdeki nesneleri tanıyabilmekte ve bu nesnelerin yerini ve ilişkilerini açıklamalara entegre edebilmektedir.

Mirza Muhammad Ali Baig ve ark. (2018), çalışmalarında, semantik kelime yerleştirme ve mevcut son teknoloji nesne tanımlama algoritmalarını kullanarak bilinmeyen nesneleri ve sınıfları tanımlamak için Zero-Shot Learning kavramlarını içeren bir çözüm önerilmektedir. Önerilen modelde, Novel Word Injection kullanarak önceden eğitilmiş bir altyazı oluşturma ile veri kümesinde bulunmayan nesneleri resim yazısına enjekte etmek için oluşturma çıktısı üzerinde çalışılmıştır.

Kenan Emir Ak ve ark. (2022), yaptıkları çalışmada, bir sahnede neyin değiştiğini tanımlamayı amaçlar ve her iki görevin de üretici ağlara dayandığı TIM'in ters versiyonu olarak kabul edilen CC (Change Captioning) önermişlerdir. TIM yöntemlerindeki son gelişmeler, manipüle edilmesi gereken doğru bölgelere odaklanamamaları nedeniyle giriş görüntüleri ve metinsel açıklamalar arasındaki ilişkiyi yakalamakta zorlanmaktadır. Sınıf koşullu etiketlere göre karmaşık metin verilerini anlamakta da zorlanılır. Bunun için TIM'i değişiklik altyazısı (CC) ile birleştiren ve birlikte eğitimin faydalarını kullanan bir çerçeve sunulmuştur.

Chunlei Wu ve ark. (2020), Resnet'in çok seviyeli öznetelik haritalarının hiyerarşik bir şekilde entegre edildiği, takviyeli öğrenme yöntemlerine dayalı resim altyazısı için temel olarak bir Hiyerarşik Dikkat Füzyon (HAF) modeli sunmaktadır.

Yeniden deęerleme aęı (REN), bir alt yazıdaki her kelimenin nemine gre her kelimeye farklı aęırlıklar atayarak CIDEr (Consensus-based Image Description Evaluation) puanını yeniden deęerlendirmek iin kullanırlar.



3. MATERYAL VE YÖNTEM

3.1. Yapay Zeka (Artificial Intelligence, AI)

Günümüz teknolojisinde gerçekten önemli bir yere sahip olan yapay zeka, sesleri algılama, insansı davranışlar gösterme, hareket, konuşma gibi yeteneklerin toplamı olan yazılım ve donanım sistemleridir.

3.1.1. Yapay Zeka Türleri

AI, insan zekasının nasıl kullanıldığına bağlı olarak üç şekilde incelenebilir.

3.1.1.1. Dar Yapay Zeka (Artificial Narrow Intelligence, ANI)

Bugün günlük hayatta görülen yapay zeka dar yapay zekadır. Aslında, bugün var olan tek yapay zeka türü de budur. Sınırlı işlevselliği nedeniyle zayıf kabul edilir. Dar yapay zekanın arka planında aslında tam anlamıyla bir düşünme ve kavrama yoktur. Belirli bir konuda insana yakın veya insandan daha iyi sonuçlar ortaya çıkarır. Özellikle belirli alanlara odaklandığından dolayı nedensel ilişkiler kurmaya çalışır.

- Otonom araçlar
- Sağlık alanında biyogörüntüleme, robot asistanlar
- Yüz algılama
- Siber güvenlik

Yukarıdaki alanlar dar yapay zekanın çevremizdeki uygulama alanlarıdır. Siri, Alexa ve Google asistan dar yapay zekaya örnektir.

3.1.1.2. Genel Yapay Zeka (Artificial General Intelligence, AGI)

Genel yapay zeka, yalnızca insanlar kadar karmaşık, zekanın işleyişiyle ilgili görevleri gerçekleştirebilen bir türdür. İster daha önce öğrenmiş olsun ister ilk kez deneyimlemiş olsun, öğrendiği becerileri kullanır. Kendisine atanan görevleri kendi kendini keşfetme yoluyla da tamamlayabilir. Yalnızca teorik olarak söz konusudur.

Günümüzde bir varsayım olarak bulunan genel yapay zeka uygulamaya geçildiğinde insan zekasının işleyişine, sahip olduğu yeteneklere erişmiş olacaktır. Bu

aynı zamanda diğer akıllı sistemlerin duygularını, inançlarını, ihtiyaçlarını ve düşünce süreçlerini tanıyabileceği anlamına gelir.

3.1.1.3. Süper Yapay Zeka (Artificial Super Intelligence, ASI)

Genel yapay zeka uygulamaya geçtikten sonra süper yapay zeka konuşulabilir. İnsanların anlaması mümkün olmayan tahmin etme, düşünme yeteneklerine sahip olur.

İnsan beyninin temel çalışma ilkelerine ve işlevlerine dayanan fakat insanın sahip olduğu beynin ve insan beynine ait olan düşünce sisteminden daha üstün özelliklere sahip yazılım ve algoritmaların ortaya çıkacağı tahmin edilmektedir. Süper yapay zekanın kendi donanımını oluşturabileceği gibi kendi algoritmalarını da oluşturabileceği düşünülmektedir.

Hawking "Etkili yapay zeka yaratmadaki başarı, uygarlığımızın tarihindeki en büyük olay olabilir. Ya da en kötüsü. Sadece bilmiyoruz. Bu nedenle, yapay zekâ tarafından sonsuz bir şekilde yardım edilip edilmeyeceğimizi veya onun tarafından görmezden gelinip, bir kenara atılıp atılmayacağımızı veya onun tarafından makul bir şekilde yok edilip edilmeyeceğimizi bilemeyiz" diyerek yapay zeka yaratma ve kullanma kurallarının belirlenmesine dikkat çekilmiştir (Öztürk ve Şahin, 2018).

3.1.2. Yapay Zeka Bileşenleri

3.1.2.1. Makine Öğrenmesi (Machine Learning, ML)

Makine öğrenimi, ilk olarak 1900'lerin ikinci yarısında ortaya çıkmaya başlamıştır. Bilgisayar öğrenimi ve yapay zekada örüntü tanıma çalışmalarında geliştirilen bilgisayar biliminin bir alt dalıdır. Makine öğrenimi, verileri yapılandırma ve tahminlerde bulunma işlevi olarak öğrenme yeteneğine sahip algoritmaların çalışmasını ve inşasını inceleyen bir sistemdir. Bu algoritmalar, dengeli program yönergelerini sıkı bir şekilde takip etmek yerine, girdi örneklerinden elde edilen verilere dayalı olarak tahminler ve kararlar almak için bir model oluşturarak çalışır.

Makine öğrenmesi algoritmaları amaçlarına göre 4'e ayrılmıştır. Bu kategoriler ilerleyen bölümlerde incelenecektir.

3.1.2.2. Derin Öğrenme (Deep Learning, DL)

Yapay zekanın bir alt dalı olan makine öğrenmesinin alt dalıdır. Birden çok katmandan oluşan makine öğrenimi yöntemidir.

Makine öğrenimi ile derin öğrenme arasında ortak yönler olsa da derinlemesine incelendiğinde iki teknoloji arasında ciddi farklar olduğu görülür. Örneğin, makine öğreniminde işlemleri tek bir katmanda gerçekleştirmek mümkün iken derin öğrenmede birden çok katmanı aynı anda işlemek mümkündür.

Öte yandan, makine öğreniminde daha fazla yapıya ihtiyaç vardır. Burada, verilerin bir etiketi olması çok önemlidir. Ancak derin öğrenmede algoritmaların belirli etiketleri olmamasına rağmen veri öğeleri adına ilişkiler belirlenebilir.

Sahip olunan verinin hacmi ne kadar geniş olursa, derin öğrenme o kadar mükemmel olur. İnsanlar için bazı görevler oluşturmak basittir, ancak bilgisayarlar için aynı şey söylenemez. Bilgi soyutlaması, görevleri karmaşıklaştıran katmanlı sinir ağlarında giderek daha önemli hale gelmektedir. Derin öğrenmede, bilgi nispeten daha soyut yöntemler kullanılarak işlenmelidir (Eker ve Duru, 2021).

3.1.2.3. Yapay Sinir Ağları (Artificial Neural Network, ANN)

Biyolojik sistemlerde öğrenme, nöronlar arasındaki sinaptik bağlantıları düzenleyerek gerçekleşir. Başka bir deyişle, insanlar doğumdan itibaren bir öğrenme süreci içindedirler. Bu süreçte beyin sürekli bir büyüme sergiler. Yaşadığı süre boyunca elde edilen deneyimler ile iki nöronun bağlantı noktaları olan sinaptik bağlantılar ayarlanır ve hatta yeni bağlantılar oluşur. Öğrenme bu şekilde gerçekleşir.

Yapay Sinir Ağı da karmaşık problemleri çözmek için insan beynindeki sinir hücrelerinden model alınarak oluşturulan sayısal sinir ağı modelidir.

Yapay sinir ağlarının genel yapısı aşağıdaki gibidir.

Girdi katmanı: Dış dünyadan gelen girdilerin yapay sinir ağına giriş yaptığı katmandır. Bu katmanda dışarıdan gelen girdi kadar hücre bulunmasa bile, girdiler herhangi bir işlem görmeden alt katmanlara iletilir.

Ara (gizli) katman(lar): Girdi katmanının çıktısı bu katmanın girdisidir. Ara katmanın sayısı ağlar arasında değişiklik gösterebilir. Bazı sinir ağlarında orta katman bulunmazken, bazı sinir ağlarında çok sayıda orta katman bulunur. Bu katmanlarda bulunan hücre sayısı girdi ve çıktı sayılarına bağlı değildir. Gizli katmanlardaki

hücrelerin sayısının artması hesaplama karmaşıklığını ve süresini artırsa da yapay sinir ağlarının daha karmaşık problemlerin çözümünde kullanılmasına da olanak sağlar.

Çıkış Katmanı: Ara katmanın çıktısı bu katmana girdi olarak gelir ve bu bilgiler işlenerek ağ sonuçları üretilir. Bu katmanın sonucu ise dış dünyaya gönderilecek olan son çıktıdır. Sonuç kullanılarak yapay sinir ağı için yeni ağırlıklar hesaplanıyorsa, bunlara geri beslemeli ağlar denir (Şeker ve ark., 2017).

Aktivasyon Fonksiyonu:

- Bu fonksiyon, hücrenin net girdisini işler ve hücrenin bu girdiye yanıt olarak üretmesi gereken çıktıyı belirler,
- Aktivasyon fonksiyonlarının lineer olmaması, yapay sinir ağlarının da doğrusal olmadığı anlamına gelir,
- Aktivasyon fonksiyonunun türevinin hesaplanmasının karmaşık olmayanlardan seçilmesi de göz önünde bulundurulması gereken bir noktadır. Geribesleme ağlarında bu fonksiyonların türevi de kullanıldığından dolayı türevi daha kolay hesaplanabilen bir işlev seçilerek hesaplamada yavaşlamanın önüne geçilir,
- Günümüzde en fazla kullanılan "Çok Katmanlı Perceptron", aktivasyon işlevi olarak genellikle "sigmoid işlevi" kullanılmaktadır, (Ser ve Bati, 2019).

