KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

GÖRÜNTÜ BAŞLIKLAMA YÖNTEMİ İLE ÜRÜN GÖRÜNTÜLERİNE UYGUN BAŞLIKLARIN ÜRETİLMESİ

BİRKAN ATICI

KOCAELİ 2021

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ

ANABİLİM DALI

YÜKSEK LİSANS TEZİ

GÖRÜNTÜ BAŞLIKLAMA YÖNTEMİ İLE ÜRÜN GÖRÜNTÜLERİNE UYGUN BAŞLIKLARIN ÜRETİLMESİ

BİRKAN ATICI

Doç.Dr. Sevinç İLHAN OMURCA		
Danışman,	**Kocaeli Üniversitesi**	**.....**
Prof.Dr. Nevcihan DURU		
Jüri Üyesi,	**Kocaeli Üniversitesi**	**.....**
Dr.Öğr. Üyesi Alper Kamil DEMİR		
Jüri Üyesi,	**Adana Bilim ve Teknoloji Üniv.**	**.....**

Tezin Savunulduğu Tarih: 29.01.2021

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında derin öğrenme temelli görüntü başlıklama yöntemi ile ürün görüntülerine uygun başlıkların otomatik olarak üretilmesi amaçlanmıştır.

Lisans, yüksek lisans ve tez çalışmamda, beni destekleyen ve çalışmalarıma yön veren danışmanım Doç. Dr. Sevinç İLHAN OMURCA hocama teşekkürlerimi sunarım.

Yüksek lisans çalışmamda ve iş hayatımda her türlü desteği veren, gelişimimde büyük rol oynayan Ömer Faruk ÇELEBİ ve Enes TAYLAN'a teşekkürlerimi sunarım.

Sahibinden Bilgi Teknolojileri şirketine yüksek lisans ve tez çalışmamdaki desteklerinden dolayı teşekkürlerimi sunarım.

Ve tabi ki hayatım boyunca beni desteklerinden eksik etmeyen, bana güvenen ve her daim yanımda olan aileme sonsuz teşekkürlerimi sunarım.

Şubat – 2021 Birkan ATICI

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR i
İÇİNDEKİLER ii
ŞEKİLLER DİZİNİ iii
TABLOLAR DİZİNİ iv
SİMGELER VE KISALTMALAR DİZİNİ v
ÖZET vii
ABSTRACT viii
GİRİŞ 1
1. İLGİLİ ÇALIŞMALAR 4
2. YÖNTEM 13
2.1. Yapay Sinir Ağları 13
2.1.1. Aktivasyon fonksiyonu 14
2.1.2. Kayıp fonksiyonu 15
2.1.3. En iyileme algorimaları 16
2.2. Evrişimli Sinir Ağları 17
2.2.1. Evrişim katmanı 18
2.2.2. Havuzlama katmanı 19
2.2.3. Tam bağlanım katmanı 20
2.3. Yinelemeli Sinir Ağları 22
2.4. Kelime Temsil Yöntemleri 28
2.4.1. Kelime çantası yöntemi 28
2.4.2. Word2Vec yöntemi 29
2.4.3. GloVe temsili 30
2.4.4. FastText temsili 31
2.5. Kodlayıcı – Kod Çözücü Mimarisi 31
2.6. Dikkat Mekanizması 32
2.7. Değerlendirme Ölçütleri 34
2.7.1. BLUE 34
2.7.2. METEOR 34
2.7.1. CIDEr 35
3. DENEYSEL ÇALIŞMA 36
3.1. Veri Kümesi 36
3.2. Derin Öğrenme Model Tasarımı 38
3.2.1. Ürün görüntülerinden özellik vektörü çıkartımı 39
3.2.2. Ürün başlıkların ön işlemesi ve vektör çıkartımı 40
3.2.3. Tensorflow modellerinin oluşturulması 41
3.3. Model Eğitimi 44
3.4. Deneysel Çalışma Sonuçları 44
4. SONUÇLAR VE ÖNERİLER 49
KAYNAKLAR 51
KİŞİSEL YAYIN VE ESERLER 62
ÖZGEÇMİŞ 63

ŞEKİLLER DİZİNİ

Şekil 1.1. Obje, sahne ve olay üçlü gösterimi 4
Şekil 1.2. Ordonez ve diğerlerinin önerdiği model akışı 5
Şekil 1.3. m-RNN mimarisi....6
Şekil 1.4. DeepVS model genel yaklaşımı....7
Şekil 1.5. Dikkat mekanizmasının görüntü üzerindeki odaklanması 8
Şekil 1.6. MS COCO veri kümesinden örnekler 12
Şekil 2.1. Algılayıcı modeli....13
Şekil 2.2. İki gizli katmandan oluşan nöron ağı 14
Şekil 2.3. Yaygın olarak kullanılan aktivasyon fonksiyonları 15
Şekil 2.4. Görüntü temsili için oluşturulan örnek piksel temsili 17
Şekil 2.5. Evrişimli sinir ağı katmanları 18
Şekil 2.6. Evrişim işlemi 18
Şekil 2.7. Zero padding işlemi....19
Şekil 2.8. Same padding işlemi 19
Şekil 2.9. Havuzlama teknikleri 20
Şekil 2.10. Düzleme işlemi 21
Şekil 2.11. Yinelemeli sinir ağı modeli 23
Şekil 2.12. LSTM mimarisi....25
Şekil 2.13. Hücre durum vektörü ve kapıları 26
Şekil 2.14. GRU mimarisi ve kapılar üzerinde gerçekleştirilen işlemler 28
Şekil 2.15. CBOW ve Skip-gram modelleri....29
Şekil 2.16. Word2Vec uzay temsili 30
Şekil 2.17. Kodlayıcı - kod çözücü genel yapısı 31
Şekil 2.18. Dikkat mekanizmasının yapısı....33
Şekil 3.1. Veri kümesinden örnekler 36
Şekil 3.2. Görüntü filtrelemek için kullanılan sınıflandırıcı mimarisi....37
Şekil 3.3. Metin filtrelemek için oluşturulan sınıflandırıcı mimarisi 38
Şekil 3.4. InceptionV3 mimarisi....39
Şekil 3.5. Tensorflow üzerinde InceptionV3 modeli ile transfer öğrenme modelinin kurulması 39
Şekil 3.6. Başlıkların ön işlenmesi ve vektöre çevrilmesi 40
Şekil 3.7. Dikkat mekanizması modeli 41
Şekil 3.8. Kodlayıcı modelinin Tensorflow üzerinde oluşturulması 42
Şekil 3.9. Kodlayıcı modelinin özet bilgisi 42
Şekil 3.10. Kod çözücü modelinin Tensorflow üzerinde oluşturulması....43
Şekil 3.11. Kod çözücü modelinin özet bilgisi....43
Şekil 3.12. Model eğitimi sonucunda ortaya çıkan kayıp grafiği....44
Şekil 3.13. Deneysel çalışmada elde edilen başarılı sonuçlar 46
Şekil 3.14. Deneysel çalışmada elde edilen kısmen başarılı sonuçlar 47
Şekil 3.15. Deneysel çalışmada elde edilen başarısız sonuçlar 48

TABLOLAR DİZİNİ

Tablo 1.1. Görüntü başlıklama çalışmalarının genel özeti 9
Tablo 1.2. MS COCO veri kümesi ile yapılan çalışmalarının başarımları 11
Tablo 2.1. Evrişimli sinir ağı mimarileri 22
Tablo 2.2. Yinelemeli sinir ağı mimarileri 24
Tablo 2.3. Hafıza hücre kapıları 26
Tablo 3.1. Veri kümesi detayları 38
Tablo 3.2. Deneysel çalışma hata oranları 45

SİMGELER VE KISALTMALAR DİZİNİ

a	:	Görüntünün parçaları
a	:	Görüntü parçalarının ağırlık değerleri
b	:	Eğilim (bias) değeri
c	:	Kodlayıcı – kod çözücü durum vektörü
C	:	LSTM durum vektörü
D	:	Görüntü parça vektörü
e	:	Hizalama model çıktısı
h	:	Gizli katman vektörü
I	:	Görüntü vektörü
L	:	Görüntü parçalarının sayısı
m	:	Matris sütun sayısı
n	:	Matris satır sayısı
p	:	Olasılık matrisi
S	:	Başlık vektörü
W	:	Nöron ağırlıkları
x	:	CNN modelinin ürettiği görüntü vektörü
x	:	Nöron giriş vektörü
y	:	Gerçek çıktı vektörü
y_{tahmin}	:	Modelin ürettiği çıktı vekörü
$\hat{y}$	:	One-hot vektörü
$\hat{y}_{tahmin}$	:	Ayrık olasılık vektörü
z	:	Tekrarlı sinir ağı durum vektörü
θ	:	Model parametreleri
σ	:	Sigmoid fonksiyonu

Kısaltmalar

AP	:	Average Precision (Ortalama Hassasiyet)
API	:	Application Programming Interface (Uygulama Programlama Arayüzü)
AdaGrad	:	Adaptive Gradient Algorithm (Uyarlanabilir Gradyan Algoritması)
BLUE	:	Bilingual Evaluation Understudy (İki Dilli Değerlentirme Aktörü)
CIDEr	:	Consensus-based Image Description Evaluation (Fikir Birliğine Dayalı Görüntü Açıklama Değerlendirmesi)
CLS	:	Classification Score (Sınıflandırma Puanı)
CNN	:	Convolutional Neural Network (Evrişimli Sinir Ağları)
CUDA	:	Compute Unified Device Architecture (Birleşik Hesaplamalı Cihaz Mimarisi)
DeepVS	:	Deep Visual-Semantic (Derin Görsel Anlam)
E-GAN	:	Evaluation Generative Adversarial Networks (Çekişmeli Üretici Ağ Değerlendirmesi)
F1	:	F-score (F skoru)

GAN	:	Generative Adversarial Network (Üretken Tartışmalı Ağ)
GRU	:	Gated Recurrent Unit (Geçitli Tekrarlayan Birim)
HE	:	Human Evaluation (İnsan Değerlendirmesi)
IAPRTC	:	International Association for Pattern Recognition Technical Committee (Uluslararası Örüntü Tanıma Teknik Komitesi Derneği)
ILSVRC	:	ImageNet Large Scale Visual Recognition Challenge (ImageNet Büyük Ölçekli Görsel Tanıma Yarışması)
IoU	:	Intersection over Union (Birliğin Kesişnesi)
LSA	:	Latent Semantic Analysis (Gizli Anlamsal Analiz)
LSTM	:	Long Short-Term Memory (Uzun Kısa Süreli Bellek)
METEOR	:	Metric for Evaluation of Translation with Explicit ORdering (Açık ORdering ile Çevirinin Değerlendirilmesi için Metrik)
MNLM	:	Multimodal Neural Language Models (Multimodal Sinir Dili Modelleri)
MS COCO	:	Microsoft Common Objects in Context (Microsoft Bağlam Kapsamında Ortak Objeler)
NIC	:	Neural Image Captioning (Nöral Görüntü Başlıklama)
PASCAL	:	Pattern Analysis, Statistical Modelling and Computational Learning (Örüntü Analizi, İstatistiksel Modelleme ve Hesaplamalı Öğrenme)
PPLX	:	Perplexity (Karışıklık)
R@K	:	Recall Rate at Top K Candidates (En iyi K adayın Hatırlanması)
RNN	:	Recurrent Neural Network (Tekrarlayan Sinir Ağları)
ROUGE	:	Recall-Oriented Understudy for Gisting Evaluation (Bilgilendirme Değerlendirmesi için Geri Çağırma Odaklı Aktör)
ResNet	:	Residual Neural Network (Artık Sinir Ağı)
RmsProp	:	Root Mean Square Propagation (Kök Ortalama Kare Yayılımı)
SC-NML	:	Structure-Content Neural Language Model (Yapı-İçerik Sinir Dili Modeli)
SGD	:	Stochastic Gradient Descent (Stokastik Gradyan İnişi)
SPICE	:	Semantic Propositional Image Caption Evaluation (Anlamsal Önerme Resim Başlığı Değerlendirmesi)
STAIR	:	Software Technology and Artificial Intelligence Research (Yazılım Teknolojisi ve Yapay Zeka Araştırmaları)
SVM	:	Support Vector Machine (Destek Vektör Makineleri)
Seq2Seq	:	Sequence to Sequence (Sıralıdan Sıralıya)
TPU	:	Tensor Processing Unit (Tensör İşleme Birimi)
VGGNet	:	Visual Geometry Group Network (Görsel Geometri Grup Ağı)
m-RNN	:	multimodal Recurrent Neural Network (Multimodal Tekrarlayan Sinir Ağı)
mrank	:	Median Ranking (Medyan Sıralaması)

GÖRÜNTÜ BAŞLIKLAMA YÖNTEMİ İLE ÜRÜN GÖRÜNTÜLERİNE UYGUN BAŞLIKLARIN ÜRETİLMESİ

ÖZET

Teknolojik gelişmelerle birlikte insanların ürettikleri veri miktarında oldukça artış görülmektedir. Artan veri miktarı ve hesaplama teknolojilerindeki gelişmelerin sonucunda ise derin öğrenme çalışmaları son yıllarda hız kazanmıştır. Bazı problemlerde, geleneksel makine öğrenmesi tekniklerinin yetersiz ya da fazla başarı gösterememesi nedeniyle derin öğrenme teknikleri bu alanlarda önem kazanmıştır. Bu alanlardan birisi de görüntü başlıklama çalışmalarıdır.

Görüntü başlıklama, görüntü içerisindeki obje, sahne ve olayı anlamlandırıp doğal dile otomatik olarak aktarmayı sağlayan bir çalışma alanıdır. Literatürde bu çalışmalar oldukça zorlu bir problem olarak görülmektedir. Bunun en önemli nedeni ise, görüntü işleme ve doğal dil işleme tekniklerinin bir arada kullanılıyor olmasıdır. Diğer bir nedeni ise bu çalışmalar için kullanılacak veri kümesinin yetersiz kalmasıdır. Görüntü başlıklama çalışmaları, derin öğrenme teknikleri öncesinde geleneksel makine öğrenmesiyle, şablon ve getirme tabanlı olarak yapılmaktaydı. Ancak bu teknikler oldukça kısıtlı ve görüntüyü açıklamakta yetersiz kalmaktaydı. CNN ve LSTM gibi derin öğrenme teknikleri ile yapılan bu çalışmalar model başarısını oldukça arttırmıştır.

Bu çalışmada, sahibinden.com sitesindeki ürün ilanların görüntü-başlık ikili verileriyle, ürün resimlerine uygun olacak başlıkların çıkartılması amaçlanmıştır. Derin öğrenme yöntemleri kullanılarak yapılan bu çalışmada, yaklaşık 1,8 milyon görüntü-başlık verisi ile eğitim gerçekleştirilmiştir. BLUE, METEOR ve CIDEr skor başarım ölçütleri ile değerlendirilen çalışmamızda oldukça başarılı sayılabilecek sonuçlar elde edilmiştir.

Anahtar Kelimeler: Dikkat Mekanizması, Evrişimli Sinir Ağları, Görüntü Başlıklama, Kodlayıcı – Kod Çözücü Modeli, Uzun-Kısa Vadeli Bellek.

GENERATING TITLES FOR PRODUCT IMAGES WITH AN IMAGE CAPTIONING TECHNIQUE

ABSTRACT

Recently, with evolution of technology, there has been a significant increase in the amount of data produced by people. As a result of the increasing amount of data and developments in computing technologies, deep learning studies have gained momentum in recent years. In some problems, deep learning techniques have gained importance due to the insufficient results of traditional machine learning techniques. One of these areas is image captioning studies.

Image captioning studies provide that make sense of the object, scene and action in the image and generate automatically title of the image. These studies are considered challenging problem in the literature. One of the most important reasons why it is seen as challenging is that uses image processing and natural language processing techniques together. Another reason is that the dataset is not sufficient for model training. Before the deep learning techniques, these studies were done with traditional machine learning techniques such as template and retrieval based methods. But these traditional techniques are very limited and insufficient to fully define the image. With the deep learning techniques, the success of these studies has increased.

This study aimed to generate titles that will be suitable for the product images with the image - title pair of the product classified ads on sahibinden.com website. The model developed in this study by using deep learning methods, approximately 1.8 million image-title pair were used. The deep learning model, which was evaluated with the BLUE, METEOR and CIDEr score performance metrics, achieved results that could be considered successful.

Keywords: Attention Mechanism, Convolutional Neural Networks, Image Captioning, Encoder - Decoder Model, Long-Short Term Memory.