Yapay sinir ağlarının sınıflandırılması:

- Yapılarına Göre:
 - İleri Beslemeli Ağlar
 - Geri Beslemeli Ağlar
- Öğrenme Algoritmalarına Göre
 - Danışmalı Öğrenme
 - Danışmasız Öğrenme
 - Destekleyici Öğrenme
- Öğrenme Zamanına Göre
 - Statik Öğrenme
 - Dinamik Öğrenme

İleri beslemeli ağlar:

İleri beslemeli ağlarda, nöronların baştan sona kadar düzenli katmanları vardır. Katmanlar arası sadece bir bağlantı vardır. Yapay sinir ağına gelen bilgiler sırasıyla orta

katmanlarda ve çıkış katmanında işlendikten sonra giriş katmanına geçerek daha sonra dış dünyaya çıkar.

- Tek yönlü bir bilgi akışı mevcuttur.
- İleri beslemeli ağ modelinde giriş katmanından elde edilen çıktılar gizli katmana iletilir.
- Ara ve çıkış katmanlarından, bilginin işlenmesi sonucu ile çıkış değeri hesaplanır.

Geri beslemeli ağlar:

Geri beslemeli sinir ağlarında, ileri beslemeli ağlardan farklı olarak, bir hücrenin verdiği sonuç bir sonraki hücrenin katmanına giriş verisi olarak sağlanmaz. Ayrıca bir önceki katmandaki herhangi bir hücreye veya kendi katmanına girdi olarak bağlanabilir.

- Geri beslemeli sinir ağı, çıkış katmanından ve ara katmandan gelen çıktıların önceki birimlere ya da gizli katmanlara geri yayıldığı bir ağ yapısıdır. Böylece, bilgi hem ileri hem de geri yönlü işlenir (Backpropagation).
- Bu tür yapay sinir ağları dinamik belleğe sahip olmasından dolayı çıktı hem mevcut girişi hem de önceki girişi anında gösterir. Bu nedenle, tahmin uygulamaları için özellikle uygundur.
- Geri bildirim ağları, farklı türdeki zaman serilerini tahmin etmede çok başarılı olmuştur. Hopfield, SOM (Self-Organizing Map), Elman ve Jordan ağları örnek olarak verilebilir.

Denetimli öğrenme:

Denetimli öğrenme sırasında sinir ağına sağlanan girdi verileri için çıkış değerleri de sağlanır. Gelen giriş verileri ile çıktıları üretmek için ağ, kendi ağırlıklarını güncelleyerek ilerler. Ağ çıkışı ile elde edilmesi istenen çıkış arasındaki hata değeri hesaplanır ve yeni ağ ağırlığı hesaplanan bu değere göre sıralanır.

Denetimsiz öğrenme:

Öğrenme sürecinde ağa yalnızca örnek girdiler iletilir. Elde edilmesi istenen çıktı değerleri verilmez. Giriş bölümünde verilen bilgilere göre ağ, her sürümü diğerinden sınıflandırmak için kendi kurallarını oluşturur. Aynı özelliklere sahip dokuları ayırmak için ağ bağlantı ağırlıklarını güncelleyerek öğrenme sürecini tamamlar.

Destekleyici öğrenme:

Bu yöntemde, ağın her yinelemesinin sonucunun başarı durumu hakkında bilgi verilir. Ağ, bu bilgilere göre tekrardan düzenlenir. Bu şekilde ağ, bir girdi dizisi ile hem öğrenme hem de çıkarım yapmaya devam eder.

Statik öğrenme:

Sinir ağı, kullanımdan önce eğitilen statik bir öğrenme kuralıyla çalışır. Bu eğitimin tamamlanmasının ardından istenildiği gibi kullanılabilir. Fakat bu ağı kullanırken ağırlıklarda herhangi bir değişiklik yoktur.

Dinamik öğrenme:

Dinamik öğrenmenin temel mantığı ise sinir ağlarının çalışırken öğreneceğini öngörerek tasarlanır. Ağın eğitim aşamasının tamamlanmasının ardından sonraki kullanımlarda çıktının onayına göre ağırlıkları değiştirerek çalışmaya devam eder (Şeker ve ark., 2017).

3.1.2.4. Doğal Dil İşleme (Natural Language Processing, NLP)

NLP yapay zekanın en önemli konularından birisidir. Doğal dil işleme, bir bilgisayarın konuşulan dili anlama, anladığını yorumlama ve hatta sıfırdan kendi cümlelerini oluşturma yeteneği olarak adlandırılır. Bilgisayar bilimi (özellikle yapay zeka ve makine öğrenimi) ile dilbilimi birleştiren bir çalışmadır.

Bankacılık sitelerinde konuşan chatbot'lar, telefonda asistana verilen komutlar, Google/Microsoft Translate ile oluşturulan çeviriler, mesaj yazarken telefonun bir sonraki kelimeyi tahmin etmesi gibi teknolojiler tamamen doğal dil işleme sürecidir. NLP, makine öğrenimi algoritmaları ile birlikte kullanıldığında, görevleri kendi kendine yapmayı öğrenen ve deneyimleyerek gelişen sistemler oluşturur (Küçük ve Arıcı, 2018).

Doğal dil işlemede sözdizimsel analiz ve anlamsal analiz olmak üzere iki teknik uygulanır.

Sözdizimsel analiz, kelime sırasını ve kelimeler arasındaki ilişkileri belirlemek için temel gramer kurallarını kullanarak metni analiz eder.

- Simgeleştirme; simge isimleri verilmiş olan cümle ya da sözcüklerden oluşan daha minik parçalara bölünmesi ana metnin işlenmesini kolaylaştırır.
- Konuşma etiketleme; kelimelerin etiketlenerek (isim, fiil, zarf gibi) anlamının anlaşılmasını sağlar.

- Lemma ve kök; daha kolay ayrıştırma için bir sözcüğü tanıdık temel biçimine indirgeme.

Semantik analiz, metnin anlamını çıkarmak için kullanılan doğal dil işleme tekniğidir. Başlangıçta, kelime anlamları kontrol edilir. Ardından kelimelerin kombinasyonlarını ve bağlam içindeki anlamlarını kontrol eder. Semantik analizin bazı temel görevleri şöyledir:

- Kelimelerin anlamındaki net olmayan durumları ortadan kaldırma; net olarak belirlenen bir bağlamda kullanılan bir kelimenin anlamını belirlemeye çalışmaktır.
- Bağlantı kurma; mekanlar, insanlar vb. gibi varlıkların nasıl olduğunu metin üzerinde anlamaya çalışmak.
- Kelimeleri çıkarma; anlam değeri olmayan sık kullanılan kelimeleri kaldırma.

Doğal dil işlemenin bazı kullanımları:

Duygu analizi: Metindeki duyguları tanımlamak ve yorumları olumlu, olumsuz veya nötr olarak sınıflandırmak.

Dil çevirisi: Farklı dillerde iletişim kurmalarına izin vererek işletmelerin küresel iletişimlerini geliştirmelerine veya yeni pazarlara girmelerine yardımcı olmak.

Metin çıkarma: Metinden önceden tanımlanmış bilgileri ve ilgili anahtar kelimeleri çıkarmaya izin vermek.

Konu sınıflandırması: Yapılandırılmamış metni kategoriler halinde düzenlemeye yardımcı olmak (Yüksel ve Karabıyık, 2022).

3.1.2.5. Bilgisayarlı Görü (Computer Vision, CV)

Kısaca yapay görme denilebilir. Temel olarak bilgisayarla görme, insan beyninin görme konusunda yapabildiğini makinelerle yaptırmaya çalışmaktır. Ana görevi nesneleri tanımdır. Nesneleri tanımlar ve gruplandırır. Amaç, dijital bir görüntünün ya da bir videonun içeriğini anlamaktır. Makinelerin görüntülerden nesneleri, metinleri veya modelleri çıkarmasına izin verir.

Doğal görmede insan beyninde gerçekleşen işlemler oldukça karmaşıktır. Dolayısıyla gören ve gördüğünü anlamlandıran makinelerin bir kamera ve belirli algoritmalar ile ortaya çıkması sanıldığından daha zordur. Bu alanda uzun süredir

alıřmalar, arařtırmalar yapılmasına raėmen henüz tam anlamıyla insan gibi gren bir makine yapılmamıřtır.

Diėer bir sebep ise resim dnyasının ok renkli ve karmařık olmasıdır. Birden fazla rtřen nesne arasında ayırım yapmak insan beyni iin kolaydır, ancak makineler iin tam tersi olabilir. Konum, ıřıėın geliři, nnde veya yanında bir engel olup olmadığı gibi sayısız durum vardır ve bir nesneyi tanımak iin bir algoritmanın bu sonsuz durumları analiz etmesi ok zor olabilir.

Bilgisayar grs ok fazla veri gerektirir. Farklılıkları ayırt edene ve sonunda grnty tanıyana kadar veri analizini tekrarlar. rneėin, bir bilgisayara bitkileri tanımayı ėretmek, farklılıklarını ėretmek ve zellikle bitki trlerini tanımak iin birok bitki resminin ve bitki ile ilgili grntlerin girdi olarak verilmesi gerekir.

Grntler zerinde alıřmaya bařlandığında derin ėrenme algoritmalarını kullanmak gerekir. Bu algoritmalar sayesinde grntler zerindeki nesneler birbirinden ayırt edilebilir (nay, 2021).

3.1.3. Yapay Zeka Nasıl alıřır?

Yapay zeka, insan zekasını taklit ettiėinden dolayı insan beyninin alıřtığı gibi alıřır. Byk miktarda veriyi analiz eder, korelasyonlar, modeller oluřturur ve sonuları belirli durumlar hakkında tutarlı tahminler yapmak iin kullanır. rneėin, bir chatbot, gerek bir insan gibi iletiřim kurmayı ėrenmek iin oluřturduėumuz modelleri kullanabilir. Benzer řekilde milyonlarca rneėi tarayarak grntlerdeki nesneleri nasıl tanımlayacaėını ėrenebilir. Yapay zeka sistemleri  biliřsel yeteneėe dayanmaktadır.

ėrenme sreci:

Byk miktarda veri toplanır ve bunların nasıl kullanılabilir bilgiye dnřtrleceėine dair kurallar oluřturulur. Bu kurallara algoritma denir. Yapay zekaya belirli bir grevi nasıl tamamlayacaėına dair adım adım talimatlar verir.

Muhakeme sreci:

Yapay zeka, istenen bir sonuca ulařmak iin doėru algoritmanın nasıl seileceėini gsterir.

Kendi kendini dzeltme sreci:

Bu devam eden bir sretir. Algoritmalar mmkn olan en doėru sonuları vermek iin srekli geliřtirilmektedir. Zamanla yapay zeka daha tutarlı sonular saėlayacaktır (ztrk ve řahin, 2018).

3.2. Makine Öğrenmesi

Makine öğrenimi, bilgisayarların programlama olmadan çalışmasını, verileri keşfetmesini, kalıpları tanımasını ve büyük verilerdeki değişkenler arasındaki ilişkileri öğrenmesini sağlayan algoritmaların ve istatistiksel modellerin bilimsel olarak incelenmesidir.

Algoritmalar, makine öğreniminin temelini oluşturur. Farklı makine öğrenimi türleri vardır çünkü kullandıkları algoritmalar farklı yapılara sahiptir. Makine öğreniminin temelini oluşturan ve makine öğreniminin türlerini ortaya koyan algoritmalar bu bölümde incelenecektir.