GİRİŞ

Günümüzde teknolojinin gelişmesiyle birlikte yapay zekâ çalışmaları da önem kazanmıştır. Bu çalışmaların gelişmesinin en büyük nedenlerinden biri oyun sektörü için yatırım yapılan grafik kartlarının aynı zamanda derin öğrenme çalışmaları için de kullanılabilir duruma gelmesidir. 2007 yılında Nvdia firmasının CUDA (Compute Unified Device Architecture) olarak adlandırdığı paralel hesaplama yapabilen ve yapay zekâ alanında çalışan araştırmacılarında üzerinde kodlama yapabileceği ekran kartını çıkartması ile birlikte derin öğrenme çalışmaları için ihtiyaç duyulan hesaplama yeteneği ortaya çıkmıştır. Google 2016 yılında, ekran kartlarının da ötesine giderek derin öğrenme modelinin temel birimi olan tensörler üzerinde çok daha hızlı hesaplama yeteneği olan TPU (Tensor Processing Unit) teknolojisini tanıtmıştır. Bu gelişmelerle birlikte derin öğrenme modellerini eğitmek için gerekli olan hesaplama gücü ortaya çıkmıştır. Derin öğrenme çalışmalarının ivmelenmesinin diğer bir nedeni ise veri kümeleridir. İnternet kullanımının yaygınlaşması ile birlikte, internet kullanıcılarının modelleri eğitmek için gerekli olan veri kümelerinin hazırlanmasına ön ayak olmuştur. Doğal dil işleme alanında Wikipedia, görüntü işleme çalışmalarında Flickr ve video işleme çalışmalarında ise Youtube üzerindeki etiketlemmiş veri kümeleri, derin öğrenme çalışmalarının hız kazanmasına yardımcı olmuştur. Bu gelişmelerle birlikte, kullanılan algoritmalarda ve modelleri kolayca tasarlanıp, tensör işlemlerini basit fonksiyonlar ile gerçekleştirilmesini sağlayan Tensorflow, Theano, Pytorch, Keras gibi kütüphanelerin ortaya çıkmasıyla birlikte derin öğrenme modellerinin eğitilip gerçek problemler üzerinde kullanılmasını da kolaylaşmıştır.

Derin öğrenme çalışmalarının geleneksel makine öğrenmesi çalışmalarına göre en önemli avantajlarından biri modeli eğitmek için gerekli olan öznitelik çıkarımının derin öğrenme çalışmalarında otomatik olarak gerçekleştirilmesidir. Geleneksel makine öğrenmesi ile yeterli başarım elde edilemeyen bazı problemler, derin öğrenmenin avantajlarından yararlanmak isteyen araştırmacıları da derin öğrenme çalışmalarına yöneltmiştir. Bu problemlerden birisi de görüntü başlıklama (image captioning) çalışmalarıdır.

Görüntü başlıklama, temelde görüntü içerisindeki obje, sahne ve eylemden anlam çıkartıp bunu semantik ve dil kurallarına uygun olacak şekilde doğal dile aktarmayı sağlayan çalışma alanıdır.

Görüntü başlıklama çalışmaları, otonom araçlarda sürüş esnasında bulunduğu ortamı anlandırıp sürüş desteği vermek, görme yitisini kaybetmiş insanlara etrafı açıklamak, güvenlik kameralarında ise ortamdaki olayların tehlike içerip içermediğini tanımlamak için kullanılmaktadır.

Derin öğrenme teknikleri öncesi bu çalışmalar geleneksel makine öğrenmesi teknikleri olan şablon tabanlı ve getirme tabanlı yöntemler ile yapılmaktaydı. Şablon tabanlı görüntü başlıklama çalışmalarında, belirli sayıda boşluk içeren birden fazla şablondan, görüntü üzerinde tanımlanan obje, nesne ve olay sınıflandırılıp, şablon üzerindeki boşlukların doldurulması ile başlık üretme yapılmıştır. Getirme tabanlı çalışmalarda ise ilk önce görüntü-başlık ikili veri kümesi oluşturulur. Sonrasında başlıklandırılmak istenen görüntü, veri kümesindeki görüntüler ile karşılaştırılarak, benzerliği en yüksek olan görüntünün başlığı, yeni gelen görüntünün başlığı olarak kabul edilir. Şablon ve getirme tabanlı bu yöntemler görüntüyü tanımlamak için oldukça yetersiz ve sınırlı kalmaktaydı.

Derin öğrenme teknikleriyle birlikte diğer çalışmalarda olduğu gibi görüntü başlıklama çalışmalarında da oldukça başarılı sonuçlar elde edilmiştir. Aynı zamanda MS COCO (Microsoft Common Objects in Context) ve Flickr gibi görüntü-başlık veri kümelerinin kullanıma sunulması da bu çalışmalarda başarılı sonuçların alınmasına yardımcı olmuştur.

Derin öğrenme tabanlı gerçekleştirilen görüntü başlıklama çalışmaları temel olarak kodlayıcı-kod çözücü (encoder-decoder) mimarisine dayanmaktadır. İlk önce makine çevirisi için kurulan bu mimari sıralı şekilde olan verileri yine sıralı veri dizisine dönüştürmek için kullanılmaktadır. Örneğin makine çevirisi problemi, bir dildeki sıralı kelimelerden oluşan cümleyi başka bir dile yine sıralı kelime dizisi olarak çevirmeyi amaçlar. Burada cümle içerisindeki kelimeler sıralaması cümle anlamını sağlamasında önem arz etmektedir. Bu yüzden makine çevirisi çalışmalarında hem kodlayıcı hem de kod çözücü olarak Tekrarlayan Sinir Ağları (Recurrent Neural Network, RNN)

kullanılmaktadır. Tekrarlayan Sinir Ağları, eğitim esnasında önceki durum bilgisini hatırlayarak kelimelerin cümle içerisindeki anlamını da öğrenir.

Makine çevirisinden ilham alınarak, kodlayıcı-kod çözücü mimarisi görüntü başlıklama çalışmalarında da kullanılmaya başlanmıştır. Makine çevirisinden farklı olarak kodlayıcı kısmında, görüntülerin özelliklerini çıkartmak için Evrişimli Sinir Ağları (Convolutional Neural Network, CNN) kullanılmıştır. Kod çözücü kısmında ise makine çevirisinde olduğu gibi Tekrarlayan Sinir Ağları kullanılmıştır. Bu alanda yapılan çalışmalarda modelin başarısını arttırmak için kodlayıcı ve kod çözücü kısımlarında çeşitli çalışmalar yapılmıştır. Bu çalışmalardan en dikkat çekeni ise dikkat mekanizmasıdır (attention mechanism). Dikkat mekanizması, kod çözücü kısmında başlık üretilirken girdi olarak verilen görüntünün dinamik olarak farklı kısımlarına odaklanmasını sağlamıştır.

Bu çalışmada derin öğrenme tabanlı görüntü başlıklama mimarisi, ikinci el ürünlerin görüntü-başlık veri kümesine uygulanmıştır. Sahibinden.com sitesi üzerinden alınan ve filtrelenen yaklaşık 1,8 milyon görüntü-başlık ikili veri kümesi model eğitiminde kullanılmıştır. Bu alandaki en başarılı çalışmalardan biri olan kodlayıcı – kod çözücü üzerine dikkat mekanizması uygulanan mimari kullanılmıştır. Görüntü başlıklama çalışmaları için değerlendirme ölçütleri olan BLUE, METEOR ve CIDEr metrikleri ile değerlendirilen çalışmamızda oldukça başarılı sayılacabilecek sonuçlar elde edilmiştir.

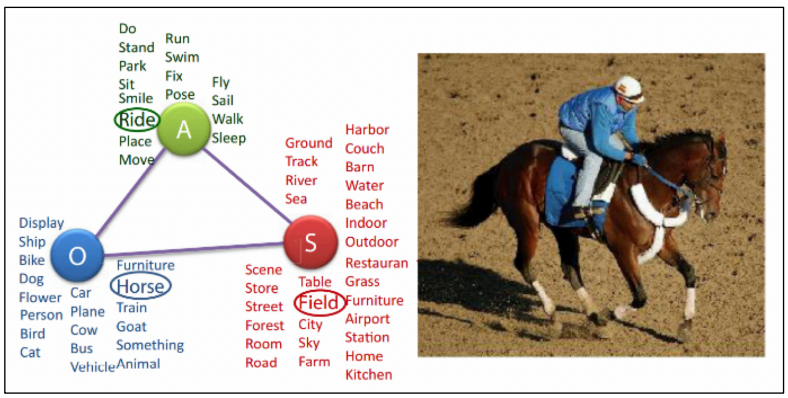
Tez çalışmasının ilk bölümünde geçmişten günümüze geleneksel yöntemler ve derin öğrenme tabanlı yapılan çalışmalara, ikinci bölüm olan yöntem kapsamında derin öğrenme tabanlı görüntü başlıklama çalışmalarında kullanılan Evrişimli sinir ağları, yinelemeli sinir ağları ve kelime temsil yöntemlerine değinilmiştir. Üçüncü bölümde ise tez çalışması kapsamında yapılan deneysel çalışmanın detaylarına yer verilmiştir. Son bölümde ise sonuçlar ve gelecek çalışmalarda uygulanabilecek önerilere yer verilmiştir.

1. İLGİLİ ÇALIŞMALAR

Görüntü başlıklama, yapay zekâ alanında çalışan araştırmacılar için oldukça popüler bir çalışma alanı haline gelmiştir. Bu çalışmalarda temel amaç görüntü üzerinden özellik çıkarımı yapılarak, bu özelliklerin semantik ve doğal dil kurallarına uygun bir şekilde otomatik olarak cümle üretmektir. Bu çalışmaları geleneksel makine öğrenmesi tabanlı ve derin öğrenme tabanlı olmak üzere iki kategoride incelemek mümkündür [1].

Derin öğrenme öncesi geleneksel makine öğrenme yöntemleriyle yapılan çalışmalarda, görüntü üzerindeki obje, sahne ve olay gibi özellikleri çıkartmak için Destek Vektör Makineleri (Support Vector Machine, SVM) gibi görüntü sınıflandırma algoritmaları kullanılmıştır.

Geleneksel makine öğrenmesi ile yapılan çalışmaları şablon tabanlı (template based) ve getirme tabanlı (retrieval based) olarak iki şekilde inceleyebiliriz. Şablon tabanlı görüntü başlıklama çalışmalarında ilk olarak belirli sayıda şablonlar oluşturulur. Görüntü üzerinden tespit edilen obje, sahne ve olay gibi örüntüler çıkartılarak önceden oluşturulan şablondaki boşluklar doldurulur ve görüntüye uygun başlık üretme sağlanır. Farhadi ve diğerleri [2], görüntüyü anlamlandırmak için çeşitli sınıflandırma algoritmaları kullanarak obje, sahne ve olay üçlü gösterimini oluşturmuştur (Şekil 1.1). Oluşturulan üçlü gösterim ile önceden hazırlanan şablonlardaki boşluklar doldurularak görüntü için başlık üretme yapmıştırlar.

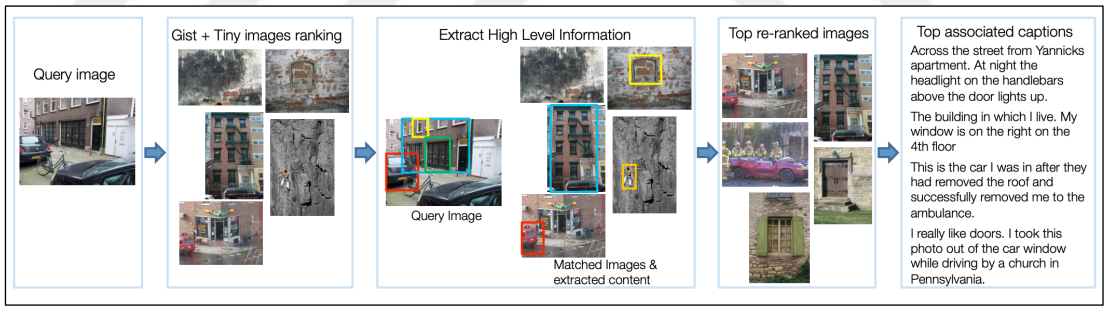


Şekil 1.1. Obje, sahne ve olay üçlü gösterimi [2]

Li ve diğerleri tarafından önerilen n-gram tabanlı çalışmada [3], görüntü üzerinden obje, görüntü nitelikleri ve bunların ilişkileri çıkartarak verilen görüntü için başlık üretme yapmışlardır.

Getirme tabanlı görüntü başlıklama çalışmalarında [4-7], ilk olarak görüntü-başlık havuzu oluşturulur [1]. Başlıklanmak istenen görüntünün, oluşturulan havuzdaki görüntüler ile benzerliği hesaplanarak belirli sayıda başlık bu havuz içerisinden alınır. Sonrasında çeşitli skorlama teknikleri kullanılarak alınan başlıklar için sıralama oluşturulur. Sıralamada en yüksek skoru elde eden başlık, modele girdi olarak verilen görüntünün başlığı kabul edilir.

Ordonez ve diğerleri [4], kapsamlı bir Flickr veri kümesi ile yaptıkları çalışmada, ilk olarak başlıklanmak istenen görüntünün veri kümesindeki benzer görüntülerini bulmuşlardır. Alınan bu görüntülerden, görüntüyü tanımlayan çeşitli bilgi çıkarımı sağlamışlar. Çıkartılan bu bilgiler ve görüntüler skorlama algoritmasına verilerek bir sırala elde etmişler. En iyi sıralamaya sahip ilk 4 görüntünün başlığı, modele girdi olarak verilen görüntünün muhtemel başlıkları olarak kabul edilmiştir (Şekil 1.2).

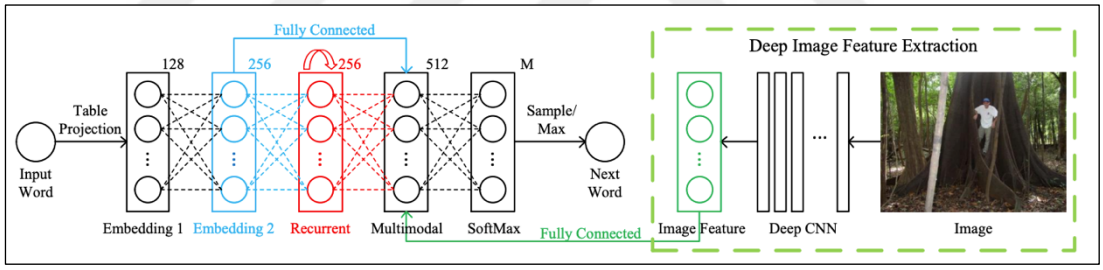


Şekil 1.2. Ordonez ve diğerlerinin önerdiği model akışı [4]

Şablon ve getirme tabanlı görüntü başlıklama çalışmaları her ne kadar dil bilgisine uygun sonuçlar üretse de çıkan bu sonuçlar sınırlı olduğu için görüntüyü tam olarak tanımlamakta yetersiz kalmıştır [1].

Derin öğrenme çalışmalarının hız kazanması ve alınan başarılı sonuçlar ile birlikte diğer çalışma alanlarında olduğu gibi görüntü başlıklama çalışmalarını da derin öğrenmeye yöneltmiştir. Derin öğrenme tabanlı yapılan ilk görüntü başlıklama çalışması, 2014 yılında Kiros ve diğerleri [8] tarafından, MNLM (Multimodal Neural Language Models) tekniği öne sürülerek yapılmıştır. Bu model temel olarak,

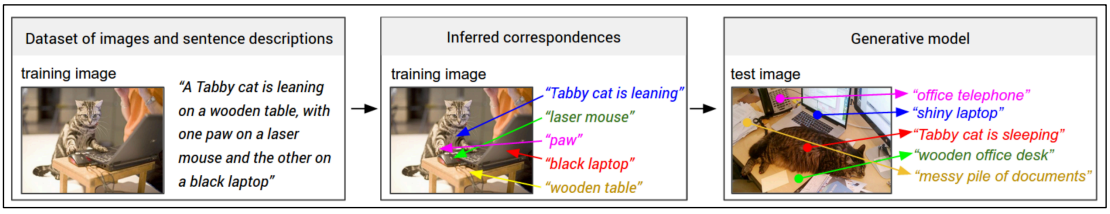
görüntünün özellik vektörünün çıkartıldığı görüntü kodlayıcı, başlık özellik vektörünün çıkartıldığı dil kodlayıcı, çıkartılan bu özelliklerin birlikte öğrenilmesini sağlayan çok modlu uzay (multimodal space) ve kod çözücü olmak üzere 4 bileşenden oluşmaktadır. Çok modlu uzay bileşeninde, görüntü ve başlıktan çıkartılan özellik vektörlerinin derin öğrenme modeli ile birleşik öğrenilmesi sağlanır. Kod çözücü bileşeninde ise çok modlu uzayda öğrenilen bilgiler doğal dile aktarılır. Bu sayede modelin görüntü üzerinden başlık üretimi sağlanır. Bu çalışma, önceki çalışmaların aksine herhangi bir şablon, kural ya da kısıta bağlı olmadan başlık üretilmesini sağlamıştır [1]. Kiros ve diğerlerinin yaptıkları bu çalışmayı [8], Mao ve diğerleri [9] genişleterek m-RNN (multimodal Recurrent Neural Network) mimarisini geliştirdiler. Geliştirdikleri mimaride (Şekil 1.3), önceki çalışmalardan farklı olarak kelime kalıplama (word embedding) katmanında önceden hesaplanan parametreleri kullanmak yerine, parametreler rastgele belirlenerek, model eğitimi esnasında bu parametrelerin kendi kendine optimize edilmesi sağlanmıştır. Bu da aldıkları sonuçlara olumlu yönde etki etmiştir [9].



Şekil 1.3. m-RNN mimarisi [9]

Kısa bir süre sonra Kiros ve arkadaşları, önerdikleri MNLM modelini genişleterek, SC-NML (Structure-Content Neural Language Model) [10] mimarisini önermiştir. Makine çevirisindeki kodlayıcı – kod çözücü mimarisinden ilham alan Kiros ve diğerleri, görüntü başlıklama probleminde de bu mimari kullanmıştırlar.

Karpathy ve diğerleri yaptıkları çalışmada [11], DeepVS (Deep Visual-Semantic) modelini önermiştirler. Bu modelde ilk olarak görüntü üzerinde farklı bölgeler tanımlanmıştır. Sonrasında tanımlanan her bir bölgeye özel açıklama üretme işlemi gerçekleştirmişler (Şekil 1.4).



Şekil 1.4. DeepVS model genel yaklaşımı [11]

Vinyals ve arkadaşları [12], Stokastik Gradyan İnişi (Stochastic Gradient Descent, SGD) ile baştan sonra eğitilebilen NIC (Neural Image Captioning) modelini önermiştirler. Önerdikleri modelde Denklem (1.1)'deki gibi;

$$\theta^* = \arg\max_{\theta} \sum_{(I,S)} \log p(S|I;\theta) \tag{1.1}$$

kayıp fonksiyonunu belirlemiştirler. θ model parametrelerini, I görüntü vektörünü, S ise görüntü başlığını ifade etmektedir. Bu denklemde model eğitimi esnasında θ parametresi optimize edilerek modelin optimize edilmesi amaçlanmıştır. Önerilen NIC modeli Denklem (1.2), (1.3) ve (1.4) gibi gösterilmiştir;

$$x_{-1} = CNN(I) \tag{1.2}$$

$$x_t = W_e S_t, \qquad t \in \{0 \dots N-1\} \tag{1.3}$$

$$p_{t+1} = LSTM(x_t), \qquad t \in \{0 \dots N-1\} \tag{1.4}$$

Görüntü kodlayıcı bileşeninde CNN modelinden çıkan görüntü özellikleri x_{-1} vektörü ile gösterilmiştir. Eğitim sırasında t anında modele verilen kelime S_t, kelime vektör matrisi ise W_e olarak belirtilmiştir. LSTM (Long Short-Term Memory) modelinden bir sonraki kelime tahmini için çıkan olasılık matrisi ise p_{t+1} ile belirtilmiştir. Önerilen bu modelde, görüntü ve dil işleme için son zamanlardaki en başarılı alt modeller birleştirmiştir. Bu sayade önceden eğitilmiş görüntü ve kelime öznitelik vektörlerini oluşturan hazır modeller, transfer öğrenme tekniği ile kullanılabilir hale getirilmiştir. Değerlendirme ölçütlerinden biri olan BLUE skorlamada %59'luk başarı ile oldukça başarılı bir model olduğunu göstermiştir.

Vinyals ve arkadaşlarının önerdikleri modelde [12], görüntünün kodlayıcıdan çıkan özellik vektörü, her iterasyonda kod çözücüye aynı şekilde verilmekteydi. Ancak

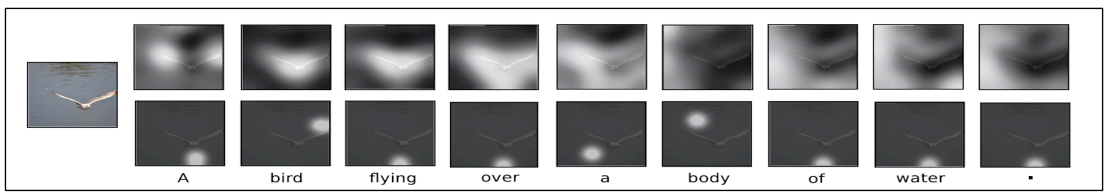
görüntüden çıkan özellik vektörünün tamamı, başlık içerisindeki kelimelerin hepsi aynı oranda ilişkili olmadığından model başarısını düşürmekteydi. Bu kısıtı aşmak için Xu ve arkadaşları yaptıkları çalışmada [13], kodlayıcı–kod çözücü mimarisi ile dikkat mekanizmasını birleştiren bir model önermiştirler. Önerdikleri dikkat mekanizmalı mimaride, görüntüyü dinamik olarak farklı parçalara ayırmıştırlar. Başlık içerisindeki kelimelerin, hangi parça ile ilişkili olduğunu modele öğreterek, modelin her adımda görüntünün farklı parçalarına odaklanmasını sağlamıştırlar. Xu ve diğerleri, dikkat mekanizmasını, Denklem (1.5), (1.6) ve (1.7)'deki gibi;

$$\mathrm{a} = \{\mathrm{a}_1, \ldots, \mathrm{a}_\mathrm{L}\}, \quad \mathrm{a}_\mathrm{i} \in \mathbb{R}^\mathrm{D} \tag{1.5}$$

$$\sum_{i=1}^{L} a_{ti} = 1 \tag{1.6}$$

$$z_t = \sum_{i=1}^{L} a_{ti}\mathrm{a}_\mathrm{i} \tag{1.7}$$

ifade etmiştirler. Evrişimsel sinir ağı modeli ile görüntüyü farklı parçalara ayırarak, Denklem (1.5)'teki gibi LxD boyutunda vektör elde etmiştirler. Buradaki her bir a_i vektörü görüntü üzerindeki farklı parçayı ifade etmektedir. Dikkat mekanizması da her a_i bölgesi için bir a_{ti} ağırlığı üretir. Bu ağırlık *t* zamanında, görüntünün *i.* bölgesi için ne kadar anlamlı olduğunu ifade etmektedir. Sonrasında elde edilen bu bilgiler ile z_t durum vektörü oluşturulmuştur. Durum vektörü kod çözücüye verilerek, her adımda görüntünün farklı bölgelerine odaklanılması sağlanmıştır. Şekil 1.5'te dikkat mekanizmasının her iterasyonda kelime üretilirken, görüntü üzerindeki farklı bölgelere odaklanılması gösterilmiştir.



Şekil 1.5. Dikkat mekanizmasının görüntü üzerindeki odaklanması [13]

Tablo 1.1'de Hossain ve diğerlerinin [1] hazırladığı tablo genişletilerek, görüntü başlıklama çalışmalarının yıllara göre veri kümesi ve değerlendirme metriklerinin özet bilgisi verilmiştir.

Tablo 1.1. Görüntü başlıklama çalışmalarının genel özeti [1]

Çalışma	**Yıl**	**Veri Kümesi**	**Skorlama yöntemleri**
Kiros ve diğ. [8]	2014	IAPR TC-12, SBU	BLEU, PPLX
Kiros ve diğ. [10]	2014	Flickr 8K, Flickr 30K	R@K, mrank
Mao ve diğ. [9]	2014	IAPR TC-12, Flickr 8K/30K	BLEU, R@K, mrank
Karpathy ve diğ. [14]	2014	PASCAL1K, Flickr 8K/30K	R@K, mrank
Mao ve diğ. [15]	2015	IAPR TC-12, Flickr 8K/30K, MS COCO	BLEU, R@K, mrank
Chen ve diğ. [16]	2015	PASCAL, Flickr 8K/30K, MS COCO	BLEU, METEOR, CIDEr
Fang ve diğ. [17]	2015	PASCAL, MS COCO	BLEU, METEOR, PPLX
Jia ve diğ. [18]	2015	Flickr 8K/30K, MS COCO	BLEU, METEOR, CIDEr
Karpathy ve diğ. [11]	2015	Flickr 8K/30K, MS COCO	BLEU, METEOR, CIDEr
Vinyals ve diğ. [12]	2015	Flickr 8K/30K, MS COCO	BLEU, METEOR, CIDEr
Xu ve diğ. [13]	2015	Flickr 8K/30K, MS COCO	BLEU, METEOR
Sugano ve diğ. [19]	2016	MS COCO	BLEU, METEOR, ROUGE, CIDEr
Mathews ve diğ. [20]	2016	MS COCO, SentiCap	BLEU, METEOR, ROUGE, CIDEr
Wang ve diğ. [21]	2016	Flickr 8K/30K, MS COCO	BLEU, R@K
Johnson ve diğ. [22]	2016	Visual Genome	METEOR, AP, IoU
Mao ve diğ. [23]	2016	ReferIt	BLEU, METEOR, CIDEr
Wang ve diğ. [24]	2016	Flickr 8K	BLEU, PPLX, METEOR
Tran ve diğ. [25]	2016	MS COCO, Adobe-MIT, Instagram	HE
Ma ve diğ. [26]	2016	Flickr 8k, UIUC	BLEU, R@K
You ve diğ. [27]	2016	Flickr 30K, MS COCO	BLEU, METEOR, ROUGE, CIDEr
Anne ve diğ. [28]	2016	MS COCO, ImageNet	BLEU, METEOR
Yao ve diğ. [29]	2017	MS COCO	BLEU, METEOR, ROUGE, CIDEr
Lu ve diğ. [30]	2017	Flickr 30K, MS COCO	BLEU, METEOR, CIDEr
Chen ve diğ. [31]	2017	Flickr 8K/30K, MS COCO	BLEU, METEOR, ROUGE, CIDEr
Gan ve diğ. [32]	2017	Flickr 30K, MS COCO	BLEU, METEOR, CIDEr
Ren ve diğ. [33]	2017	MS COCO	BLEU, METEOR, ROUGE, CIDEr
Park ve diğ. [34]	2017	Instagram	BLEU, METEOR, ROUGE, CIDEr
Wang ve diğ. [35]	2017	MS COCO, Stock3M	SPICE, METEOR, ROUGE, CIDEr
Tavakoli ve diğ. [36]	2017	MS COCO, PASCAL 50S	BLEU, METEOR, ROUGE, CIDEr
Liu ve diğ. [37]	2017	Flickr 30K, MS COCO	BLEU, METEOR
Gan ve diğ. [38]	2017	FlickrStyle10K	BLEU, METEOR, ROUGE, CIDEr
Dai ve diğ. [39]	2017	Flickr 30K, MS COCO	E-NGAN, E-GAN, SPICE, CIDEr
Shetty ve diğ. [40]	2017	MS COCO	HE, SPICE, METEOR
Liu ve diğ. [41]	2017	MS COCO	SPIDEr, HE
Gu ve diğ. [42]	2017	Flickr 30K, MS COCO	BLEU, METEOR, CIDEr, SPICE
Yao ve diğ. [43]	2017	MS COCO, ImageNet	METEOR
Rennie ve diğ. [44]	2017	MS COCO	BLEU, METEOR, CIDEr, ROUGE
Venugopalan ve diğ. [45]	2017	MS COCO, ImageNet	METEOR
Zhang ve diğ. [46]	2017	MS COCO	BLEU, METEOR, ROUGE, CIDEr
Wu ve diğ. [47]	2018	Flickr 8K/30K, MS COCO	BLEU, METEOR, CIDEr
Aneja ve diğ. [48]	2018	MS COCO	BLEU, METEOR, ROUGE, CIDEr
Wang ve diğ. [49]	2018	MS COCO	BLEU, METEOR, ROUGE, CIDEr
Feng ve diğ. [50]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Shuster ve diğ. [51]	2019	Flickr 30K, MS COCO	BLEU, ROUGE, CIDEr, SPICE
Li ve diğ. [52]	2019	MS COCO	METEOR, CIDEr, SPICE
Yang ve diğ. [53]	2019	Visual Genome, MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Yin ve diğ. [54]	2019	Visual Genome	AP, METEOR
Qin ve diğ. [55]	2019	Visual Genome, MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Gao ve diğ. [56]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Zheng ve diğ. [57]	2019	MS COCO	METEOR, CIDEr, SPICE
Wang ve diğ. [58]	2019	MS COCO	BLUE, CIDEr, LSA
Dognin ve diğ. [59]	2019	MS COCO	METEOR, CIDEr
Guo ve diğ. [60]	2019	FlickrStyle10K, MS COCO	BLUE, METEOR, CIDEr, PPLX

Tablo 1.1. (Devam) Görüntü başlıklama çalışmalarının genel özeti [1]

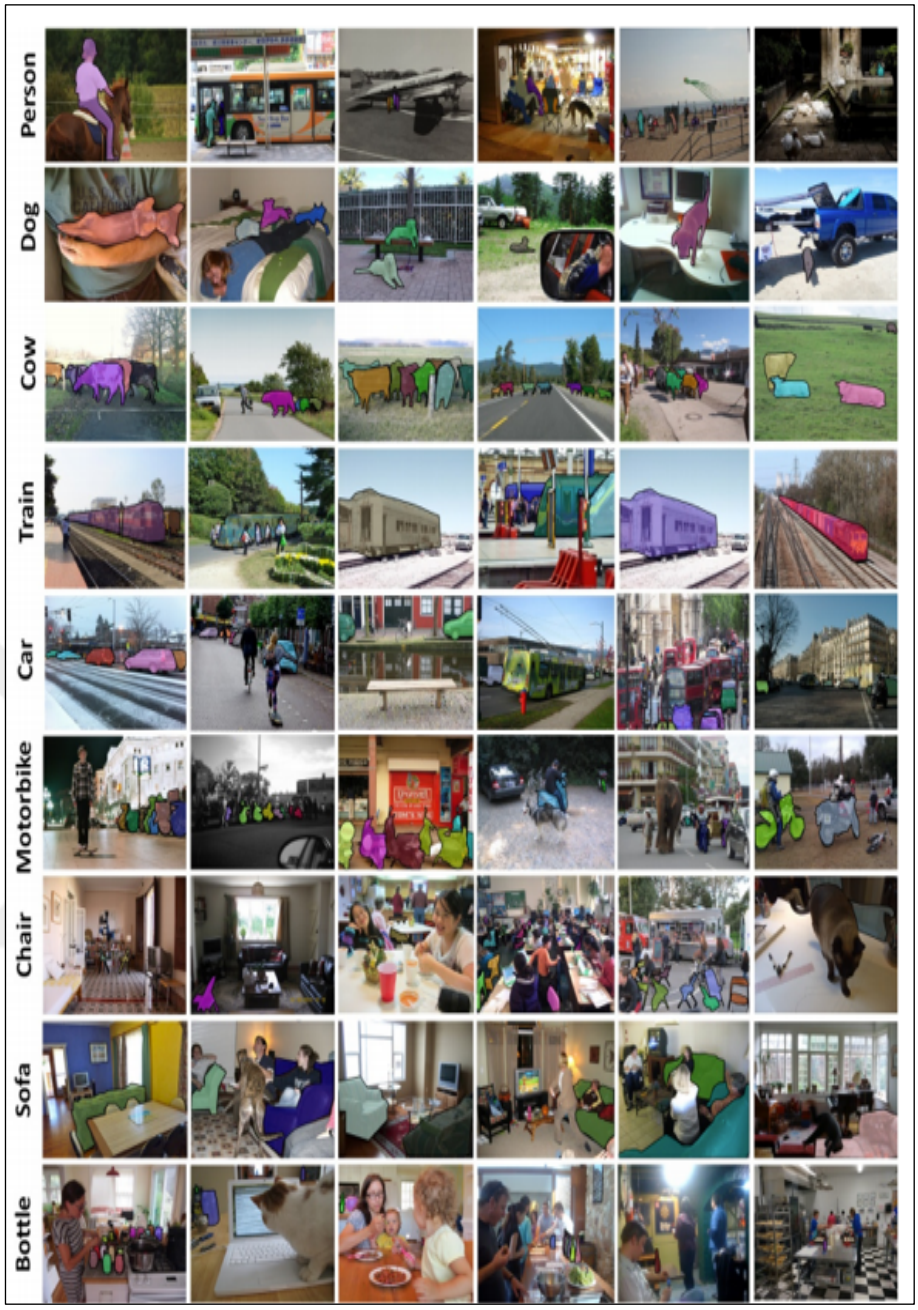
Çalışma	Yıl	Veri Kümesi	Skorlama yöntemleri
Deshpande ve diğ. [61]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Biten ve diğ. [62]	2019	GoodNews	BLEU, METEOR, ROUGE, CIDEr, SPICE
Kim ve diğ. [63]	2019	Visual Genome	METEOR, AP
Cornia ve diğ. [64]	2019	Flickr 30K, MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Li ve diğ. [65]	2019	Visual Genome, MS COCO	AP
Chen ve diğ. [66]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Fan ve diğ. [67]	2019	Flickr 30K, MS COCO	BLEU, METEOR, ROUGE, CIDEr
Demirel ve diğ. [68]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE, F1
Sammani ve diğ. [69]	2019	MS COCO	BLEU, ROUGE, CIDEr, SPICE
Khare ve diğ. [70]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Yan ve diğ. [71]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Yao ve diğ. [72]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Li ve diğ. [73]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Huang ve diğ. [74]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Ke ve diğ. [75]	2019	Visual Genome, MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Yang ve diğ. [76]	2019	Visual Genome, MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Herdade ve diğ. [77]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Huang ve diğ. [78]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Chen ve diğ. [79]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Liu ve diğ. [80]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Liu ve diğ. [81]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Wang ve diğ. [82]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr
Yan ve diğ. [83]	2019	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Jiang ve diğ. [84]	2019	Flickr 8K/30K, MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Zhao ve diğ. [85]	2020	MS COCO, SentiCap, FlickrStyle10K	BLEU, METEOR, PPLX, CLS
Zhou ve diğ. [86]	2020	Flickr 30K, MS COCO	BLEU, METEOR, CIDEr
Wang ve diğ. [87]	2020	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Seo ve diğ. [88]	2020	Conceptual Captions	HE
Liu ve diğ. [89]	2020	MS COCO	BLEU, METEOR, ROUGE, CIDEr
Cao ve diğ. [90]	2020	MS COCO	METEOR, ROUGE, CIDEr, SPICE, F1
Hou ve diğ. [91]	2020	MS COCO	BLEU, METEOR, CIDEr, SPICE
Zhang ve diğ. [92]	2020	Instagram, YFCC100M	BLEU, METEOR, ROUGE, CIDEr, SPICE
Guo ve diğ. [93]	2020	MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Chen ve diğ. [94]	2020	Visual Genome, MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Shi ve diğ. [95]	2020	Visual Genome, MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE
Alikhani ve diğ. [96]	2020	Conceptual Captions	HE, F1
Yi ve diğ. [97]	2020	Flickr 8K, MS COCO	BLEU, METEOR, ROUGE, CIDEr, SPICE