Makine öğrenimi süreci; problemin, neyin tahmin edilmesi gerektiği ve bu tahminleri yapmak için hangi gözlemlerin gerekli olduğu açısından tanımlanması ile başlar. Ardından bir veri seti oluşturmak için yapılandırılmış verileri veya yapılandırılmamış verileri toplamak gerekir. Daha sonra makine öğrenimine uygun veriler hazırlanır ve verilerde anlamlı değişkenler oluşturulur. Verilerin hazırlığından sonra yapılacak olan önemli bir işlem ise tahmin için bir model seçmektir. Sorunu en iyi tanımlayan ve verilere uyan model seçilir. Eldeki veriler, modelin çıktıları tahmin etme yeteneğini kademeli olarak geliştirir. Veri seti üç bölüme ayrılır. Bunlar; eğitim veri seti, doğrulama veri seti ve test veri setidir. Eğitim ve doğrulama verileri model eğitiminde, test verileri ise model doğrulamada kullanılır. Ardından modelin eğitim için hiç kullanılmamış veriler üzerinde test edilmesi gerekir. İyileştirme ve yorumlama için eğitim sonuçları kullanılır. Elde edilen sonuçlar değerlendirdikten sonra, sonuçların daha da iyileştirilip iyileştirilemeyeceğini görmek önemlidir. En iyi sonuçları veren uygun parametreler ayarlanarak sonuçlar iyileştirilir. Son olarak ise modelin bilmediği veriler kullanılarak tahminler yapılır.

3.2.1. Makine Öğrenmesi Algoritmaları

3.2.1.1. Gözetimli Öğrenme (Supervised Learning)

Denetimli öğrenme boyunca ağa iletilen girdi verileri için çıktı sonuçları da belirtilir. Ağ, verilen girdiye göre beklenen çıktıyı üretmek için ağırlıkları yeniden hesaplar ve günceller. Gerçek çıktılar ile beklenen çıktılar arasındaki hata hesaplanır ve yeni ağırlıklar bu hata oranına göre yeniden güncellenir.

Denetimli öğrenmenin en kötü yanı, eğitim verilerini oluşturmaktır. Eğitim verileri, makine öğrenimi tekniklerini kullanarak bir işlev oluşturmak için kullanılır. Bu işlev, yeni gelen verileri tahmin etmeye çalışır. Bu eğitim verilerini hazırlamak, denetimli öğrenmenin en çok zaman alan kısmıdır. Kötü hazırlanmış eğitim verileri çok zayıf tahminlere yol açar. Bu eğitim verileri oluşturulurken iki veya üç kişinin rehberlik etmesi avantajlıdır.

3.2.1.2. Gözetimsiz Öğrenme (Unsupervised Learning)

Denetimsiz öğrenmede, eğitim sırasında ağa yalnızca örnek giriş verileri verilir. İstenen çıktı değerleri verilmez. Giriş verilerine göre ağ, her örneği birbirine göre sınıflandıran kendi kurallarını oluşturur. Ağ bağlantı ağırlıkları aynı özelliklere sahip başka bir doku üzerine yerleştirilerek öğrenme işlemi tamamlanır.

Denetimsiz öğrenmede eğitim için ayrılan bir veri seti bulunmaz. Bu bölümde kullanılan algoritmalar, elde bulunan verileri gruplandırmaya ve daha sonra gelen verileri en uygun gruba atamaya çalışır. Denetimsiz öğrenmenin uygulanmasının kolay olmasının sebebi ise eğitim verisinin olmayışındır. Fakat bazı kolay olmayan sorunlarda çok iyi sonuç vermeyebilir.

3.2.1.4. Takviyeli Öğrenme

Bu yöntemde, ağın her yinelemesinin sonucunun başarı durumu hakkında bilgi verilir. Ağ, bu bilgilere göre tekrardan düzenlenir. Bu şekilde ağ, bir girdi dizisi ile hem öğrenme hem de çıkarım yapmaya devam eder (Şeker ve ark., 2017).

3.3. Derin Öğrenme

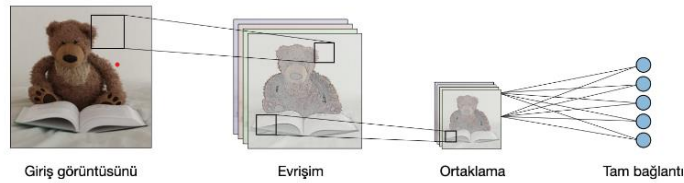
Makinelerin, insan beyninin işleyişini ve yapısını kopyalayarak öğrenme ve kavrama yeteneklerine insan denetimi olmadan sahip olmasıdır. Kısacası düşünüen ve hisseden makinelere doğru giden bir yoldur.

Makine öğrenmesinin bir alt dalı olan derin öğrenme, makine öğrenmesi gibi öğrenmesi sınırlı değildir. Derin öğrenme sistemlerinde sonuçlar, girdi verilerinin boyutu ve tutarlılığına bağlı olarak sürekli iyileşir.

İnsan zihni ile büyük veriyi işlemek oldukça zordur, hatta imkansızdır. Yeterli insan zihni elde edilmiş olsa dahi bu veri setlerinden özellikleri çıkarmak oldukça uzun zamanlar alır. Derin öğrenme bu özellik çıkarma, sınıflandırma sürecini çok iyi bir derecede basitleştirir. Derin öğrenme sayesinde yapay zeka sistemleri insan denetimi olmadan kendi kendini eğitebilir ve geliştirebilir. Bazı derin öğrenme mimarileri aşağıda maddeler halinde açıklanmıştır.

3.3.1. Evrişimli Sinir Ağı (Convolutional Neural Network, CNN)

Geleneksel bir Konvolüsyonel Sinir Ağı mimarisi genellikle üç katmandan oluşur. Bunlar; Evrişim katmanı (CONV), Ortaklama (POOL), Tam Bağlantı (Fully Connected, FC) katmanlarıdır (Şekil 3.1.). Fotoğraf ve video görüntüleri, metin, ses benzeri giriş verileri için bazı derin öğrenme algoritmaları kullanılarak öğrenme gerçekleşir. Normal sinir ağlarının aksine, CNN'ler girdi verilerindeki özellikleri tanımlamak için filtreler veya çekirdekler adı verilen küçük matrisler kullanır. Bu filtreler giriş verilerini kaydırır ve her noktada bir çıktı üretir. Bu şekilde ağ, girdi verilerinden özellikleri öğrenir ve bu özellikleri daha üst düzey özelliklerde birleştirir.



Şekil 3.1. Evrişimli Sinir Ağı (Evrişimli Sinir Ağı, 2018)

Bilgisayar herhangi bir görüntüyü algıladığında (bir görüntüyü girdi olarak aldığı), bir dizi piksel değeri görür. Görüntünün boyutuna ve piksellerine bağlı olarak belli boyutlarda bir sayı dizisi görüntülenecektir. Netleştirmek için JPG formatında renkli bir görselin olduğunu ve boyutunun 480x480 olduğunu varsayalım. Bu noktadaki piksel yoğunluğunu temsil eden dizi 480x480x3'tür. Bu sayıların görüntü sınıflandırması için bir anlamı yoktur, ancak bilgisayar tarafından kullanılabilen girişlerdir. Buradaki amaç, bu sayı kümesi verildiğinde, bilgisayarın görüntünün herhangi bir sınıfa ait olma olasılığını temsil eden bir sayı döndürmesi gerektiğidir.

Bilgisayarların herhangi bir görüntüyü tanıyabilmesini ve bir tavşanı tavşan ya da fili fil yapan benzersiz özellikleri bulabilmesi istenir. Bu, zihnimize ve

bilinçaltımızda gerçekleşen doğal bir süreçtir. Bir filin resmine bakılıp ardından görüntü üzerinde hortum veya dört ayak gibi tanınabilir özelliklere sahip olup olmadığına göre sınıflandırılabilir. Bu özelliklere benzer olarak, bilgisayarlar görüntü sınıflandırmasını daha soyut kavramlara dayandırabilir. Bunlar kenarlar ve eğriler gibi düşük seviyeli özelliklerdir. Ve evrişimli katmanlar ile bu soyut özelliklere bakarak sınıflandırma yapılabilir (Evrişimli Sinir Ağı, 2018).

3.3.1.1. CNN Katmanları

Convolutional Layer: Bu katman, CNN'in temel taşlarındandır. Görüntü özelliğini tanımak üzerine görevlidir. Görüntüden farklı seviyelerde (düşük, yüksek) özellikleri çıkarmak için görüntüye birkaç filtre uygulanır. Uyguladığı filtre, örneğin kenar algılama için bir filtre olabileceği gibi birden fazla boyutu vardır ve piksellerin değeri vardır (5x5x3 gibi). Burada 5, matrisin yüksekliğini ve genişliğini, 3 ise matrisin derinliğini gösterir. Bu filtrenin nasıl uygulandığına bakılacak olunursa;

Anlaşılır olması adına tek bir kanal için filtre işlemi yapılır. Görüntünün 1'ler ve 0'lar içeren 5x5'lik bir görüntü olduğu görülür (Şekil 3.2.). 3x3 boyutunda bir filtre oluşturulur (Şekil 3.3.).

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

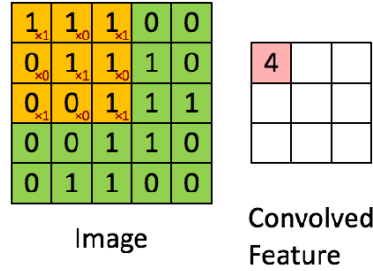
Şekil 3.2. Örnek resim

1	0	1
0	1	0
1	0	1

Şekil 3.3. Filtre

İlk olarak filtre, görüntünün sol üst köşesine yerleştirilir. Bu durumda, iki matrisin (görüntü ve filtre matrisleri) indekslerine çarpım işlemi uygulanır. Daha sonra tüm sonuçlar toplanır ve çıkış matrisi elde edilir. Ardından, bu filtre bir piksel sağa taşınarak bu çarpma işlemi yeniden gerçekleştirilir. Birinci satır tamamlandığında ikinci satıra geçilir ve işlem yeniden uygulanır. Tüm işlemler tamamlandıktan sonra bir çıktı matrisi elde edilir. Burada 3x3 filtre, 5x5 görüntü matrisinde yatay ve dikey olarak 3

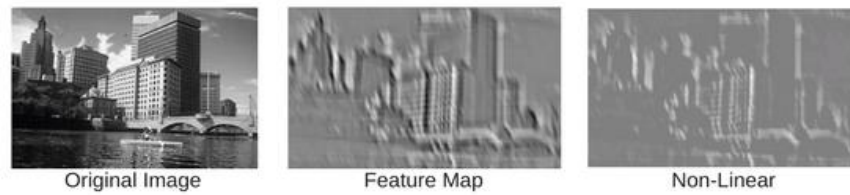
kez hareket eder, dolayısıyla çıkış matrisi 3×3 olur. Görüntü 6×4 ve filtre 3×3 ise çıkış matrisi 4×2 olacaktır (Şekil 3.4.). Bu çıktı matrisine genellikle özellik haritası denir. Yani, filtreyi görüntü üzerinde hareket ettirir ve özellikleri algılamak için basit matris çarpımını kullanır. Birden çok özelliği algılamak için genellikle birden çok filtre kullanılır. CNN ağlarının birden çok evrişimli katmanı vardır.



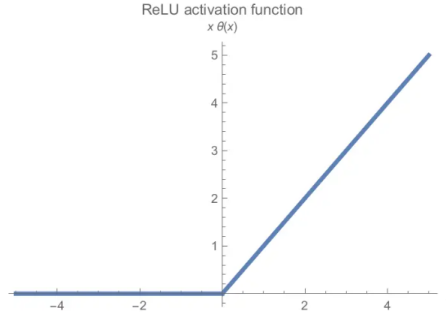
Şekil 3.4. Filtre uygulaması

Non-Linearity Layer: Doğrusal olmayan katmanlar genellikle tüm evrişimli katmanlardan sonra yerleştirilir. Bu katman, etkinleştirme işlevlerinden birini kullandığı için etkinleştirme katmanı olarak adlandırılır. Daha önce lineer olmayan fonksiyonlar (Sigmoid, Tahn) kullanılırken, sinir ağı öğrenme hızı açısından en iyi sonuçları verdiği için artık komütasyon fonksiyonu (ReLU) kullanılmaktadır.

Özellik haritasına (Şekil 3.5.) ReLu fonksiyonunu uygulamak, yukarıdaki gibi sonuç verir (Şekil 3.6.). Özellik haritasının siyah seviyesi negatiftir. Relu fonksiyonunun uygulanmasından sonra siyah düzeyi kaldırılır ve 0 ile değiştirilir.



Şekil 3.5. Özellik çıkarımı



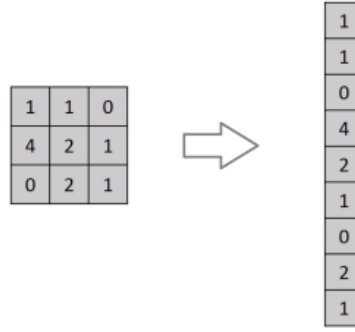
Şekil 3.6. ReLu Fonksiyonu

Pooling Layer: Bu katman genellikle bir CNN'de birbirini izleyen evrişimli katmanlar arasına eklenir. Bu katmanın görevi, filtre kaydırma boyutunu ve ağıdaki parametre ve hesaplama sayısını güncellemektir. Bu şekilde ağıdaki uyumsuzluklar kontrol edilir. Birçok havuzlama işlemi vardır, ancak en yaygın olanı maksimum havuzlamadır (Şekil 3.7.). Bu filtre aşağıdaki (4×4) görselde görülebilir. Resimden de görüleceği gibi, filtreler kapladıkları alan içinde en büyük sayıyı seçer. Bunun sebebi sinir ağının doğru karar verebilmesi için gerekli olan bilgiyi içeren küçük çıktılar kullanılmasıdır.

Tip	Maksimum ortaklama	Ortalama ortaklama
Amaç	Her ortaklama işlemi, geçerli matrisin maksimum değerini seçer	Her ortaklama işlemi, geçerli matrisin değerlerinin ortalaması alır
Görsel Açıklama		
Açıklama	- Algılanan özellikleri korur - En çok kullanılan	- Boyut azaltarak örnekleme - Öznitelik haritası - LeNet'te kullanılmış

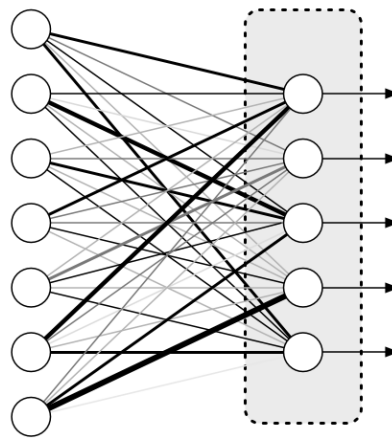
Şekil 3.7. Ortaklama türleri (Evrişimli Sinir Ağı, 2018)

Flattening Layer: Bu katmanın tek görevi, tam bağlantılı katmanın girdisi için verileri uygun hale getirmektir. Genel olarak, sinir ağı girdi verileri tek boyutlu dizilerden gelir. Bu sinir ağı için veriler, evrişim ve havuzlama katmanlarında oluşan matrislerin tek boyutlu bir diziye dönüştürülmesiyle oluşur (Şekil 3.8.).



Şekil 3.8. Düzleştirme katmanı

Fully Connected Layer: Tamamen bağlı katman, havuzlama katmanını takip eder ve evrişimli bir sinir ağının son aşamasıdır. Havuzlama katmanının ortaya çıkan çıktısı, bu katmanda tek boyutlu bir vektöre dönüştürülür. Bunun nedeni, tek boyutlu bir vektöre dönüştürülen girdinin sinir ağına eklenmesidir. Modelin girdileri, tek boyutlu lineer vektörler biçiminde olmalıdır. Bu nedenle, düzgün ve mükemmel bir şekilde yapıştırılmış bir katman gereklidir (Şekil 3.9.).



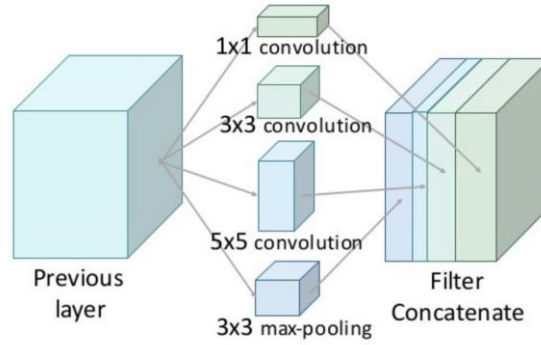
Şekil 3.9. Tam bağlı katman (Evrişimli Sinir Ağı, 2018)

3.3.1.2. CNN Mimarileri

Xception, Derinlemesine Ayrılabilir Evrişimleri (Depthwise Separable Convolutions) içeren derin bir CNN mimarisidir. Bu model, Google çalışanı Francois Chollet tarafından tanıtıldı. Inception modelinin ‘extreme’ versiyonu olarak bilinir (Chollet, 2016).

Inception, yinelenen modüllerden oluşan yapısı olan derin bir sinir ağıdır (Şekil 3.10.). Bu yinelenen modüllere başlangıç modülü denir. Bir başlangıç modülü, yukarıdaki şekilde gösterildiği gibi, aynı girdi üzerinde birden fazla farklı dönüşüm

hesaplar ve ardından tüm çıktıları birleştirerek modelin hangi özelliklerin ne kadar alınacağına karar vermesine olanak tanır.



Şekil 3.10. Inception mimarisi (Chollet, 2016)

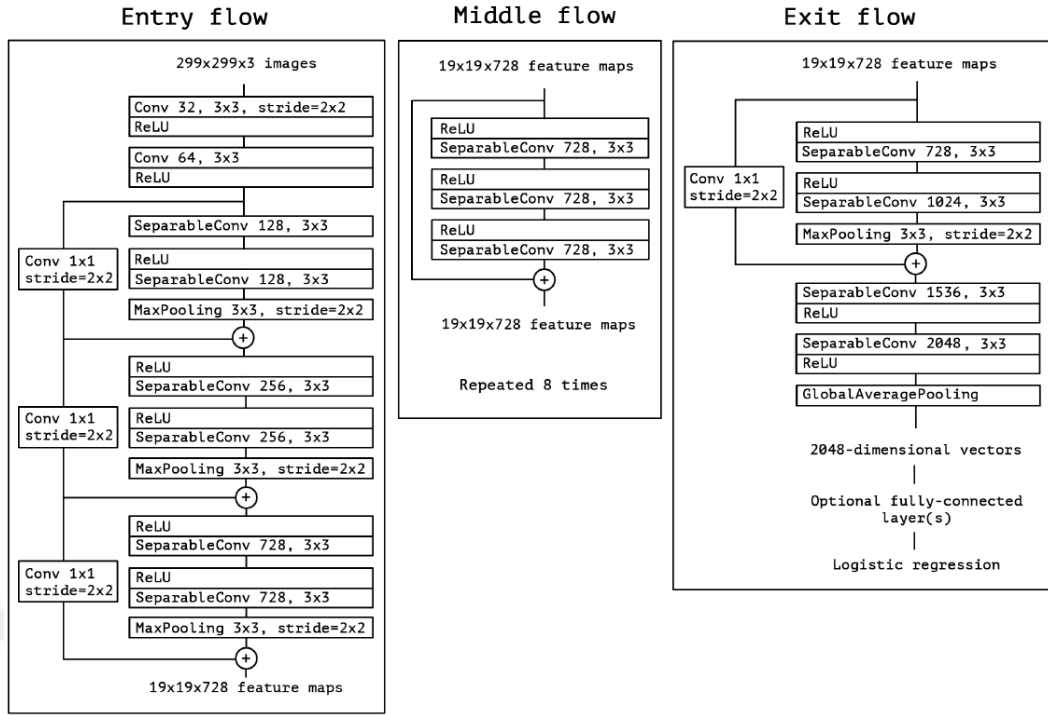
Xception, Inception modelinde girdiyi sıkıştırmak için 1x1 konvolüsyon kullanılan adımı tersine çevirir. Yani ilk olarak filtreleri derinlik haritalarına uygular ve daha sonra derinlik boyunca 1x1 konvolüsyon uygular.

Bu iki model arasında bir fark daha bulunmaktadır. Bu, Inception'da yukarıdaki her iki adımdan sonra da ReLu fonksiyonunun uygulanmasıdır. Xception'da ise sadece filtreler uygulandıktan sonra yani ilk adımda ReLu uygulanır.

Xception'ın Genel Mimarisi (Giriş Akışı > Orta Akış > Çıkış Akışı) (Derinlemesine Ayrılabilir Evrişim, 2018).

Xception modelinin dayandığı iki ana nokta vardır:

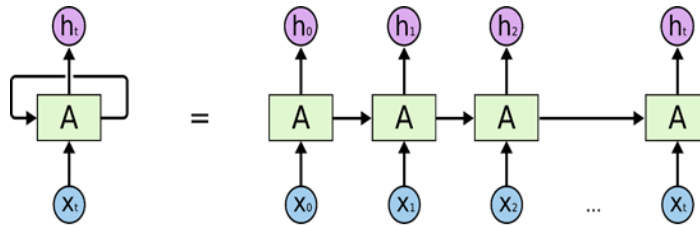
- Derinlemesine Ayrılabilir Evrişim
- ResNet'te olduğu gibi Evrişim blokları arasındaki kısayollar (Şekil 3.11.)



Şekil 3.11. Xception mimarisi

3.3.2. Özyinelemeli Sinir Ağı (Recurrent Neural Network, RNN)

Bazı durumlarda bir önceki çıktısını giriş verisi olarak kullanmayı sağlayabilen RNN, bir derin öğrenme mimarisidir. Bu yönüyle diğer ağlardan hatırlamasıyla ayrılır. Yavaş hesaplama ve önceki bilgilere erişimde zorluklar olsa bile girdi büyüklüğü değişmesi durumunda model boyutu artmaz, herhangi bir uzunluktaki veriyi işleyebilir. Genellikle doğal dil işleme ve konuşma tanıma gibi alanlarda kullanılır. Aşağıda görüldüğü gibi çıktı olan bir sonuç girdiyi etkiler (Şekil 3.12.).



Şekil 3.12. RNN yapısı

Geleneksel bir sinir ağı, yalnızca o anda verilen girdi bilgisini kullanarak bir görüntüdeki nesneleri tanır. Daha önce sisteme gönderilen bilgiler kullanılmadan

tahmin yapılması amaçlanır. RNN'ler ise yalnızca o anda sağlanan bilgileri değil, daha önce sisteme girilen veya daha sonra sisteme yüklenen bilgileri de kullanır. Dolayısıyla bu sinir ağları hafızalı ağlardır. Eski ve yeni verileri karşılaştırarak sonuca varırlar.