Tablo 1.2'de literatür önemli görülen görüntü başlıklama çalışmalarının BLUE skor değerlendirmesi ile alınan sonuçlar verilmiştir. Xu ve diğerleri [13], çalışmalarında kodlayıcı – kod çözücü mimarisi üzerinde, soft ve hard olmak üzere iki farklı dikkat mekanizması kullanmışlardır. Soft dikkat mekanizması, hard dikkat mekanizmasına göre daha az başarım sağlamış olmasına rağmen, uygulamada gerçekleştirmesi daha kolaydır. Pedersoli ve diğerleri [100], çalışmalarında dikkat mekanizmasını, görüntü bölgeleri, başlıktaki kelimeler ve RNN durumu ile modelleyerek oldukça başarılı sonuç elde etmiştir.

Tablo 1.2. MS COCO veri kümesi ile yapılan çalışmalarının başarımları [1]

Çalışma	**B@1**	**B@2**	**B@3**	**B@4**	**METEOR**	**ROUGE-L**	**CIDEr**
Xu ve diğ. 2015 [13], soft	0,707	0,492	0,344	0,243	0,239	-	-
Xu ve diğ. 2015 [13], hard	0,718	0,504	0,357	0,25	0,23	-	-
Jin ve diğ. 2015 [98]	0,697	0,519	0,381	0,282	0,235	0,509	0,838
Wu ve diğ. 2016 [99]	-	-	-	0,29	0,237	-	0,886
Pedersoli ve diğ. 2017 [100]	-	-	-	0,307	0,245	-	0,938

Görüntü başlıklama çalışmalarında genel olarak Flickr8k/30k [101], MS COCO [102], STAIR (Software Technology and Artificial Intelligence Research), PASCAL (Pattern Analysis, Statistical Modelling and Computational Learning) gibi çeşitli veri kümeleri kullanılmıştır. İngilizce olarak hazırlanan bu veri kümeleri, her bir görüntüye karşılık beş farklı başlık içermektedir. Bunların yanı sıra, tek tip amaca uygun olan Görsel Genom [103, 104], IAPRTC-12 (International Association for Pattern Recognition Technical Committee 12) [105], Lifelog [106] gibi farklı veri kümeleri de bu çalışmalarda yerini almıştır [107]. MS COCO veri kümesinin içerdiği örnek görüntüler ve görüntü üzerinde eşlendiği objeler Şekil 1.6'da verilmiştir.



Şekil 1.6. MS COCO veri kümesinden örnekler [102]

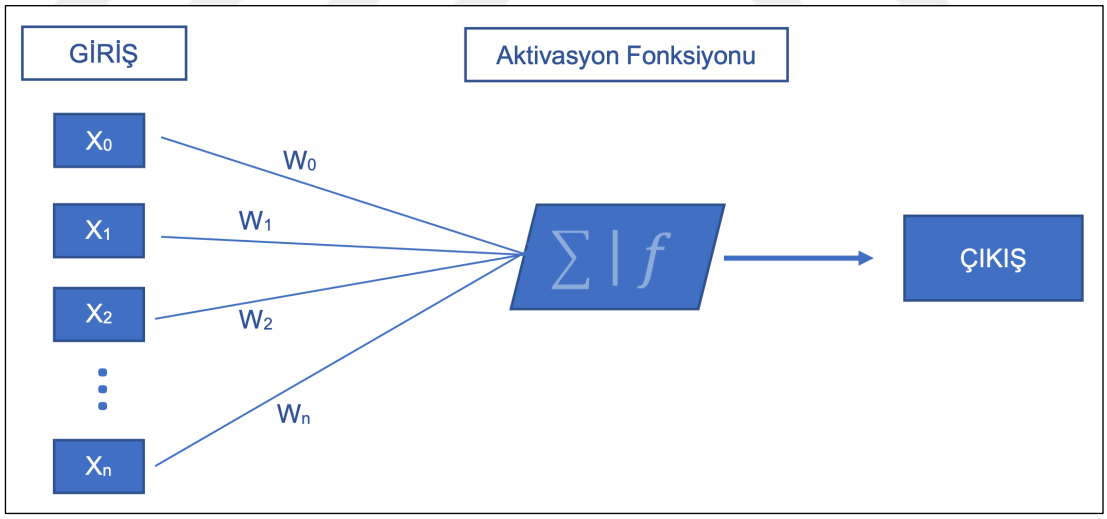
Literatürde, görüntü başlıklama çalışmalarının başarısını ölçmek için makine çevirisi çalışmalarında da kullanılan BLUE [108] ve METEOR [109], metin özetleme çalışmaları için kullanılan ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [110], görüntü başlıklama problemleri için geliştirilen CIDEr [111] ve SPICE (Semantic Propositional Image Caption Evaluation) [112] gibi farklı değerlendirme yöntemleri kullanılmıştır. Bu yöntemler temel olarak n-gram bazında modelin ürettiği metin ile veri kümesindeki metini karşılaştırarak modelin başarını ölçmeyi sağlar.

2. YÖNTEM

Derin öğrenme bazlı görüntü başlıklama çalışmalarında temel olarak kodlayıcı ve kod çözücü mimarisi kullanılmaktadır. Görüntülerin özellik vektörlerini çıkartmak için Evrişimli sinir ağları, metinlerin vektörel temsili için kelime kalıplama yöntemleri kullanılmaktadır. Görüntü ve metinden çıkartılan bu özellik vektörleri ile sıralı öğrenme gerçekleştirmek için tekrarlayan sinir ağları kullanılmaktadır. Bu bölümde görüntü başlıklama çalışmalarında kullanılan yöntemlerin detaylarına yer verilmiştir.

2.1. Yapay Sinir Ağları

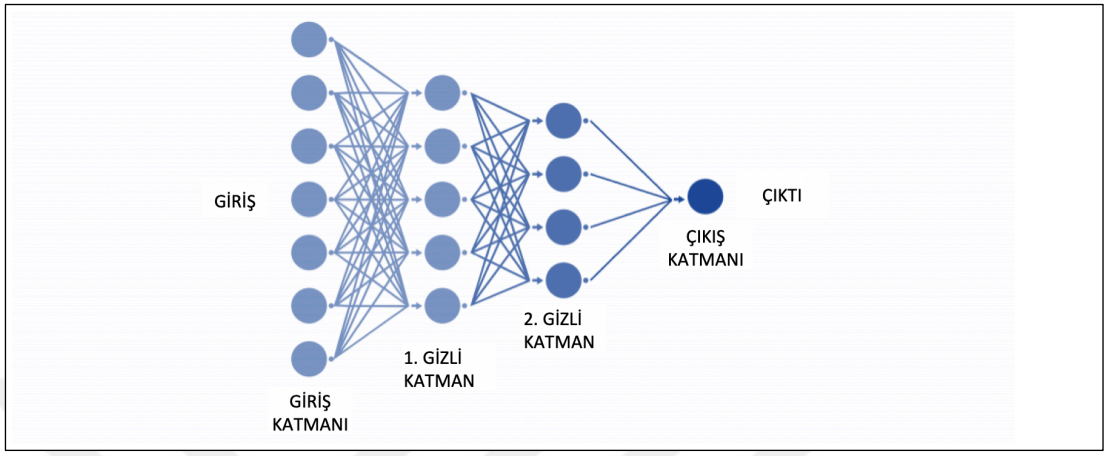
Yapay sinir ağları insan beynindeki nöronların çalışmasından ilham alınarak geliştirilen ve bilgisayarların da bu şekilde öğrenmesini amaçlayan, makine öğrenmesinin bir alt alanıdır. Bir yapay sinir ağı giriş katmanı, gizli katman ve çıkış katmanı olmak üzere 3 tip katman içerebilir ve her bir katman algılayıcılardan oluşur.



Şekil 2.1. Algılayıcı modeli

Basit bir algılayıcı modeli Şekil 2.1'de gösterilmiştir. Burada x_0, x_1, x_2, ..., x_n ile gösterilen giriş verileri, w_0, w_1, w_2, ..., w_2 ağırlıkları ile çarpılır. Çarpımdan çıkan sonuçlar toplanarak çıkış verisi üretmek için aktivasyon fonksiyonuna verilir.

Nöron ağlarında, derinlik kavramını gizli katman sayısı belirlemektedir. Çözülmesi amaçlanan problemin karmaşıklığına göre gizli katman sayısı artıp azalabilir. Şekil 2.2'de iki gizli katmanlı bir nöron ağı verilmiştir.



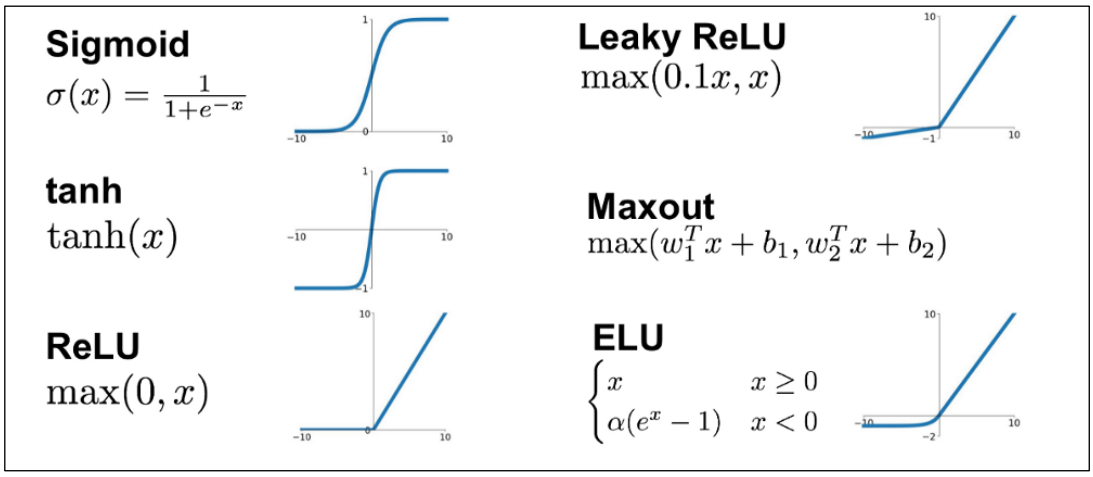
Şekil 2.2. İki gizli katmandan oluşan nöron ağı

Nöron ağı eğitimi sırasında, giriş verisi nöron ağından sırasıyla geçerek bir çıkış üretir. Çıkış verisi ile olması gereken gerçek veri karşılaştırılarak modelin hata oranı hesaplanır. Veri kümesindeki tüm veriler için hata oranı hesaplandıktan sonra ağ üzerinde geriye yayılım uygulanır. Geriye yayılım esnasında algılayıcıların üzerindeki ağırlıklar güncellenerek modelin hata oranı düşürülmeye çalışılır. Model eğitimi, hata oranı istenilen seviyeye gelene kadar ya da belirli bir iterasyona kadar devam eder.

2.1.1. Aktivasyon fonksiyonu

Aktivasyon fonksiyonun temel amacı giriş verilerini belirli bir aralığa indirgemektir. Derin öğrenme çalışmalarının artmasıyla birlikte aktivasyon fonksiyonları kendi başına bir çalışma alanı olmuştur ve zamanla bir çok aktivasyon fonksiyonu önerilmiştir.

Modelin karmaşık problemleri çözebilmesi için gizli katmanlar üzerindeki aktivasyon fonksiyonları genellikle doğrusal olmayan denklemler ile kurulur. Gerçek dünyada model eğitimi için kullanılan görüntü, metin, ses gibi veri kümeleri her zaman doğrusal düzene sahip değildir. Bu nedenle derin öğrenme modelinin eğitim kümesini öğrenebilmesi ve anlam çıkartabilmesi için aktivasyon fonksiyonlarının doğrusal olmayan fonksiyonlar ile kurulması gerekmektedir. Literatürde yaygın olarak kullanılan aktivasyon fonksiyonları Şekilde 2.3'te grafikleri ile birlikte verilmiştir.



Şekil 2.3. Yaygın olarak kullanılan aktivasyon fonksiyonları [113]

2.1.2. Kayıp fonksiyonu

Kayıp fonksiyonları, modelinin ürettiği sonuçlar ile gerçek değerleri karşılaştırarak modelin başarısını ölçmek için kullanılır. Belirlenen bir kayıp fonksiyonu sonucunun, model eğitimi esnasında azaltılması amaçlanır. Bu sebeple bu fonksiyonlara amaç fonksiyonu da denilmektedir. Kayıp fonksiyonu modelin doğru yönde eğitilmesi için oldukça büyük bir öneme sahiptir. Çözülmek istenen problem ile uyumsuz olan bir kayıp fonksiyonu seçimi modelin başarısını olumsuz yönde etkiler [114].

2.1.2.1. Ortalama kare hatası

Ortalama kare hatası, oldukça yaygın kullanılan ve uygulanması kolay olan bir kayıp fonksiyonudur. Denklem (2.1)'de gibi gösterildiği gibi;

$$okh = \frac{1}{n}\sum_{i=0}^{n}(y_i - y_{tahmin_i})^2 \tag{2.1}$$

ortalama kare hatasını hesaplamak için ilk olarak gerçek sonuçtan modelin ürettiği sonuç çıkartılır. Çıkan sonucun karesi alınır ve bu işlem tüm veri kümesine uygulanıp çıkan sonuçların ortalaması alınarak hesaplanır.

2.1.2.2. Ortalama mutlak hata

Ortalama mutlak hata, Denklem (2.2)'de gösterildiği gibi;

$$omh = \frac{1}{n}\sum_{i=0}^{n} |y_i - y_{tahmin_i}| \quad (2.2)$$

ortalama kare hatasına benzer olarak, model tahmini ile gerçek sonucun farkı alınır ve çıkan farkların ortalamasının bulunması ile hesaplanır.

2.1.2.3. Negatif log olasılığı

Negatif log olasılığı yaygın olarak çoklu sınıflandırma çalışmalarında kullanılmaktadır. Denklem (2.3)'de gösteriliği gibi;

$$nlo = -\sum_{i=0}^{n} \hat{y}_i * \log(\hat{y}_{tahmin_i}) \quad (2.3)$$

negative log olasılığı fonksiyonunda, $\hat{y}$ one-hot vektorünü, $\hat{y}_{tahmin}$ ise modelin ürettiği ayrık olasık vektörünü temsil etmektedir.