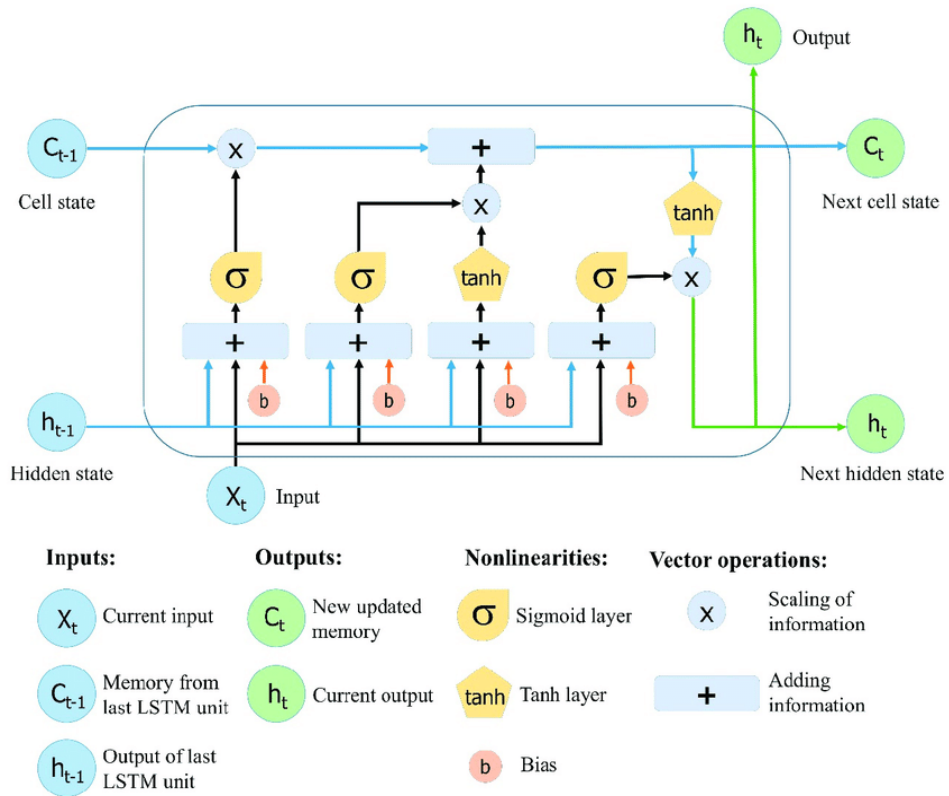
Tekrarlayan bir sinir ağı, zaman içinde birçok farklı yineleme gerçekleştirebilir. Bu yeniden denemeler, veriler üzerinde çok az etkiye sahip oldukları için verilerle ilişkili bazı parametrelerin sistemden kaldırılmasına neden olabilir. Bu parametreler sinir ağına uzun zaman önce eklendiyse, kullanıcılar bu bilgilere erişimi kaybedebilir. Bu atılan ve unutulmuş bilgileri bir havuzda toplamak için bir mimari geliştirilmiştir. LSTM bu mimarinin bir örneğidir. (Cho, 2014).

3.3.3. Uzun Kısa Süreli Bellek (Long Short Term Memory, LSTM)

Uzun Kısa Süreli Bellek, temel olarak RNN içerisindeki normal bir adımın farklılaştırılmasıdır. Bu farklılaşma düz ilerleyen bir sinir ağı yapısı yerine içerisinde farklı birkaç matematiksel işlem daha yapılarak yeni parametrelerin hesaplanması ve saklanmasıdır. Değerleri rastgele aralıklarla hatırlayan bir RNN mimarisidir denilebilir. Böylece daha önceki adımlardaki girdi ya da çıktı bilgilerine gerek duyulduğunda ulaşılabilir. Farklı farklı problemlerde çok iyi çalışmalarından dolayı günümüzde çok sık bir şekilde kullanılmaktadır.

LSTM'ler hem Uzun Süreli Bellek (LTM) hem de Kısa Süreli Bellek (STM) ile ilgilendir ve hesaplamaları basit ve etkili hale getirmek için kapı kavramını kullanır (Şekil 3.13.).

- Forget Gate (Unutma Kapısı): LTM bu kapıya gelir ve yararlı olmayan bilgileri unuttur.
- Learn Gate (Öğrenme Kapısı): Yeni Öğrenilen bilgilerin mevcut girdiye uygulanabilmesi için mevcut girdi ve STM birleştirilir.
- Remember Gate (Hatırlama Kapısı): Kalan LTM bilgileri ile STM ve mevcut girdi bu kapıda birleştirilir.
- Use Gate (Kullanma Kapısı): Mevcut olayın çıktısının tahmin edildiği kapı (Onan, 2022).



Şekil 3.13. LSTM diyagramı (Maniath, 2017)

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Derin öğrenme, yapay sinir ağları ve büyük veri kümeleri ile çalışarak karmaşık görevleri gerçekleştirebilen bir makine öğrenme yöntemidir. Görüntü işleme, doğal dil işleme ve derin öğrenmeyi birleştirmek, görüntülere otomatik altıyazı ekleme gibi ilginç uygulamalar sunar. Bu tez çalışması, derin öğrenmeyi kullanarak altıyazı oluşturmayı kapsar.

Altıyazı, bir görüntünün içeriğini doğal dilde açıklayan metin oluşturmaktır. Bunun, görme engelli kişilerin resimlerin anlamını anlamasını kolaylaştırmak ve resim içeriğini arama motorları için daha erişilebilir hale getirmek gibi birçok pratik kullanımı vardır. Derin öğrenmeye dayalı altıyazı, genellikle birkaç adımdan oluşan bir süreçtir.

Veri toplama: Bir derin öğrenme modelinin eğitimi, bir veri kümesi gerektirir. Altıyazı için, görüntü ve veri çiftlerini içeren bir veri seti oluşturulması gerekir. Bu kayıt, görüntüleri ve her görüntü için doğru başlığı içermelidir.

Ön işleme: Görüntü verileri derin öğrenme modellerine uygun hale getirilmelidir. İlk olarak, görüntü ölçeklenir ve normalleştirilir. Ardından, özellik çıkarma yöntemi kullanılarak görüntünün özellik vektörü çıkarılabilir. Özellik çıkarımı, görüntüyü daha düşük boyutlu bir temsile dönüştürmek için önceden eğitilmiş sinir ağları kullanılarak gerçekleştirilir.

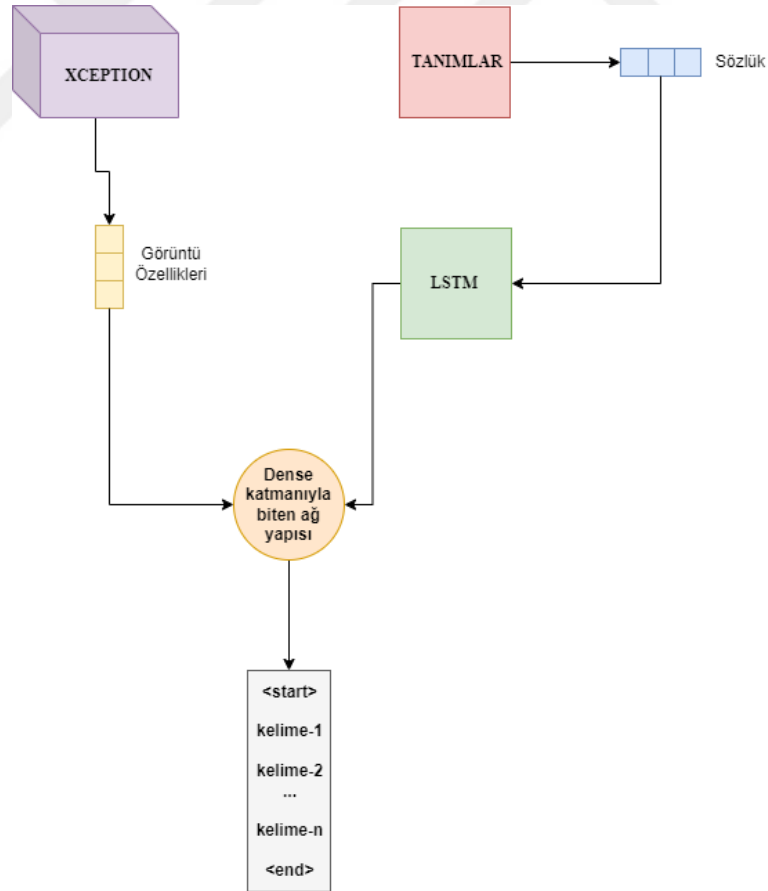
Model eğitimi: Görüntüleri ve altıyazıları eşleştirmek için bir derin öğrenme modeli eğitilir. Evrişimli Sinir Ağları ve Uzun Kısa Süreli Bellek gibi mimariler yaygın olarak kullanılmaktadır. CNN'ler görüntü verilerinden özellik çıkarmak için kullanılır ve LSTM'ler dil modelleri oluşturmak için kullanılır. Bu iki ağ birlikte, resim yazıları oluşturan bir model oluşturur.

Altıyazı oluşturma: Eğitilen model, yeni görüntüler için altıyazı oluşturmak için kullanılabilir.

Bu tez çalışmasında Evrişimli Sinir Ağı (CNN) ve Uzun Kısa Süreli Bellek (LSTM) derin öğrenme mimarileri kullanılarak görüntüler anlamsal olarak analiz edilmiştir. ImageNet veri seti kullanılarak daha önceden eğitilmiş model olan Xception'dan görüntü özellikleri çıkarılmıştır. Bir CNN modeli olan Xception'dan çıkarılan özellikler LSTM modeline girdi olarak verilerek altıyazılar oluşturulmuştur. Geliştirme Python dili ile yapılmıştır. Keras ve Numpy gibi kütüphaneler de kullanılmıştır.

MSCOCO veya Flickr_30K gibi veri setlerinin eğitim sırasında oldukça uzun zamanlar almaması adına daha küçük boyutta olan Flickr_8K veri seti kullanılmıştır. Bir fotoğraf paylaşım sitesi olan Flickr'dan oluşturulmuştur. Görüntü analizi ve doğal dil işleme gibi alanlarda kullanılan bir veri setidir. Toplam 8092 görüntüden oluşur ve her bir görüntü 5 altıyazı ile etiketlenmiştir. Bu da toplam 40460 altıyazı yapar. Yaklaşık 8000 görüntünün 6000 tanesi eğitim verisi, 2000 tanesi test için kullanıldı. Bu aşamada K-Fold çapraz doğrulama uygulaması da yapıldı. Veri seti 5 ayrı parçaya bölünerek, sırayla her parça veri seti test adımında kullanıldı. Eğitim için ayrılan veri setlerinde ise 10 epoch değeri verildi. Flickr_8K veri setini kullanarak, resimlerdeki nesneleri ve sahneleri tanıma, resimleri açıklamalarla eşleştirme, resimleri arama, resim altıyazıları otomatik fotoğraf oluşturma gibi çeşitli görevler gerçekleştirilebilir. Bu görevler, yapay zeka sistemlerinin görsel ve gramer bilgisini birleştirme yeteneğini geliştirmek için önemlidir.

Modelin şeması ve çapraz doğrulama sonuçları Şekil 4.1'de gösterilmektedir.