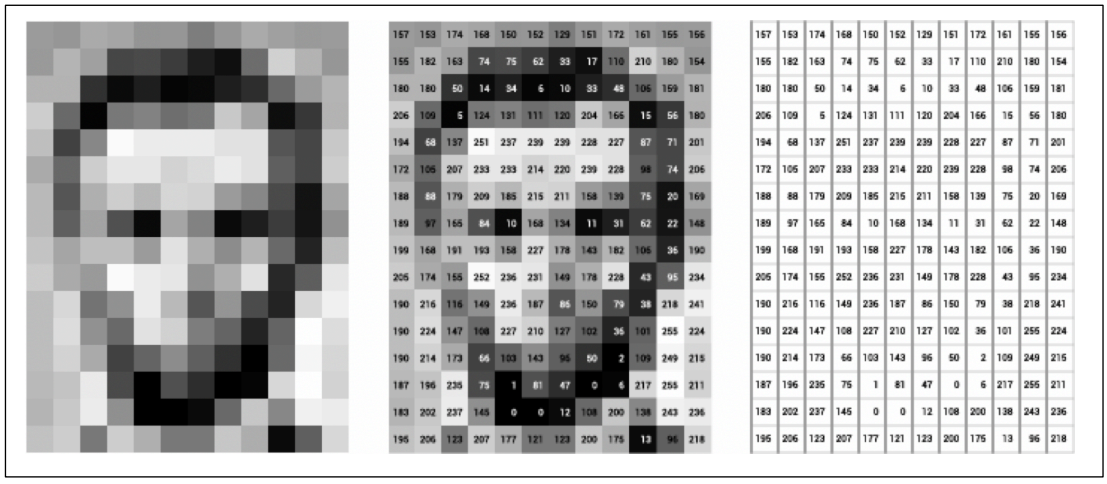
2.1.3. En iyileme algorimaları

Derin öğrenme algılayıcıların ağırlıklarının nasıl güncellenmesi gerektiğini en iyileme algoritmaları belirler. Model eğitilirken, geri yayılım esnasında kayıp fonksiyonunun türevine göre ağırlıkların artması ya da azalması belirlenir. Bu ağırlıkların ne kadar artıp azaltılacağı ise en iyileme algoritmaları ile belirlenmektedir. En iyileme algoritmasının seçimi, modelin doğru ve hızlı eğitilmesi için oldukça önemlidir. Zaman içerisinde farklı problemlerin daha hızlı ve başarılı bir şekilde çözülmesi için çeşitli en iyileme algoritmaları üzerinde çalışılmıştır. Derin öğrenme çalışmalarında en çok kullanılan en iyileme algoritmalarının başında Gradyan İnişi (Gradient Descent) yer almaktadır. Model eğitimi esnasında, her iterasyonda değerlendirilen veri kümesinin büyüklüğüne göre Yığın Gradyan İnişi (Batch Gradient Descent), Stokastik Gradyan İnişi (Stochastic Gradient Descent) ve Mini-Yığın Gradyan İnişi (Mini-batch Gradient Descent) olmak üzere üç farklı gradyan inişi bulunmaktadır [115]. Zaman içerisinde gradyan inişi methodunu temel alan, Adam, RmsProp (Root Mean Square Propagation), AdaGrad (Adaptive Gradient Algorithm) gibi çeşitli algoritmalar üzerinde çalışılmıştır.

2.2. Evrişimli Sinir Ağları

Evrişimli sinir ağları, görüntüler üzerinde çalışan çok katmanlı bir derin öğrenme mimarisidir. Görüntülerin pikselleri üzerinde çalışan evrişimli sinir ağı mimarisi ilk olarak 1989 yılında LeCun ve arkadaşları [116] tarafından önerilmiş olmasına rağmen yetersiz hesaplama gücü nedeniyle uzun bir süre işlevsiz olarak kalmıştır. 1998 yılında yine LeCun ve arkadaşları [117] tarafından bu mimari üzerindeki ilk başarılı çalışma gerçekleştirilmiştir. Günümüzdeki hesaplama teknolojileri üzerindeki çalışmalar ile birlikte önem kazanan evrişimli sinir ağları, birçok derin öğrenme temelli görüntü işleme çalışmalarında başarılı olarak kullanılmaktadır.

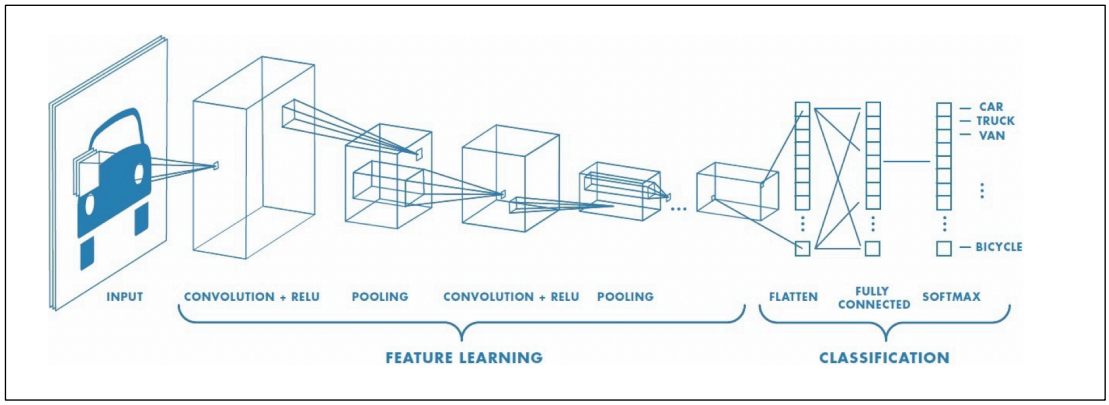
Görüntü işleme çalışmalarında, bilgisayarın görüntüleri tanımlayabilmesi için ilk olarak görüntünün matematiksel olarak temsil edilmesi gerekmektedir. Bu temsili çıkartmak için görüntü, en küçük birimi olan piksellerine ayrıştırılır. Piksek değerleri, görüntü üzerinde ifade ettiği bölgenin renk tonuna göre 0 ile 255 arasında bir değer alır. Görüntünün her bir genişlik ve yükseklik değeri için bir piksel değeri vardır. Dolayısıyla görüntüyü matematiksel olarak ifade etmek için bir piksel matrisi oluşturulur. Örnek bir görüntünün piksel matris temsili Şekilde 2.4'te verilmiştir.



Şekil 2.4. Görüntü temsili için oluşturulan örnek piksel temsili [118]

Evrişimli sinir ağları, genel olarak görüntü üzerindeki çizgi, köşe, kenar gibi öznitelikleri çıkartarak, görüntünün hangi sınıfa ait olunduğunun tespit edilmesi için kullanılır. Derin öğrenme ağı üzerindeki çeşitli katmanlar üzerinde yapılan işlemler sonucunda bu özniteliklerin çıkartılması sağlanır. Bu katmanlar genel olarak evrişim,

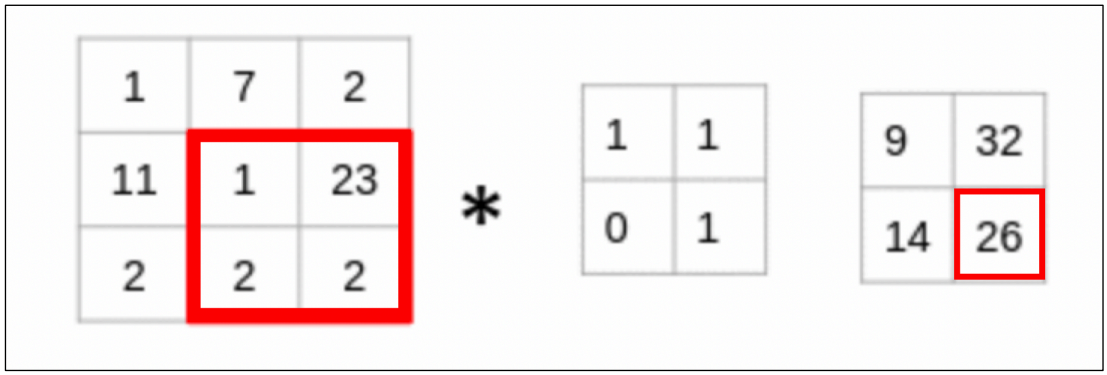
havuzlama, düzleme ve tam bağlanım katmanlarından oluşmaktadır. Örnek bir evrişimli sinir ağı katmanları Şekil 2.5'te gösterilmiştir.



Şekil 2.5. Evrişimli sinir ağı katmanları [119]

2.2.1. Evrişim katmanı

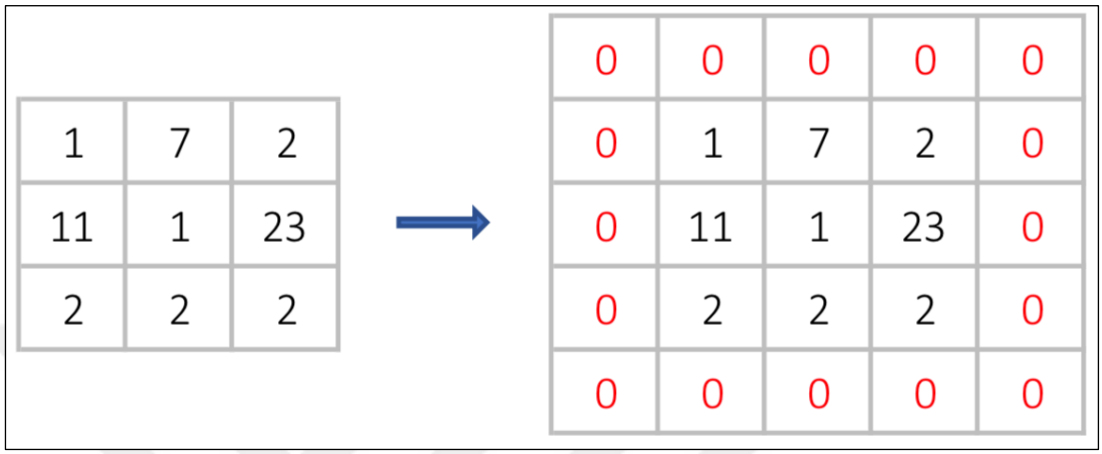
Evrişimli sinir ağında ilk gerçekleştirilen işlem evrişim işlemidir. Evrişim katmanında görüntü matrisi çeşitli filtrelerden geçirilir ve görüntü üzerindeki özniteliklerin çıkartılması sağlanır. Görüntü üzerinden çeşitli öznitelikleri çıkartmak için farklı filtreler uygulanmaktadır. Girdi olarak verilen görüntü matrisi üzerinde filtre matrisi sırayla gezdirilerek matris çarpımı gerçekleştirilir. Matris çarpımından çıkan sonuç, görüntü üzerindeki o bölgenin tekil değerini oluşturur. Örnek bir evrişim işlemi Şekil 2.6'da gösterilmiştir.



Şekil 2.6. Evrişim işlemi

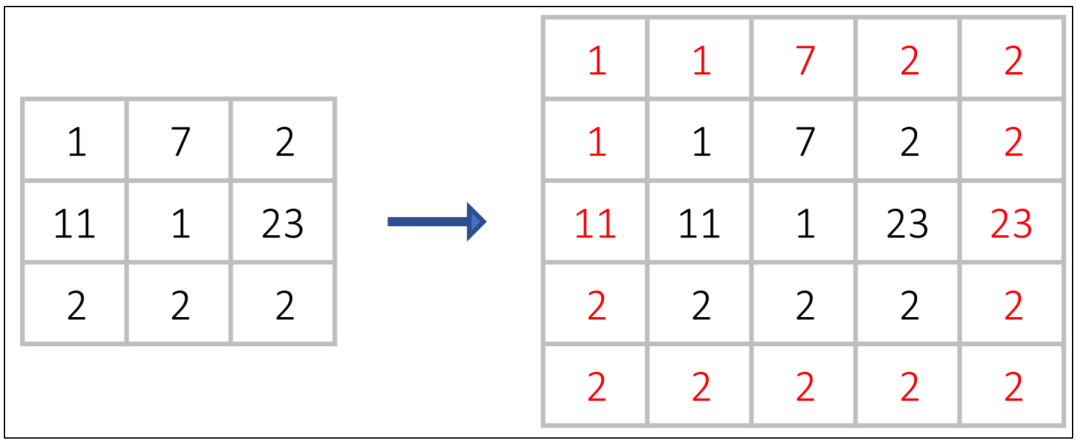
Evrişim işlemi sonucunda görüntü matrisinin boyutu azalır. Bu azalma katmanlarda her ne kadar daha kolay hesaplama sağlıyor olsa da görüntü içerisindeki bilginin azalmasına neden olmaktadır. Bilgi azalmasını engellemek için evrişim işlemi öncesinde, görüntü matrisine padding işlemi uygulanmaktadır. Padding işleminde,

görüntü matrisinin etrafı çeşitli değerler ile doldurularak, görüntü matrisinin boyutlarının arttılırması sağlanır. Zero padding ve same padding gibi farklı doldurma teknikleri bulunmaktadır. Zero padding yönteminde görüntü matrisinin etrafı sıfır değerleri ile doldurulur. Örnek bir zero padding uygulaması Şekil 2.7'de verilmiştir.



Şekil 2.7. Zero padding işlemi

Same padding tekniği ise matris çerçevesinde yer alan değerlerin aynısının matris üzerinde doldurulması ile gerçekleştirilir. Aynı değerlerin matris üzerine eklenmesi zero padding tekniğine göre daha az bilgi kaybı oluşmasını sağlar. Örnek bir same padding uygulaması Şekil 2.8'de gösterilmiştir.



Şekil 2.8. Same padding işlemi

2.2.2. Havuzlama katmanı

Evrişim işleminden sonra görüntü matrisleri havuzlama katmanına aktarılır. Bu katmanda görüntü matrisindeki bilgiler ortaklama işlemine tabii tutularak matris

boyutunun azaltılması sağlanır. Matris boyutunun azaltılması hem hızlı hesaplamaya yardımcı olur hem de derin öğrenme modelinin aşırı öğrenme eğilimini önler.

Havuzlama işlemi için ilk olarak belirli genişlik ve yükseklik değerleri seçilir. Seçilen bu değerler ile görüntü üzerinde sırayla gezilir ve uygulanan tekniğe göre o bölgenin özet bilgisi çıkartılır. Genel olarak 3 farklı havuzlama tekniğinden bahsedebiliriz. Bunlar maksimum, minimum ve ortalama havuzlama teknikleridir. Maksimum havuzlama tekniğinde, görüntü matrisi üzerinde seçilen alandaki en yüksek değer, minumum havuzlama tekniğinde en küçük değer alınır. Ortalama havuzlama tekniğinde ise seçilen bölgeninin matris elemanlarının ortalaması alınarak yeni matris bir temsili oluşturulur. Şekil 2.9'da maksimum, minimum ve ortalama havuzlama tekniklerinin örnekleri verilmiştir.

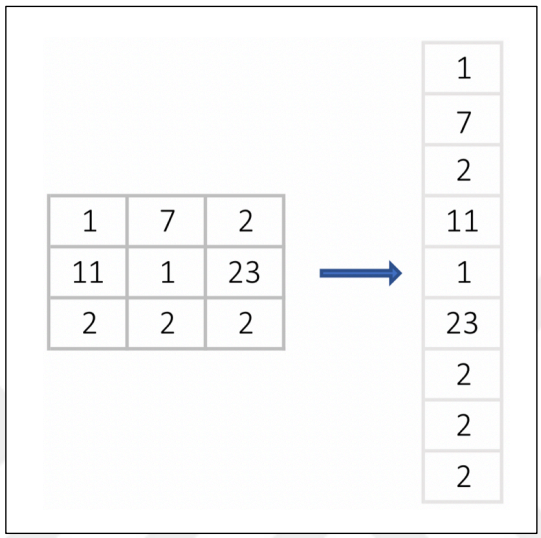


Şekil 2.9. Havuzlama teknikleri

2.2.3. Tam bağlanım katmanı

Evrişim katmanı ve havuzlama katmanı görüntülerden otomatik olarak öznitelik çıkartımını sağlar. Bu katmanlardan sonra gelen tam bağlanım katmanı, görüntüden çıkartılan özniteliklerin benzerlik durumuna göre desenlerini öğrenerek sınıflandırma işlemini gerçekleştirir.

Tam bağlanım katmanında, görüntü matrisleri üzerinden direkt olarak öğrenme işlemi gerçekleştirilememektedir. Bu yüzden havuzlama katmanından çıkartılan görüntü matrisi üzerinde ilk olarak düzleme işlemi gerçekleştirilir. Düzleme işlemi, $(n)\ x\ (m)$ boyutlu matrisi, $(n * m)\ x\ (1)$ boyutuna çevirerek, matristen vektör dönüşümünü sağlar. Örnek bir matris üzerindeki düzleme işlemi Şekil 2.10'da verilmiştir.



Şekil 2.10. Düzleme işlemi

Düzleme işleminden sonra tam bağlanım katmanına verilen öznitelik vektörü ile yapay sinir ağının ağırlıkları iteratif olarak güncellenerek modele sınıflandırma yeteneği kazandırılır.

Evrişimli sinir ağları ile görüntülerin özniteliklerinin otomatik olarak çıkartılması, pikseller arasındaki ilişkilerin ve desenlerin bulunması ile birlikte sınıflandırma problemlerinde oldukça başarılı sonuçlar elde edilmiştir. 1990'dan günümüze kadar evrişimli sinir ağları üzerinde birçok farklı mimari önerilmiştir. Önerilen bu mimarilerden ilki 1998 yılında LeCun ve arkadaşları [117] tarafından LeNet adını verdikleri mimari ile el yazımı olan rakamların görüntüleri üzerinden sınıflandırma işlemi için çalışmışlar.

2012 yılında Krizhevsky ve arkadaşları [120], AlexNet adını verdikleri mimari üzerinde çalışmıştırlar. LeNet mimarisinden çok daha fazla katman ve derinlik içeren AlexNet mimarisi, ILSVRC (ImageNet Large Scale Visual Recognition Challenge) yarışmasında %15.3 hata oranı ile o sıradaki en yüksek başarımı elde etmiştirler.

Szegedy ve arkadaşları tarafından [121], 2014 yılında önerilen GoogLeNet mimarisinde parametre sayısı (4 milyon), AlexNet mimarisindeki parametre sayısından (60 milyon) oldukça düşük olmasına rağmen hata oranını %6,67'ye düşürerek insan başarımına yakın bir başarı elde etmişler.

Simonyan ve arkadaşları VGGNet (Visual Geometry Group Network) mimarisini [122] yayınlamışlar. 16 ve 9 katmanlı olmak üzere iki farklı şekilde eğitilen VGGNet'in eğitilmiş parametrelerine açık olarak ulaşılabildiğinden bu alanda oldukça bilinen ve kullanılan modellerden birisi olmuştur.

Son olarak 2015 yılında He ve diğerleri [123] tarafından önerilen ResNet (Residual Neural Network) mimarisi %3,57 hata oranı ile insan performansını geçen bir başarım elde etmiştir.

Tablo 2.1. Evrişimli sinir ağı mimarileri

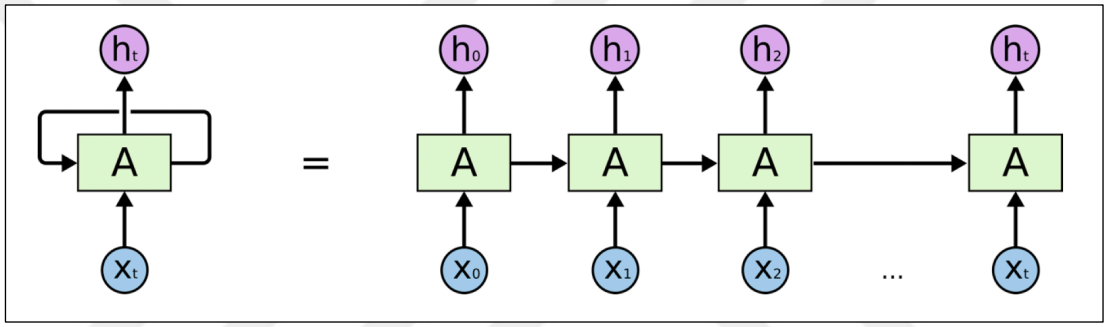
Yıl	**Mimari**	**ILSVRC Hata Oranı**	**Parametre Sayısı**
2012	LeNet [117]	-	60 bin
2012	AlexNet [120]	15,3	60 milyon
2014	GoogLeNet [121]	6,67	4 milyon
2014	VGGNet [122]	7,3	138 milyon
2015	ResNet [123]	3,37	60,3 milyon

Evrişimli sinir ağları üzerinde geliştirilen mimarilerin, parametre ve hata oranları Tablo 2.1' de özetlenmiştir.

2.3. Yinelemeli Sinir Ağları

Derin öğrenme ağlarında katmanlar arasında bilgi akışı sağlanırken sadece o sırada işlenen veriler üzerinden öğrenme gerçekleştirilir. Sıralı olan veriler, modele ayrı adımlarda verildiğinden bir önceki giriş verileri arasında anlam kurulamaz. Yinelemeli sinir ağları ise bu probleme çözüm olarak sunulmuş özel bir yapay sinir ağıdır. Yinelemeli sinir ağlarında video, metin, borsa rakamları gibi sıralı verilerle işlem yapmak için bir önceki iterasyonda modele verilen bilgiler durum bilgisi olarak tutulur. Durum bilgisi bir sonraki adımda modele giriş verisi olarak sağlanır. Bu sayede sıralı

veriler birbirleri arasındaki ilişkiyi de öğrenebilir hale gelmiştir. Örneğin İngilizce metinleri Türkçe'ye çevirme problemini düşünelim. Bu problemde İngilizce metin kelime kelime modele verilerek, sıralı bir şekilde Türkçe kelimelerin üretilmesi beklenir. Modele girdi olarak verilen İngilizce kelimeler tek başına cümlenin anlamını taşıyamaz, önceki ve sonraki kelimeler ile bir anlam oluşturur. Bu yüzden modelin önceki kelime bilgilerini de saklaması gerekir. Yinelemeli sinir ağları önceki durum bilgilerini saklayarak cümlenin ne ifade ettiğini öğrenebilir. Bir başka şekilde ifade edecek olursak, yinelemeli sinir ağlarınında durum bilgisini tutan bir bellek taşırlar. Bu bellek sıralı verilerde önceki giriş verilerini de tutarak buna göre bir çıktı üretirler. Temel yinelemeli sinir ağı modeli Şekil 2.11'de gösterilmiştir.



Şekil 2.11. Yinelemeli sinir ağı modeli [124]

Şekil 2.11'de gösterilen yinelemeli sinir ağı modelinde, x_t giriş verisini, A sinir ağı katmanlarını, h_t ise modelin çıktısını ifade etmektedir. A üzerinde bir döngü oluşturularak önceki verilerin bir sonraki adıma aktarılması sağlanır.

Yinelemeli sinir ağları; video sınıflandırma, konuşma tanıma, makine çevirisi, görüntü başlıklama gibi sıralı veri problemlerinde yaygın olarak kullanılmaktadır. Giriş ve çıkış veri sayısına göre bu ağlar aşağıdaki gibi 4 farklı şekilde sınıflandırılabilir.

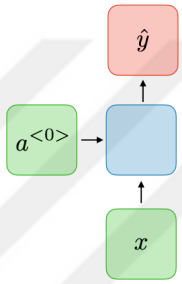
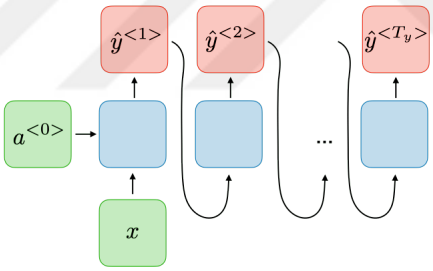
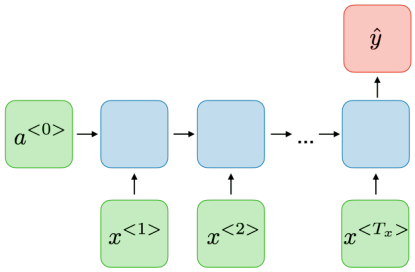
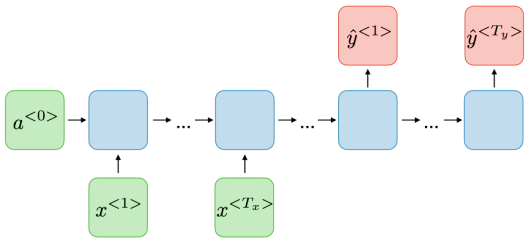
1. Bire Bir: Tek giriş verisine karşılık tek bir çıkış verisi üreten ağlardır. Bunlar klasik sinir ağlarıdır.
2. Bire Çok: Modele tek bir giriş verilip, birden fazla çıkış alınan ağ tipidir. Örneğin görüntü başlıklama çalışmalarında, modele görüntü verisi verilip çıkış olarak sıralı kelimelerden oluşan cümle beklenmektedir.
3. Çoka Bir: Birden fazla sıralı giriş verisine karşılık bir tane çıkış verisi üreten sinir ağlarıdır. Örneğin duygu sınıflandırma problemi için, modele sıralı kelimelerden

oluşan bir cümle verilir ve modelden çıktı olarak cümledeki duygunun sınıfını belirten bir çıktı beklenir.

4. Çoka Çok: Modele birden fazla sıralı verinin giriş olarak sağlanıp, birden fazla sıralı verinin çıktı olarak alındığı modellerdir. Bu modele en basit örnek olarak makine çevirisi problemini verebiliriz.

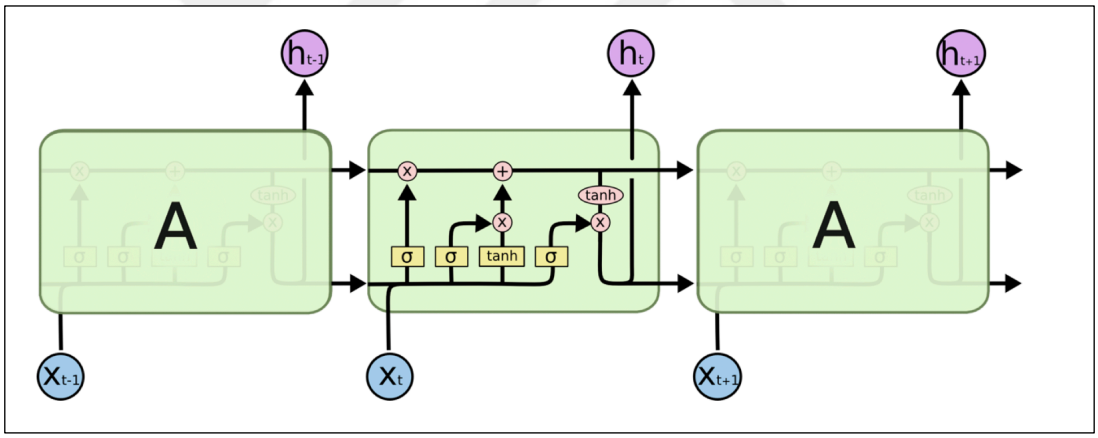
Yinelemeli sinir ağı tiplerinin mimarisi Tablo 2.2'de özetlenmiştir.

Tablo 2.2. Yinelemeli sinir ağı mimarileri [125]

RNN Tipi	**Model Gösterimi**	**Örnek Uygulaması**
Bire Bir		Klasik sinir ağı modeli
Bire Çok		Görüntü başlıklama
Çoka Bir		Duygu sınıflandırma
Çoka Çok		Makine çevirisi

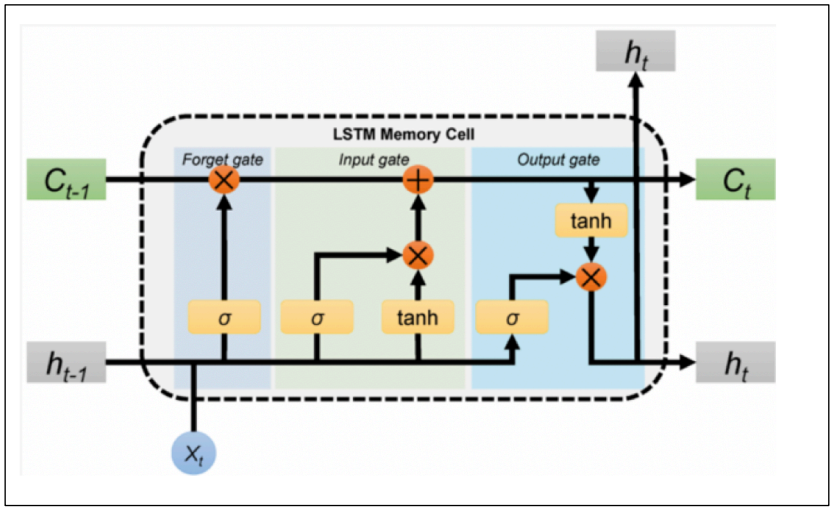
Yinelemeli sinir ağlarında, uzun geçmiş durumlarının hatırlanması teoride mümkün olsa da pratikte bunun için parametre seçimi oldukça zordur. Bu problemi, Bengio ve diğerleri [126], 1994 yılında yaptıkları çalışmada keşfetmişler. Yinelemeli sinir ağı modelinin eğitimi esnasında zamanla gradyanların çok büyük ya da çok küçük olması bu probleme neden olmaktadır. Geri yayılım sırasında maliyet fonksiyonu türevinin 0'a çok yakınsaması nedeniyle model eğitiminde gradyanlardaki değişim oldukça küçülür ve öğrenme durur. Bu problem gradyanların yok olması (vanishing gradient) olarak adlandırılmıştır. Bunun tam tersi ise gradyan patlamasıdır (exploding gradient). Gradyan patlaması, ağırlıkların giderek artması nedeniyle modelin aşırı öğrenmeye doğru eğilim göstermesidir.

Hochreiter ve diğerleri [127], yinelemeli sinir ağlarındaki uzun geçmişi hatırlayamama problemini çözmek için, özel bir yinelemeli sinir ağı modeli olan Uzun-Kısa Süreli Bellek mimarisini önermiştirler.



Şekil 2.12. LSTM mimarisi [124]

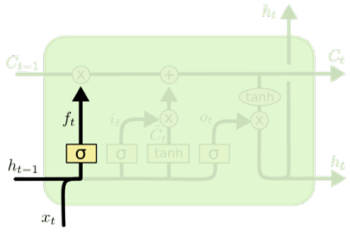
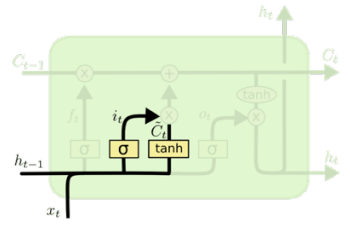
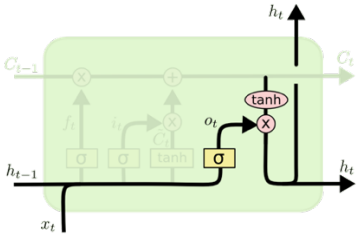
Klasik yinelemeli sinir ağı mimarisinde, tekrarlayan modül $tanh$ gibi basit bir katmandan oluşur. LSTM mimarisinde ise bu modül Şekil 2.12'de gösterildiği üzere daha karmaşık katmanlardan oluşan bir yapıya sahiptir [124]. Bu katmanlar, kapı (gate) olarak adlandırılmış. Giriş kapısı, unutma kapısı ve çıkış kapısı olmak üzere 3 farklı tipte kapı bulunmaktadır. Şekil 2.13'te LSTM hücre kapıları gösterilmiştir. Yine bu şekil üzerinde, C LSTM mimarisinin durum vektörünü, h ise model çıktısını ifade etmektedir. $t-1$ ile belirtilen değerler bir önceki durumu, t ise o andaki değerler için kullanılmaktadır.



Şekil 2.13. Hücre durum vektörü ve kapıları [128]

LSTM mimarisinde, C durum vektörü hücreler arasındaki bilgi akışını sağlar. Bilgi akışı esnasında durum vektörü, kapılar olarak adlandırılan katmanlar üzerinde çeşitli işlemlerden geçirilir. Tablo 2.3'te hücrenin kapıları ve burada gerçekleştirilen işlemlerin notasyonları verilmiştir.