Şekil 5.1. Model Şeması

4.1. Kütüphane Hazırlığı

Bu çalışmada kullanılan kütüphaneler Şekil 4.2’de gösterilmektedir.

```
from keras.applications.xception import Xception, preprocess_input
from keras.preprocessing.text import Tokenizer
from keras.utils import plot_model
from keras.utils import to_categorical
import matplotlib.pyplot as plt
import numpy as np
from pickle import dump, load
from PIL import Image
import string
from tensorflow.keras.layers import Dense, Input, Conv2D, MaxPool2D, LSTM, add
from tensorflow.keras.layers import Activation, Dropout, Flatten, Embedding, concatenate
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import load_img, img_to_array
import os
from tqdm import tqdm_notebook as tqdm
tqdm().pandas()
```

<ipython-input-2-00699940c3aa>:18: TqdmDeprecationWarning: This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm\_notebook`
tqdm().pandas()
0/? [04:45<?, ?it/s]

Şekil 4.2. Kütüphaneler

4.2. Veri Temizleme

Verilen yoldaki dosyanın içeriğini okur (Şekil 4.3.).

```
def loadDocument(filename):
    file = open(filename, 'r')
    fileText = file.read()
    file.close()
    return fileText
```

Şekil 4.3. Veri okuma

Her görüntüye ait olan 5 etiketli altyazıyı, dosya adı ile birlikte bir sözlük yapısı haline getirir (Şekil 4.4.).

```

def getAllImagesCaptions(fileName):
    file = loadDocument(fileName)
    captions = file.split('\n')
    descriptions = {}
    for caption in captions[:-1]:
        image, caption = caption.split('\t')
        if image[:-2] not in descriptions:
            descriptions[image[:-2]] = [caption]
        else:
            descriptions[image[:-2]].append(caption)
    return descriptions

```

Şekil 4.4. Altyazı listesi

Bir önceki fonksiyonun çıktısı veri temizleme fonksiyonunun parametresidir. Etiket listesini alarak metinsel olarak bir temizleme işlemi yapar. Tüm kelimeler küçük harflerle ifade edilir, noktalama işaretleri ve sayılar etiketlerden kaldırır (Şekil 4.5.).

```

def dataCleaning(captions):
    table = str.maketrans('', '', string.punctuation)
    for image, captions in captions.items():
        for i, imageCaption in enumerate(captions):
            imageCaption.replace("-", " ")
            description = imageCaption.split()
            description = [word.lower() for word in description]
            description = [word.translate(table) for word in description]
            description = [word for word in description if (len(word)>1)]
            description = [word for word in description if (word.isalpha())]
            imageCaption = ' '.join(description)
            captions[image][i] = imageCaption
    return captions

```

Şekil 4.5. Veri temizleme

Elimizde bulunan etiket listesinin içerisindeki tüm metinleri tarayarak benzersiz olan sözcükleri bulur. Bu işlem sözlük yapısını ortaya çıkarır (Şekil 4.6.).

```

def descriptionVocabulary(descriptions):
    vocabulary = set()
    for key in descriptions.keys():
        [vocabulary.update(d.split()) for d in descriptions[key]]
    return vocabulary

```

Şekil 4.6. Sözlük oluşturma

Etiketlerin temizlenmiş halini bir dosyaya, yine etiketin ait olduğu dosya ismi ile yazar (Şekil 4.7.).

```
def saveDescriptions(descriptions, filename):
    lines = list()
    for key, descriptionList in descriptions.items():
        for description in descriptionList:
            lines.append(key + '\t' + description)
    data = "\n".join(lines)
    file = open(filename, "w")
    file.write(data)
    file.close()
```

Şekil 4.7. Kayıt işlemi

4.3. Özellik Çıkarma

Xception modelinin girdi boyutu 299x299x3'tür. Tüm görüntüler uygun boyutta bu modele verildiğinde görüntülere ait özellikleri çıkarır. Elde edilen bu özellik vektörünü ".p" uzantılı bir dosyaya kaydeder (Şekil 4.8.). Ardından bu dosya yüklenir (Şekil 4.9.).

```
def extractImageFeatures(directory):
    model = Xception(include_top=False, pooling='avg')
    features = {}
    for img in tqdm(os.listdir(directory)):
        filename = directory + "/" + img
        image = Image.open(filename)
        image = image.resize((299, 299))
        image = np.expand_dims(image, axis=0)
        #image = preprocess_input(image)
        image = image/127.5
        image = image - 1.0
        feature = model.predict(image)
        features[img] = feature
    return features
```

Şekil 4.8. Xception ile özellik çıkarma

```
features = extractImageFeatures(datasetImagesPath)
dump(features, open("features.p", "wb"))

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/xception/xception_weights_tf_dim_ordering_tf_kernels_notop.h5
83683744/83683744 [=====] - 1s 0us/step
<ipython-input-16-677cb42a706f>:4: TqdmDeprecationWarning: This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
  for img in tqdm(os.listdir(directory)):
100% [=====] 8091/8091 [1:21:15-00:00, 2.01it/s]
Görüntülenen çıkış son 5000 satıra kısaltıldı.
1/1 [=====] - 1s 635ms/step
1/1 [=====] - 0s 351ms/step
1/1 [=====] - 0s 361ms/step
1/1 [=====] - 0s 343ms/step
1/1 [=====] - 0s 356ms/step
1/1 [=====] - 0s 357ms/step
1/1 [=====] - 0s 361ms/step
1/1 [=====] - 0s 358ms/step
1/1 [=====] - 0s 358ms/step
1/1 [=====] - 0s 352ms/step
```

Şekil 4.9. Özelliklerin yüklenmesi

4.4. Veri Seti Yükleme

Görüntülere ait dosya isimlerini bir liste halinde geri döner (Şekil 4.10.).

```
def loadImages(filename):
    file = loadDocument(filename)
    images = file.split("\n")[:-1]
    return images
```

Şekil 4.10. Veri seti yükleme

LSTM modelinin, etiketlerin başlangıç bitiş noktalarını algılayabilmesi için her bir görüntüye ait olan etiketlere “<start>” ve “<end>” etiketlerini ekler (Şekil 4.11.).

```
def loadCleanDescriptions(filename, images):
    file = loadDocument(filename)
    descriptions = {}
    for line in file.split("\n"):
        words = line.split()
        if len(words)<1 :
            continue
        image, imageCaption = words[0], words[1:]
        if image in images:
            if image not in descriptions:
                descriptions[image] = []
            description = '<start> ' + " ".join(imageCaption) + ' <end>'
            descriptions[image].append(description)
    return descriptions
```

Şekil 4.11. LSTM modeline hazırlık

Yukarıdaki adımlarda yapılanlar, modelden çıkarılan özellikleri verir (Şekil 4.12.).

```
[ ] def loadFeatures(images):
    allFeatures = load(open("features.p","rb"))
    features = {k:allFeatures[k] for k in images}
    return features
```

Şekil 4.12. Özellikleri yükleme

4.5. Kelime Dağırcığı

Elimizde bulunan, görüntü etiketlerinden elde edilen, benzersiz sözlük yapısındaki her bir kelimeyi yine aynı şekilde her birini benzersiz bir sayı ile eşleştirir (Şekil 4.13.).

```
[ ] def createTokenizer(descriptions):
    descriptionList = dictionaryToList(descriptions)
    tokenizer = Tokenizer()
    tokenizer.fit_on_texts(descriptionList)
    return tokenizer
```

Şekil 4.13. Kelime-sayı eşleşmesi

Aşağıdaki çıktıda görüldüğü üzere 7577 benzersiz kelime vardır (Şekil 4.14).

```
▶ tokenizer = createTokenizer(trainingDescriptions)
  dump(tokenizer, open('tokenizer.p', 'wb'))
  vocabularySize = len(tokenizer.word_index) + 1
  vocabularySize
```

7577

Şekil 4.14. Eşsiz kelimeler

Her bir etiket in maksimum uzunluğunu bulur (Şekil 4.15.).

```
▶ def maxLength(descriptions):
    descriptionList = dictionaryToList(descriptions)
    return max(len(description.split()) for description in descriptionList)
```

```
[ ] maxLength = maxLength(descriptions)
  maxLength
```

32

Şekil 4.15. Maksimum uzunluk

4.6. Veri Oluşturma

Model in eğitim yapabilmesi için girdi ve çıktılar verilerek gözetimli bir öğrenme uygulanır (Şekil 4.16.).

X1: Görüntülerden elde edilen 2048 boyutlu özellik vektörüdür.

X2: Giriş metni.

Y: Çıktı (Şekil 4.17.).

```
▶ def createSequences(tokenizer, maxLength, descriptionList, feature):
    X1, X2, y = list(), list(), list()
    for description in descriptionList:
        sequence = tokenizer.texts_to_sequences([description])[0]
        for i in range(1, len(sequence)):
            inSequence, outSequence = sequence[:i], sequence[i]
            inSequence = padSequences([inSequence], maxlen=maxLength)[0]
            outSequence = to_categorical([outSequence], num_classes=vocabularySize)[0]
            X1.append(feature)
            X2.append(inSequence)
            y.append(outSequence)
    return np.array(X1), np.array(X2), np.array(y)
```

Şekil 4.16. Gözetimli öğrenme

```
[a,b],c = next(dataGenerator(trainingDescriptions, features, tokenizer, maxLength))
a.shape, b.shape, c.shape

((47, 2048), (47, 32), (47, 7577))
```

Şekil 4.17. Çıktı

4.7. Model Tanımlama

Kullanılacak modeli tanımlamak için Keras kullanılmıştır.

İlk adımda Dense katmanı ile elimizdeki özellik vektörünün boyutunu 256'ya indirilmiştir.

Embedding katmanında girdileri işler ve LSTM katmanı işlenir.

Son adım olarak, iki katmandan da elde edilen çıktılar birleştirilerek Dense katmanına tahmin yapması için gönderilir (Şekil 4.18.).

```
def defineModel(vocabularySize, maxLength):
    inputs1 = Input(shape=(2048,))
    fe1 = Dropout(0.5)(inputs1)
    fe2 = Dense(256, activation='relu')(fe1)
    inputs2 = Input(shape=(maxLength,))
    se1 = Embedding(vocabularySize, 256, mask_zero=True)(inputs2)
    se2 = Dropout(0.5)(se1)
    se3 = LSTM(256)(se2)
    decoder1 = add([fe2, se3])
    decoder2 = Dense(256, activation='relu')(decoder1)
    outputs = Dense(vocabularySize, activation='softmax')(decoder2)
    model = Model(inputs=[inputs1, inputs2], outputs=outputs)
    model.compile(loss='categorical_crossentropy', optimizer='adam')
    print(model.summary())
    plot_model(model, to_file='model.png', show_shapes=True)
    return model
```

Şekil 4.18. Model oluşturma

4.8. Model Eğitimi

Eğitim için ayrılan 6000 görüntü kullanılarak işlem başlatılır (Şekil 4.19.) ve aynı zamanda model kaydedilir (Şekil 4.20.).

```
print('Dataset: ', len(trainingImages))
print('Descriptions: train=', len(trainingDescriptions))
print('Photos: train=', len(trainingFeatures))
print('Vocabulary Size:', vocabularySize)
print('Description Length: ', maxLength)
model = defineModel(vocabularySize, maxLength)
epochs = 10
steps = len(trainingDescriptions)
os.mkdir("models")
for i in range(epochs):
    generator = dataGenerator(trainingDescriptions, trainingFeatures, tokenizer, maxLength)
    model.fit_generator(generator, epochs=1, steps_per_epoch= steps, verbose=1)
    model.save("models/model_" + str(i) + ".h5")
```