Tablo 2.3. Hafıza hücre kapıları [124]

Hücre Kapısı	**Gösterim**	**Notasyon**
Unutma Kapısı		$f_t = \sigma\,(W_f.[h_{t-1}, x_t] + b_f)$
Giriş Kapısı		$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i)$ $\tilde{C}_t = \tanh(W_c.[h_{t-1}, x_t] + b_C)$
Çıkış Kapısı		$o_t = \sigma\,(W_o.[h_{t-1}, x_t] + b_o)$ $h_t = o_t * \tanh(C_t)$

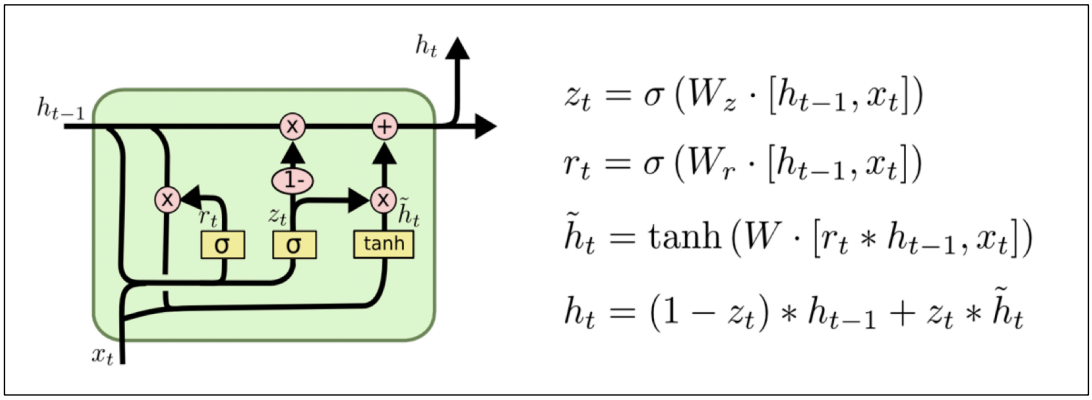
LSTM durum vektörüne uygulanan ilk işlem unutma kapısında gerçekleşir. Unutma kapısında, h_{t-1} ve x_t değerleri sigmoid fonksiyonuna verilir. Sigmoid fonksiyonu 1 ile 0 arasında bir değer üreterek durum vektörü üzerinden akan bilginin ne kadarınının sonraki aşamalara geçeceğine karar verir. Sigmoid fonksiyonundan 1 çıkması durumunda bilgilerin tamamının diğer aşamaya geçirileceğini, 0 çıkması durumunda ise bilgilerin unutularak diğer aşamaya aktarılmayacağını gösterir. Unutma kapısından çıkan f_t sonucu ile önceki durum vektörü olan C_{t-1} çarpılarak durum vektöründe unutma işlemi gerçekleştirilir.

LSTM mimarisinde ikinci aşama giriş kapısıdır. Giriş kapısında iki işlem gerçekleştirilir. Bu işlemler sonucunda durum vektörüne yeni bilgi aktarımı ve mevcut bilgilerin güncelleme işlemi gerçekleşir. İlk işlem olan sigmoid fonksiyonunda hangi mevcut bilgilerin güncelleneceği, ikinci işlem olan *tanh* fonksiyonunda ise hangi yeni verilerin durum vektörüne aktarılacağı kararı verilir. İlk işlem sonucunda i_t değerini, ikinci işlem sonucunda ise $\tilde{C}_t$ değerini elde ederiz. Bu değerlerin çarpımı ($i_t * \tilde{C}_t$) ve bir önceki durum vektörü C_{t-1} toplanarak yeni durum vektörü olan C_t elde edilir. Yeni durum vektörünü elde etmek için uygulanan işlem Denklem (2.4)'teki gibi;

$$C_t = C_{t-1} * f_t + i_t * \tilde{C}_t \tag{2.4}$$

gösterilmiştir [124]. LSTM mimarisindeki son aşama olan çıkış kapısında o andaki çıkış verisi üretilir. Diğer kapılarda olduğu gibi çıkış kapısında da ilk olarak sigmoid fonksiyonu uygulanır. Buradaki sigmoid fonksiyonu bize hangi bilgilerin çıktı olarak döneceğinin kararını verir. Daha sonra durum vektörü C_t, *tanh* fonksiyonundan geçirilerek değerleri -1 ile 1 arasına taşınır. Son olarak sigmoid fonksiyonundan çıkan sonuç ile *tanh* fonksiyonundan çıkan sonuç çarpılır ve çıkış değeri h_t üretilir [124].

Zaman içerisinde LSTM kapılarındaki işlemler değiştirilerek birçok tekrarlayan sinir ağ mimarileri önerilmiştir. Bunların en bilinen örneklerinden biri GRU (Gated Recurrent Unit)'dur. 2014 yılında Cho ve diğerleri [129] tarafından önerilen GRU mimarisinde, unutma kapısı ve giriş kapısı tek kapıda birleştirilerek güncelleme kapısı oluşturmuşlar. LSTM modeline göre daha basit olan bu mimarinin kullanımı gün geçtikçe artmaktadır [124]. GRU modelinin yapısı ve kapılar üzerinde gerçekleştirilen işlemler Şekil 2.14'te verilmiştir.



Şekil 2.14. GRU mimarisi ve kapılar üzerinde gerçekleştirilen işlemler [124]

Jozefowicz ve diğerlerinin [130], GRU ve LSTM varyantlarını karşılaştırmak için yaptıkları çalışma, GRU mimarisinin dil modelleme haricindeki diğer çalışma alanlarında daha başarılı sonuçlar elde ettiğini göstermiştir.

2.4. Kelime Temsil Yöntemleri

Metinler üzerinde yapılan makine öğrenmesi çalışmalarında, metinlerin modelin tanımlayabileceği bir formata çevrilmesi gerekmektedir. Metin verilerinin sayısal vektörlere çevrilme işlemleri kelime kalıplama yöntemleri ile yapılmaktadır. Literatüre baktığımızda birçok kelime kalıplama yöntemi bulunmaktadır. Bu yöntemler temelinde metinlerin sayısal vektör olarak nasıl ifade edildiğine göre farklılık göstermektedir.

Doküman sınıflandırma, duygu analizi, metin özetleme gibi birçok çalışma alanında kullanılan kelime kalıplama yöntemleri temelde geleneksel makine öğrenme tabanlı ve derin öğrenme tabanlı olmak üzere ikiye ayrılmaktadır. Geleneksel makine öğrenme tabanlı kelime kalıplama yöntemleri kısa metinler üzerinde hızlı çalışmasına rağmen büyük metin verilerinde başarısız sonuçlar elde etmektedir. Derin öğrenme temelli kelime kalıplama yöntemleri ise büyük metin verileri ile daha hızlı ve başarılı olmuştur.

2.4.1. Kelime çantası yöntemi

Kelime çantası yöntemi dokümanların analizi için ortaya çıkmış bir yöntemdir [131]. Oldukça basit olan bu yöntemin temelinde, kelimelerin doküman üzerinde geçme sayısına göre bir vektör çıkartılma işlemidir. Bunun için ilk olarak doküman üzerindeki

eşsiz kelime listesi çıkartılır ve bu liste sıraya dizilir. Sonrasında liste içerisindeki kelimeler sırasıyla metin üzerinde kontrol edilir. Metin içerisinde geçen kelime için 1, geçmeyen kelime için ise 0 değeri yazılarak vektör oluşturulur.

Örneğin iki cümleden oluşan doküman listemiz olsun;

İlk cümle: Apple Türkiye garantili siyah iPhone

İkinci cümle: Garantili beyaz iPhone

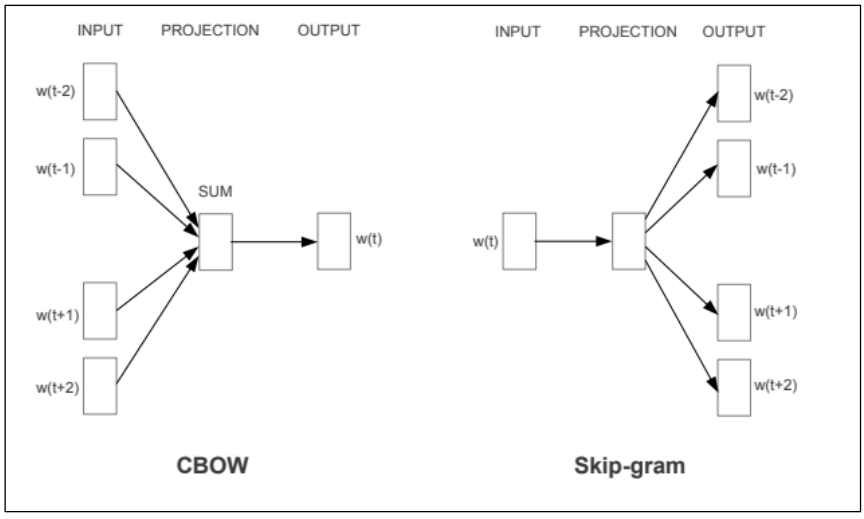
Bu cümlelerden elde edilen eşsiz kelime listesi, [apple, beyaz, garantili, iphone, siyah, türkiye] şeklinde olacaktır.

İlk cümleden elde edilen vektör temsili: [1, 0, 1, 1, 1, 1]

İkinci cümleden elde edilen vektör temsili: [0, 1, 1, 1, 0, 0]

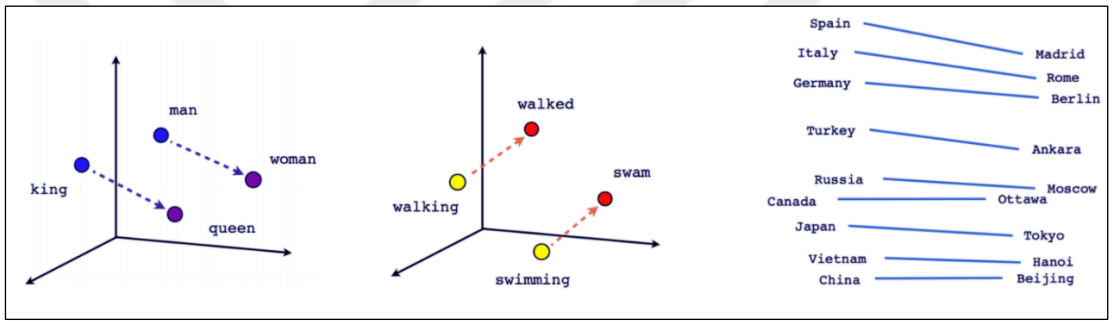
2.4.2. Word2Vec yöntemi

Word2Vec yöntemi, derin öğrenme tabanlı en sık kullanılan yöntemlerden birisidir. 2013 yılında Mikalov ve diğerleri [132] tarafından geliştirilen bu yöntem, temelinde kelimelerin birbirleri ile olan ilişkisini çıkartabilmektedir. Birer adet giriş, gizli ve çıkış katmanından oluşan Word2Vec modeli giriş verisi olarak büyük bir metin kümesi alır. Derin öğrenme modeli, kelimeleri çok boyutlu bir vektör uzayına yerleştir. İlişkili olan kelimeler vektör uzayında birbirlerine yakın olacak şekilde konumlandırılır. Böylelikle kelimelerin anlamsal olarak karşılaştırılmasının yapılabileceği sayısal vektörler oluşturulur.



Şekil 2.15. CBOW ve Skip-gram modelleri [132]

Mikalov ve diğerleri [132] çalışmalarında, CBOW ve skip-gram olmak üzere iki farklı word2vec yöntemi geliştirmiştirler (Şekil 2.15). Bu iki yöntem temelde giriş ve çıkış katmanlarının yer değiştirmesinden meydana gelir. CBOW yönteminde, modele giriş olarak birden fazla kelime verilir ve bu kelimelere anlamsal olarak en yakın kelimeyi modelin çıktısı olarak verir. Skip-gram yönteminde ise tek kelime modele giriş olarak verilir ve birden fazla anlamsal yakınlığı bulunan kelime çıktı olarak beklenir. Örneğin "Ankara Türkiye'nin başkentidir." cümlesini, ekleri ve noktalama işaretleri temizlenmiş olarak ele alalım. CBOW yönteminde "ankara" ve "türkiye" kelimeleri modele giriş olarak verilir, çıktı olarak "başkent" kelimesi beklenir. Skip-gram yönteminde ise sadece "ankara" kelimesi modele giriş olarak verilir ve model çıktısı olarak "türkiye" ve "başkent" kelimeleri beklenir.



Şekil 2.16. Word2Vec uzay temsili [133]

Cinsiyet, spor aktivitesi ve ülke-başkent ilişkisini gösteren örnek word2vec uzay vektör temsilleri Şekil 2.16'da verilmiştir.

2.4.3. GloVe temsili

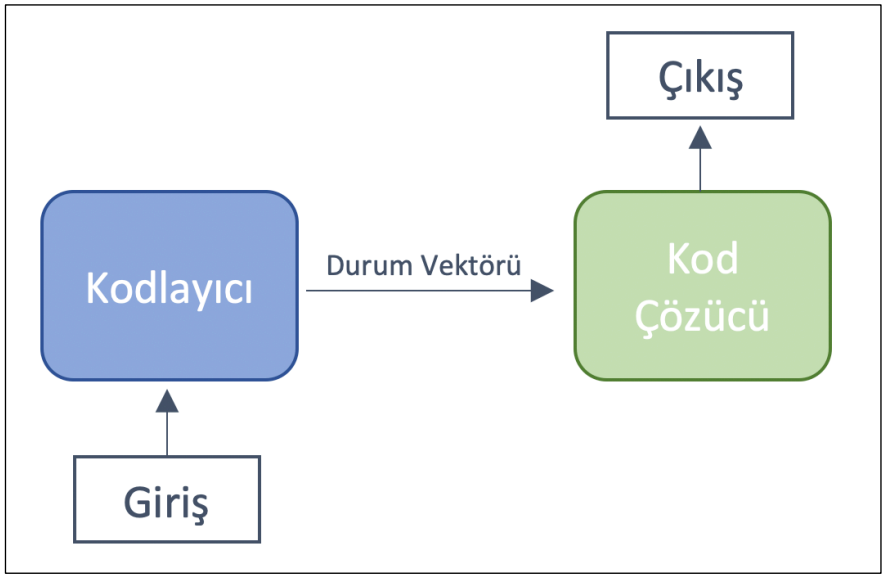
GloVe, 2014 yılında Pennington ve diğerleri [134] tarafından yapılan çalışma ile tanıtılmıştır. GloVe modeli word2vec modeline benzer şekilde kelimelerin anlamsal ilişkilerine göre vektör gösterimi sağlar. Word2Vec modeli, vektör temsilini oluşturken anlamsal bilgiyi yakalamasına rağmen kelimelerin birlikte geçme istatistiklerini kullanmaz. Pennington ve diğerleri [134], kelimelerin birlikte geçme bilgisinin önemli olduğunu vurgulayarak, kelimelerin birlikte geçme olasılığına da hesaba katan yeni bir hata fonksiyonu geliştirmişler. GloVe modeli kelimelerin analoji tespitinde %75 doğruluk oranı sağlayarak diğer yöntemlerden daha iyi başarım elde ettiğini göstermiştir.

2.4.4. FastText temsili

Word2Vec yöntemi doğal dil işleme alanında sıkça kullanılmasına rağmen veri kümesinde olmayan kelimelerin vektörlerini çıkartamaz. Bu sorunun üstesinden gelmek için Bojanowski ve diğerleri [135] tarafından 2016 yılında yapılan çalışma ile FastText kelime temsil yöntemi tanıtılmıştır. Bu çalışmada word2vec yönteminden farklı olarak kelimeleri bir bütün halinde kullanmak yerine kelimelerin derinine inip karakter bazında çalışır ve kelime vektörlerinin çıkartılmasını sağlar. FastText yönteminin iki önemli avantajı sağlar. Bunlardan ilki eğitim verisinde bulunmayan kelimeler için de temsil vektörü çıkartabilmesidir. İkincisi ise daha az veri kümesine ihtiyaç duymasıdır.

2.5. Kodlayıcı – Kod Çözücü Mimarisi

Yinelemeli sinir ağları, makine çevirisi, metin özetleme, ses tanımlama gibi sıralı veri problemlerini çözmek için başarılı olarak kullanılmaktadır. Yinemeli sinir ağlarının bir türü olan seq2seq (sequence to sequence) modeli, sıralı verilerden yine sıralı bir şekilde veri çıkartılmasını sağlayan bir modeldir. Seq2Seq modelleri oluşturmak için kullanılan en yaygın yöntemlerden birisi kodlayıcı–kod çözücü (encoder-decoder) mimarisidir. Bu mimari temel olarak iki yapıdan oluşur (Şekil 2.17). Bu yapılar, giriş verilerilerinden bilgi çıkartımı sağlayan kodlayıcı ve çıkartılan bu bilgiler ile anlamlı bir çıktı üreten kod çözücü yapısıdır.



Şekil 2.17. Kodlayıcı - kod çözücü genel yapısı

Kodlayıcı kısmında sıralı girişin dizi sayısı T_x olmak üzere, $x = (x_1, \dots, x_{T_x})$ giriş verisi, LSTM ve GRU gibi yinelemeli sinir ağları ile okunarak modelin gizli katman bilgileri durum vektörüne aktarılır. Kodlayıcıda gerçekleştirilen durum vektörü oluşturma işlemi Denklem (2.5) ve (2.6)'daki gibi;

$$h_t = f(x_t, h_{t-1}) \tag{2.5}$$

$$c = q(\{h_1, \dots, h_{T_x}\}) \tag{2.6}$$

gösterilmiştir. f ve q doğrusal olmayan fonksiyonlar olmak üzere; h_t, t anındaki gizli katman bilgisini, c ise durum vektörünü temsil etmektedir [136].