Şekil 4.19. Eğitim

Dataset: 6000
 Descriptions: train= 6000
 Photos: train= 6000
 Vocabulary Size: 7577
 Description Length: 32
 Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_3 (InputLayer)	[(None, 32)]	0	[]
input_2 (InputLayer)	[(None, 2048)]	0	[]
embedding (Embedding)	(None, 32, 256)	1939712	['input_3[0][0]']
dropout (Dropout)	(None, 2048)	0	['input_2[0][0]']
dropout_1 (Dropout)	(None, 32, 256)	0	['embedding[0][0]']
dense (Dense)	(None, 256)	524544	['dropout[0][0]']
lstm (LSTM)	(None, 256)	525312	['dropout_1[0][0]']
add_12 (Add)	(None, 256)	0	['dense[0][0]', 'lstm[0][0]']
dense_1 (Dense)	(None, 256)	65792	['add_12[0][0]']
dense_2 (Dense)	(None, 7577)	1947289	['dense_1[0][0]']

=====
 Total params: 5,002,649
 Trainable params: 5,002,649
 Non-trainable params: 0

Şekil 4.20. Model katmanları

4.8. Modelin Test Edilmesi

Model eğitiminin tamamlanmasının ardından test işlemi yapılır (Şekil 4.21.).

```
[ ] tokenizer2 = Tokenizer(split=" ")
    tokenizer2 = createTokenizer(trainingDescriptions)

    maxLength = 32

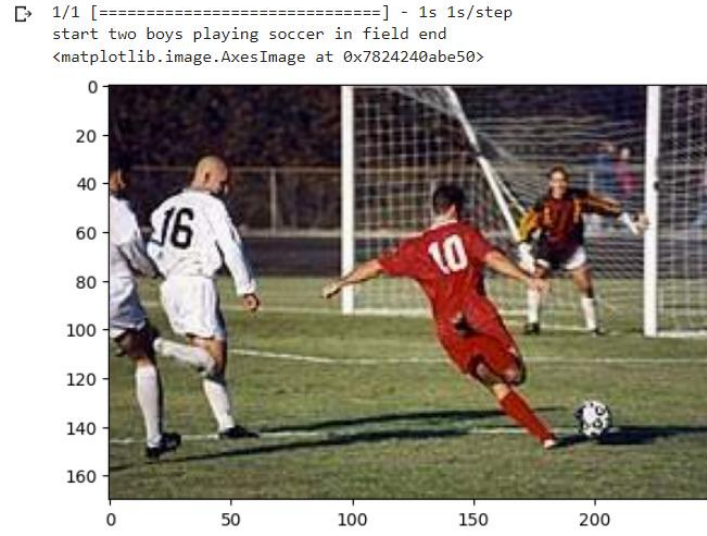
    model = load_model('/content/gdrive/MyDrive/python-project-image-caption-generator/models/model_9.h5')
    xceptionModel = Xception(include_top=False, pooling="avg")

    img_path = '/content/gdrive/MyDrive/test/futbol.jpg'
    photo = extractImageFeatures(img_path, xceptionModel)
    img = Image.open(img_path)
    description = generateDescription(model, tokenizer2, photo, maxLength)

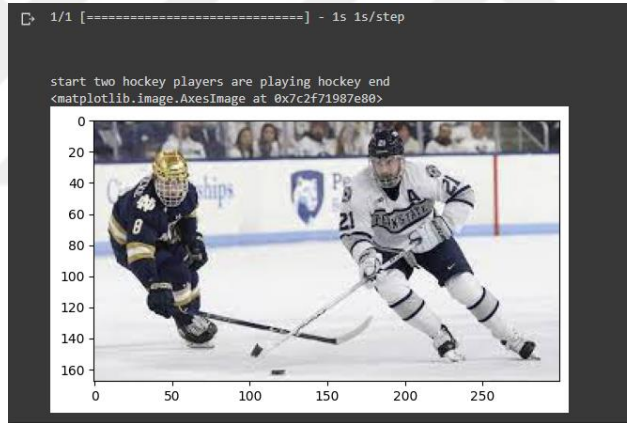
    print("\n\n")
    print(description)
    plt.imshow(img)
```

Şekil 4.21. Test aşaması

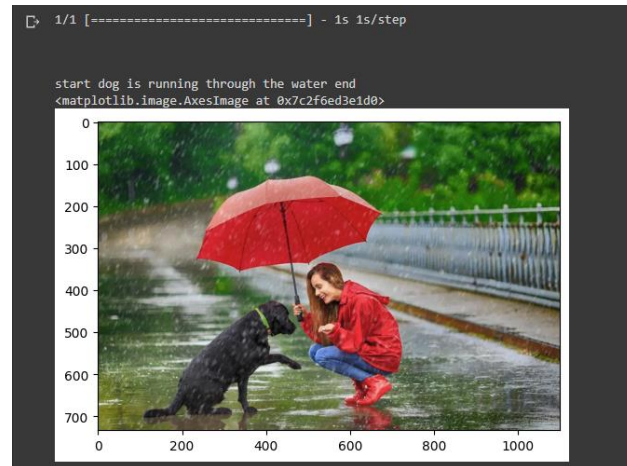
Elde edilen sonuçlar (Şekil 4.22., Şekil 4.23., Şekil 4.24.):



Şekil 4.22. Sonuç 1



Şekil 4.23. Sonuç 2



Şekil 4.24. Sonuç 3

5. SONUÇLAR VE ÖNERİLER

5.1. Sonuçlar

Bilgisayarlı görü ve doğal dil işlemenin buluştuğu bir noktada yapılan bu çalışma, resimlerin insanlar tarafından anlaşıldığı gibi yorumlanması, bilgisayar tarafından bu görüntüye ait altyazı oluşturulmasını sağlamaktır.

Bu çalışmanın amacı, derin öğrenme teknikleri kullanılarak görüntülerin anlamsal analizidir. Özellikle, evrişimli sinir ağları (CNN) ve uzun kısa süreli bellek (LSTM) ağları yardımıyla görüntü içeriği, özellikleri ve anlamı ile ilgili çeşitli analizler yapılmaktadır. Bu çalışmada Flickr_8k veri seti kullanılarak model performansı değerlendirilmiştir.

Görüntülerin anlamsal analizi, bilgisayarların görsel verileri anlama yeteneğini geliştirmek için önemli bir araştırma alanıdır. Bu çalışma ile derin öğrenme teknikleri kullanılarak bu analiz gerçekleştirilmektedir. CNN'ler, görüntü sınıflandırma ve özellik çıkarma gibi görevler için etkin bir şekilde kullanılır. Öte yandan LSTM, zamanla değişen verileri analiz etmek için kullanılan bir tür özyinelemeli sinir ağıdır. Bu çalışmada, CNN ve LSTM birleştirilerek görüntülerin anlamsal analizi yapılmaya çalışılmıştır.

Araştırmanın ilk adımı, Flickr_8k veri setinin toplanmasıdır. Flickr_8k, çeşitli kategorilerde yaklaşık 8.000 resim ve bunların açıklamalarını içeren bir veri setidir. Bu veri seti, derin öğrenme modellerinin eğitimi ve performans değerlendirmesi için kullanılmıştır.

Eğitim aşamasında CNN ve LSTM ağları birleştirilerek model oluşturulmuştur. İlk olarak, CNN kullanılarak görüntü özellikleri çıkarılmıştır. CNN'ler, görüntüyü daha küçük boyutlara indirgeyerek önemli özellikleri çıkaran evrişimli ve havuzlama katmanlarından oluşan bir mimariye sahiptir.

CNN'nin öznitelikleri daha sonra LSTM ağına girdi olarak iletilir. LSTM, dil modelleme gibi zamanla değişen verileri analiz etmek için kullanılan bir sinir ağıdır. Bu ağ, görüntüde yer alan nesnelerin ve sahnelerin açıklamasını oluşturmak için kullanılır.

Yapılan çalışma sonucunda CNN ve LSTM birleştirilerek oluşturulan bir model Flickr_8k veri seti üzerinde oldukça başarılı olmuştur. Görüntü sınıflandırma, özellik çıkarma, açıklama oluşturma gibi görevlerde başarılı sonuçlar alınmıştır. Modelin farklı kategorilere ait görüntüler üzerinde anlamsal analiz yapabildiği görülebilmektedir.

Aşağıda verilen tablolarda bazı değerlendirme sonuçları görülmektedir (Tablo 5.1. , Tablo 5.2.). İlk olarak eğitilmiş model, daha önce modelin görmemiş olduğu görüntüler ile test edildi (Tablo 5.1.). Ardından model eğitiminde ve modelin test aşamasında kullanılan görüntülerden 10 adet seçilerek bu görüntülere ait etiketler alındı ve görüntüler modele girdi olarak verilerek tahmin sonuçları elde edildi. Elde edilen bu bilgiler, tahmin sonucunun doğruluk oranı ile birlikte aşağıda verilmiştir (Tablo 5.2.).

Doğruluk oranı belirleme metriği olarak BLEU (Bilingual Evaluation Understudy) kullanılmıştır. Dilden bağımsız ve hesaplanması hızlı olmasından dolayı bu metrik tercih edilmiştir. İnsani değerlendirmeye yakındır.

BLEU skorunu hesaplariken referans cümlelere ihtiyaç vardır. Değerlendirilecek olan cümlelerin her kelimesi tek tek kontrol edilir. İlk kelimeden başlanarak; o kelimenin, referans cümleler içinde geçip geçmediğine bakılır. Referans cümleler içinde geçen kelime sayısı bulunur. Ardından değerlendirme cümlesinin kelime sayısına bölünerek Bleu sonucu elde edilir. Birkaç örnek ile açıklanırsa;

Referans Cümle 1: dog runs through the grass in the mountains

Referans Cümle 2: yellow dog is running up trail

Değerlendirilecek Cümle: start dog runs through the grass end

Değerlendirme cümlesindeki dog, runs, through, the, grass kelimeleri referans cümlelerinde en az bir kere geçmektedir. Dolayısıyla bu kelimelerin sayısı 5'tir. Değerlendirme cümlesinin kelime sayısı ise 5'tir. $5/5 = 1.0$. BLUE skoru 1 olarak hesaplanır. Bu ise mükemmel bir tahmin sonucu elde edildiği anlamına gelir.

Referans Cümle 1: children playing soccer near the goal net

Referans Cümle 2: four boys playing soccer

Değerlendirilecek Cümle: start two boys playing soccer on field end

Bu örnekte ise değerlendirme cümlesindeki boys, playing, soccer kelimeleri referans cümlelerde geçmektedir. Dolayısıyla elde edilen sayı 3'tür. Cümledeki kelime sayısı 6'dır. BLEU skoru $3/6 = 0,5$ elde edilir. Yani %50 oranında bir doğruluk anlamına gelir.

Tablo 5.1. Test sonuçları

TEST GÖRÜNTÜSÜ	TAHMİN SONUCU
	start boy is swimming underwater in pool end
	start little boy in red shirt is swinging on swing end
	start man in red shirt is walking down the street end
	start two boys playing soccer in field end
	start two hockey players are playing hockey end



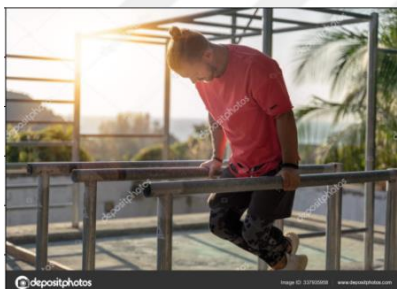
start man in red shirt is riding bike on the street end



start tennis player in red shirt is playing baseball end



start young boy in red shirt and jeans is playing with soccer ball end



start boy in red shirt is jumping in the air end



start dog is running through the water end

Tablo 5.2. Flickr_8k test sonuçları

FLICKR_8K TEST GÖRÜNTÜSÜ	ETİKETLER	TAHMIN SONUCU
	- man is skiing down mountain with another large mountain behind him and river between the mountains - man is skiing down snowy hill with large mountain in the background - man skiing down snow covered mountain - person skis down snowy mountain - skier goes down hill with mountain as backdrop	start man treks through the mountains end BLEU*=3/5=0.6
	- boy in black helmet and red long sleeve shirt rides his motorbike over rocky stream - man on motorcycle steers through swampy terrain - man rides his bike over rocks and creek - motocross bike is being ridden between markers in running stream - person is dirt biking over rocks and water	start man in red shirt is riding bike down dirt path end BLEU*=7/10=0.7
	- brown dog runs towards the camera on dirt trail with mountains in the background - brown dog runs toward the camera down rocky trail through the brush with mountains in the background - dog running on trail in the mountains - the brown dog is walking along the dirt path with beautiful mountains behind him - yellow dog is running up trail	start dog runs through the grass end BLEU*=5/5=1.0

	- baseball pitcher with shelton on his shirt throws the ball - baseball player in black and orange shirt and with glove on pitches the ball - male dressed in sports outfit tries to catch baseball - man playing baseball - the baseball player is running after the ball	start boy in red and white uniform running with soccer ball in his hand end BLEU*=7/13=0.54
	- man in blue goggles is in the swimming pool - man with blue yellow and green swimming cap is wearing goggles and swimming - swimmer doing the breaststroke in pool - swimmer swimming butterfly in pool - the swimmer is wearing green and yellow hat and goggles	start boy is swimming underwater in pool end BLEU*=4/6=0.66
	- boy with beach ball behind him playing in pool - young boy plays around in jacuzzi - children play in an inflatable pool - the boy laughed as he swam in the pool - two children are in pool with beach ball floating nearby	start young boy is swimming underwater in pool end BLEU*=4/7=0.57
	- group of young boys on opposing teams are playing soccer - children playing soccer near the goal net - four boys playing soccer - four children run along grassy field playing soccer - the young boy in the red shirt is kicking soccer ball	start two boys playing soccer on field end BLEU*=5/6=0.83

	- man in paddle boat on lake - man is rowing canoe on lake - man is rowing on his boat through very still water - old man canoeing with green lilly pads - the man in the brown cap is paddling boat on calm river	start man is kayaking in the water end BLEU*=4/6=0.66
	- climber walks along snowy peak - man climbs over snow covered rocks by shore - man is standing on snowcapped mountain - person hikes down snowy mountain - there man on top of an snow covered mountain with lake behind him	start man is walking on snowy mountain end BLEU*=5/6=0.83
	- brown dog is running through brown field - brown dog is running through the field - brown dog with collar runs in the dead grass with his tongue hanging out to the side - brown dog with his tongue wagging as he runs through field - dog running in the grass	start brown dog is running through the grass end BLEU*=7/7=1.0

Bu çalışmayı diğer çalışmalardan ayıran, kullanılan CNN modelidir. Bir CNN modeli olan Xception modeli, geleneksel CNN modellerinden farklı olarak, "derinlik ayrıştırma" adı verilen bir yaklaşım kullanır. Bu yaklaşım, daha ince bir ağın ardışık olarak birden çok konvolüsyon işlemi yapmasını sağlar. Bu model temelde iki aşamalı bir konvolüsyon işlemi olan "depthwise seperable convolution" kullanır. Bu işlem sayesinde ağıdaki her katman belirgin özellikleri öğrenmeye odaklanır. Bunun sebebi ise parametre sayısını düşürerek hesaplama maliyetini en aza indirger. Parametre sayısının az olması da Xception modelini eğitmek için daha az veriye ve hesaplama gücüne ihtiyaç olduğu anlamına gelir. Bu da modelin daha hafif olmasını ve daha az

bellek kullanmasını sağlar. Dolayısıyla Flickr_8k veri setinde bulunan yaklaşık 8000 görüntü ile de doğruluk oranı yüksek sonuçlar elde edilebilir. Öğrenilecek parametrelerin az olması, önceden eğitilmiş ağı farklı görevler için daha başarılı bir şekilde uyarlamayı kolaylaştırır. Yani kullanılan Xception modeli, transfer öğrenme için daha uygundur.

Sonuç olarak, Xception, geleneksel CNN modellerine göre daha hafif, daha etkin ve daha geliştirilebilir olma eğilimindedir. Bu sebepten dolayı bu çalışmada Xception modeli görüntülerin semantik analizinde kullanılmış, başarılı sonuçlar alınmıştır.

5.2. Öneriler

Bu çalışma, derin öğrenme tekniklerinin görüntülerin anlamsal analizinde etkili olduğunu göstermektedir. Ancak, gelecekteki araştırmalar projenin kapsamını genişletebilir ve performansını iyileştirebilir. Örneğin, daha büyük ve daha çeşitli veri kümeleri kullanmak, modelin geliştirme yeteneğini artırabilir. Ek olarak, farklı derin öğrenme mimarilerini veya öğrenme algoritmalarını test eden karşılaştırmalı çalışmalar yapılabilir. Bu sayede bu çalışmanın sonuçlarının daha da iyileştirilmesine yardımcı olacak ve bu alanda daha fazla çalışma yapılabilmesine olanak sağlayacaktır.

KAYNAKLAR

- Ak, K. E., Sun, Y., Lim, J. H. , Learning by Imagination: A Joint Framework for Text-based Image Manipulation and Change Captioning, in IEEE Transactions on Multimedia, doi: 10.1109/TMM.2022.3154154.
- Amirian, S., Rasheed, K., Taha T. R., Arabnia, H. R., Image Captioning with Generative Adversarial Network, 2019 International Conference on Computational Science and Computational Intelligence (CSCI), 2019, pp. 272-275, doi: 10.1109/CSCI49370.2019.00055.
- Baig, M. M. A., Shah, M. I., Wajahat, M. A., Zafar, N., Arif, O., Image Caption Generator with Novel Object Injection, 2018, Digital Image Computing: Techniques and Applications (DICTA), 2018, pp. 1-8, doi: 10.1109/DICTA.2018.8615810.
- Cho, K., Merriënboer, B.V., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Conference on Empirical Methods in Natural Language Processing.
- Chollet, F., 2016. Xception: Deep Learning with Depthwise Separable Convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1800-1807.
- Eker, A. G. & Duru, N. (2021). Medikal Görüntü İşlemede Derin Öğrenme Uygulamaları. Acta Infologica, 5 (2), 459-474. DOI: 10.26650/acin.927561
- Evrişimli Sinir Ağları, 2018, Evrişimli Sinir Ağları el kitabı, <https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-convolutional-neural-networks>
- Gupta, S., & Malik, J. (2015). Visual Semantic Role Labeling. ArXiv, abs/1505.04474.
- Hossain, M. Z., Sohel, F., Shiratuddin, M. F., Laga, H., Bennamoun, M., 2021, Text to Image Synthesis for Improved Image Captioning, in IEEE Access, vol. 9, pp. 64918-64928, 2021, doi: 10.1109/ACCESS.2021.3075579.
- Kumar, N. K., Vigneswari, D., Mohan, A., Laxman, K., Yuvaraj, J., Detection and Recognition of Objects in Image Caption Generator System: A Deep Learning Approach, 2019, 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, pp. 107-109, doi: 10.1109/ICACCS.2019.8728516.
- Küçük, D. & Arıcı, N. (2018). Doğal Dil İşlemede Derin Öğrenme Uygulamaları Üzerine Bir Literatür Çalışması. Uluslararası Yönetim Bilişim Sistemleri ve

- Bilgisayar Bilimleri Dergisi, 2 (2), 76-86. Retrieved from <https://dergipark.org.tr/en/pub/uybisbbd/issue/41787/443574>
- Lu, J., Xiong, C., Parikh, D., & Socher, R. (2016). Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3242-3250.
- Maniath, S., Ashok, P., Poornachandran, V.G., Sujadevi, P., Sankar, A.U. and Jan, S., "Deep learning LSTM based ransomware detection," 2017 Recent Developments in Control, Automation & Power Engineering (RDCAPE), Noida, India, 2017, pp. 442-446, doi: 10.1109/RDCAPE.2017.8358312.
- Onan, A. (2022). Türkçe Metin Madenciliği için Çalışan Bellek Bağlantıları Tabanlı Uzun Kısa Süreli Bellek Mimarisi. Avrupa Bilim ve Teknoloji Dergisi, Ejosat Special Issue 2022 (ICAENS-1), 239-246 . DOI: 10.31590/ejosat.1080239
- Öztürk, K. & Şahin, M. E. (2018). Yapay Sinir Ağları ve Yapay Zekâ'ya Genel Bir Bakış. Takvim-i Vekayi, 6 (2), 25-36. Retrieved from <https://dergipark.org.tr/en/pub/takvim/issue/40063/427526>
- Şeker, A., Diri, B., and Balık, H. H., "Derin Öğrenme Yöntemleri Ve Uygulamaları Hakkında Bir İnceleme", Gazi Mühendislik Bilimleri Dergisi, vol. 3, no. 3, pp. 47-64, Dec. 2017
- Ser, G. & Bati, C. T. (2019). Derin Sinir Ağları ile En İyi Modelin Belirlenmesi: Mantar Verileri Üzerine Keras Uygulaması. Yuzuncu Yıl University Journal of Agricultural Sciences , 29 (3) , 406-417 . DOI: 10.29133/yyutbd.505086
- Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2014). Show and tell: A neural image caption generator. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3156-3164.
- Ünay, M. (2021). Bilgisayarlı görü teknikleri kullanılarak yapay zekâ temelli limon ağacı rekolte tahmini. KMÜ, Fen Bilimleri Enstitüsü, Elektrik Elektronik Mühendisliği Ana Bilim Dalı, Karaman
- Wang, Y., Xu, J., & Sun, Y. (2022). End-to-End Transformer Based Model for Image Captioning. AAAI Conference on Artificial Intelligence.
- Wang, Q., Huang, W., Zhang, X., Li, X., 2021, Word–Sentence Framework for Remote Sensing Image Captioning, in IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 12, pp. 10532-10543, Dec. 2021, doi: 10.1109/TGRS.2020.3044054.
- Wu, C., Yuan, S., Cao, H., Wei, Y., Wang, L., Hierarchical Attention-Based Fusion for Image Caption With Multi-Grained Rewards, in IEEE Access, vol. 8, pp. 57943-57951, 2020, doi: 10.1109/ACCESS.2020.2981513.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A.C., Salakhutdinov, R., Zemel, R.S., & Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *International Conference on Machine Learning*.

Yüksel, A. S. & Karabıyık, M. A. (2022). Doğal dil işleme yöntemleriyle metinden SQL sorgusu tahmini üzerine bir çalışma. Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi, 11 (4), 846-855. DOI: 10.28948/ngumuh.1120264