Kod çözücü kısmında, c durum vektörü ve önceki çıkışlar $\{y_1, \dots, y_{t-1}\}$ kullanılarak bir sonraki çıkış bilgisi y_t'nin tahmini için eğitilir. Denklem (2.7)'de gösterildiği gibi;

$$p(\boldsymbol{y}) = \prod_{t=1}^{T} p(y_t | \{y_1, \dots, y_{t-1}\}, c) \tag{2.7}$$

$\boldsymbol{y} = (y_1, \dots, y_{T_y})$ olmak üzere kod çözücünün birleşik olasılık hesaplaması verilmiştir.

2.6. Dikkat Mekanizması

Kodlayıcı–kod çözücü mimarisinde durum vektörü her iterasyonda aynı ve sabit kalmaktadır. Bu durum, model eğitimi esnasında, uzun sıralı giriş verileri içerisindeki tüm bilgilerin çıkartılamaması gibi sorunlara neden olmaktadır. Bahdanau ve arkadaşları [136], bu soruna çözüm olarak 2015 yılında dikkat mekanizmasını yayınladıkları çalışma ile tanıtmıştırlar. Dikkat mekanizması her iterasyonda durum vektörünün farklı bölgelerine odaklanmasını sağlar. Bu sayede her bir giriş verisinin hangi çıkış verisiyle daha fazla ilgili olduğu tanımlanır ve eğitim bu odaklanmaya göre gerçekleşir.

RNN modelinin gizli katman bilgileri s_i olmak üzere, dikkat mekanizması ile oluşturulan kod çözücünün birleşik olasılıksal formülü Denklem (2.8)'deki gibi;

$$p(y_i | y_1, \dots, y_{i-1}, \boldsymbol{x}) = g(y_{i-1}, s_i, c_i) \tag{2.8}$$

verilmiştir. Denklem (2.9)'da gibi;

$$c_i = \sum_{j=1}^{T_x} a_{ij} h_j \tag{2.9}$$

her adımda farklı oluşturulan durum vektörünün formülasyonu verilmiştir. Burada a_j gizli katman ağırlığını ifade etmektedir. Bu ağırlığı bulmak için, Denklem (2.10) ve (2.11)'de;

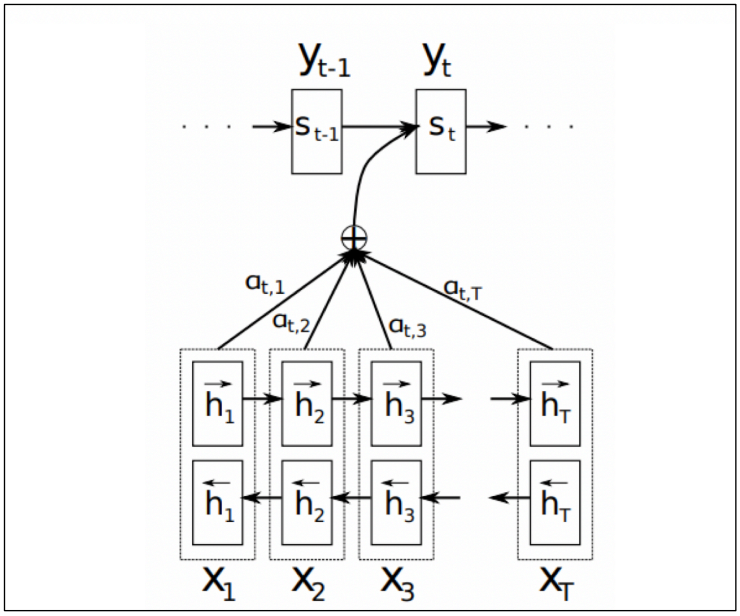
$$s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{2.10}$$

$$e_{ij} = a(s_{i-1}, h_j) \tag{2.11}$$

Hizalama modeli gösterilmiştir. Hizalama modeli j konumundaki girdi ile i konumundaki çıkış eşleşmesinin ne kadar iyi olduğunu puanlayan bir modeldir. Hizalama modelinden çıkan e_{ij} sonucu, Denklem (2.12)'ye;

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{2.12}$$

verilerek a_{ij} ağırlığı hesaplanır. Şekil 2.18'de dikkat mekanizmasının yapısı gösterilmiştir.



Şekil 2.18. Dikkat mekanizmasının yapısı [136]

2.7. Değerlendirme Ölçütleri

Görüntü başlıklama çalışmalarında modelin ürettiği başlık ile gerçek başlık karşılaştırılarak değerlendirme yapılmaktadır. Bu yöntemler genel olarak makine çevirisi gibi doğal dil işleme çalışmalarında kullanılan metriklerden oluşmaktadır.

2.7.1. BLUE

Makine çevirisinde yaygın olarak kullanılan BLUE (Bilingual Evaluation Understudy) skorlaması görüntü başlıklama çalışmaları için de en sık kullanılan yöntemlerden birisidir. Bu yöntem, gerçek başlıktaki kelimelerin, üretilen başlıkta ne kadarının doğru tahmin edildiğine dayanmaktadır. Test veri kümesi üzerinde hesaplanan skorun ortalaması alınarak modelin başarımı hesaplanır. Genel olarak 0,0 – 1,0 arasında ya da 0 – 100 arasında değerlendirme sağlanır. BLUE skorlama yönteminde n-gram tabanlı, B@1 (unigram), B@2 (bigram), B@3 (trigram) ve B@4 (four-gram) gibi farklı varyantlar bulunmaktadır. Bu varyantlar, tahminlenen başlık ile gerçek başlık korelasyonunu n-gram olarak değerlendirilmesini sağlar.

Blue skorlama formülü Denklem (2.13)'te;

$$B@n = \frac{\sum_{S \in \{C\}} count_{matched}(n-gram)}{\sum_{S \in \{C\}} count_{total}(n-gram)} \tag{2.13}$$

gösterildiği gibi *n*, BLUE skor varyasyonunu ifade etmektedir. Test veri kümesi *C*, her bir cümle ise *S* ile gösterilmiştir. Formülasyonun payda kısmında test kümesindeki metnin n-gram bazında sayısı verilmiştir. Pay kısmında ise modelin ürettiği metin ile test kümesinde yer alan metnin eşleşen n-gram sayısı yer almaktadır.

2.7.2. METEOR

METEOR (Metric for Evaluation of Translation with Explicit ORdering) skorlama yönteminde, BLUE skorlamada olduğu gibi tahminlenen başlık ile gerçek başlığı karşılaştırarak skorlama elde edilir. BLUE skorlamadan farklı olarak bir ceza puanı hesaba katılarak skorlama yapılır.

METEOR skorlama yöntemi sırasıyla Denklem (2.13), (2.14) ve (2.15)'teki gibi;

$$F_{mean} = \frac{10\,PR}{R + 9P} \tag{2.13}$$

$$Penalty = 0{,}5 * (\frac{c}{U_m}) \tag{2.14}$$

$$Score = F_{mean} * (1 - Penalty) \tag{2.15}$$

hesaplanmaktadır. P kesinlik (presicion) değeri, modelin ürettiği metindeki unigram sayısının, gerçek metindeki unigram sayısı ile eşleşme oranını, R duyarlılık (recall) ise gerçek metindeki unigram sayısının ne kadarın doğru tespit edildiğini ifade etmektedir. P ve R değerlerinin harmonik ortalaması ile F_{mean} hesaplanır. Modelin ürettiği başlığın, gerçek başlıktaki n-gram bazlı eşleşen parçaların hesaplanıp, n-gram değeri en yüksek olan parça sayısı c'ye eşitlenmiştir. U_m ise unigram olarak tahmin edilen başlık ve gerçek başlıktaki eşleşen sayıyı ifade eder. c ve U_m değerleri ile bir ceza değeri hesaplanır. F_{mean} ve ceza değeri çarpılarak METEOR skoru hesaplanır.

2.7.1. CIDEr

CIDEr (Consensus-based Image Description Evaluation) değerlendirme ölçütü diğer yöntemlerin aksine görüntü başlıklama çalışmaları için oluşturulmuş bir yöntemdir. TF-IDF (Term Frequency - Inverse Document Frequency) tabanlı gerçekleşleştirilen bu yöntem, kelimelerin cümle içerisindeki önemini hesaba katarak skorlama gerçekleştirilir. Denklem (2.16)'daki gibi;

$$CIDEr_n(C, S) = \frac{g^n(C).g^n(S)}{\cos_\text{sim}(g^n(C),\ g^n(S))} \tag{2.16}$$

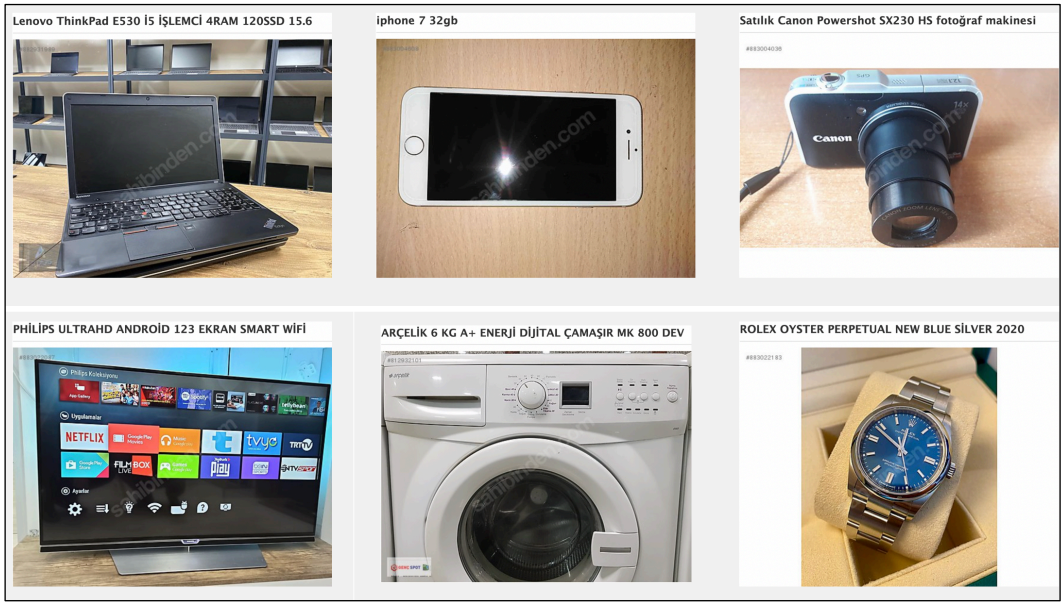
CIDEr skorlamasının formülasyonu verilmiştir. C, tahminlenen başlık, S ise gerçek başlık olmak üzere *n* değeri n-gram varyasyonunu ifade etmektedir. g^n fonksiyonu, n-gram bazında TF-IDF değerini verir. Tahminlenen başlık ile gerçek başlığın TF-IDF değerlerinin çarpımının, yine bu değerlerin kosinüs uzaklığına oranı CIDEr skorunu elde etmemizi sağlar.

3. DENEYSEL ÇALIŞMA

Tez çalışmasının bu bölümünde ilk olarak veri kümesine ve önişleme adımlarına daha sonra yapay zekâ çalışmalarında yaygın olarak kullanılan olan Tensorflow derin öğrenme kütüphanesi ile modelin nasıl oluşturulup eğitildiğine son olarak ise değerlendirme metrikleri ile elde edilen skorlara ve sırasıyla başarılı, kısmen başarılı ve başarısız olarak üretilen sonuçlara yer verilecektir.

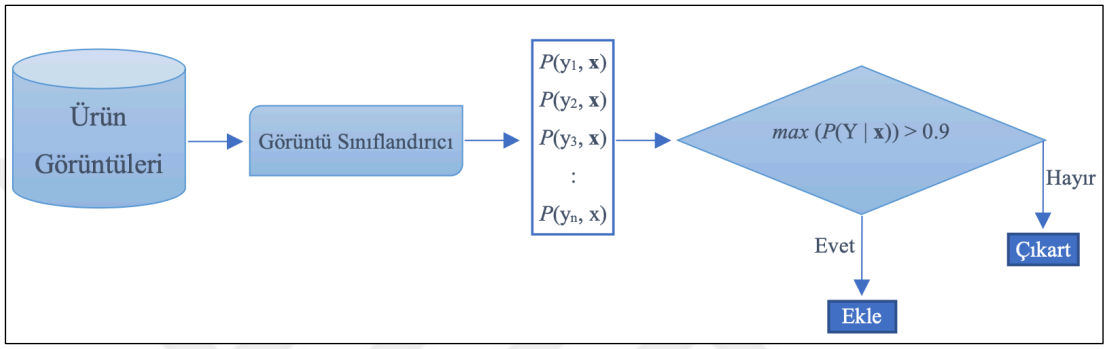
3.1. Veri Kümesi

Literatürde, görüntü başlıklama çalışmalarında çoğunlukla MS COCO, Flick gibi veri kümeleri kullanıldığı görülmektedir. Bu veri kümeleri görüntü ve bu görüntüyü tanımlayan 5 farklı başlıktan oluşan ikili verilerden oluşmaktadır. Bu tez çalışmasının konusu olan ürün görüntüleri üzerinden otomatik başlık üretme için gerekli olan ürün görüntüleri, başlıkları ve kategori bilgileri sahibinden.com sitesi üzerinden elde edilmiştir. Bir ilan üzerindeki ortalama 5 farklı görüntüye karşılık, bir başlık olmak üzere yaklaşık 5,5 milyon görüntü – başlık ikili veri kümesi sahibinden.com sitesinde, İkinci El ve Sıfır Alışveriş kategorisi altından elde edilmiştir. Örnek görüntü – başlık ikili verileri Şekil 3.1'de verilmiştir.



Şekil 3.1. Veri kümesinden örnekler

Toplanan veri kümesi üzerinde oldukça fazla kirli veriler bulunmaktaydı. Görüntü verilerinde tekrarlı kayıtlar ve kalitesi düşük olan görüntüler, başlıklarda ise kısa, hatalı ve görüntüyü tanımlamakta yetersiz olan metinler bulunmaktaydı. Makine öğrenme çalışmalarında kirli veriler modelin başarısını oldukça düşürmektedir. Bu nedenle kirli verileri elemek için görüntüler ve metinler üzerinde uygulanmak üzere iki farklı sınıflandırma yöntemi oluşturulmuştur. İlk olarak görüntü-kategori ikili verileri ile görüntü sınıflandırma modeli kurulmuştur.

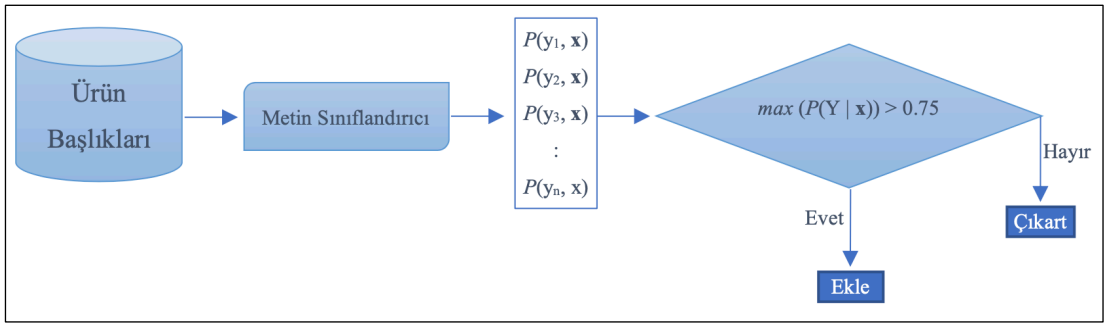


Şekil 3.2. Görüntü filtrelemek için kullanılan sınıflandırıcı mimarisi (x: görüntü, y_1, y_2, y_3, .., y_n : kategori, P(Y | x): x giriş verisinde Y gelme olasılığı)

Görüntüler için oluşturulan sınıflandırıcı mimarisi Şekil 3.2'de gösterilmiştir. Görüntü sınıflandırıcı, evrişimli sinir ağı ile oluşturulmuştur. Hazır eğitilmiş model olan InceptionV3 modeli kullanılarak transfer öğrenme yöntemi ile görüntünün özellik vektörü çıkartılmıştır. Çıkartılan özellik vektörü 4 katmanlı derin öğrenme modeline verilerek görüntülerin hangi kategoriye ait olma olasılık dağılımı elde edilmiştir. Veri kümemizdeki kirli verileri elemek için son katmanda herhangi bir kategoriye ait olma olasılığı 0,9'dan küçük olan veriler kirli veri olarak işaretlenerek veri kümesinden çıkartılmıştır.

Görüntülerin filtrelenmesine benzer şekilde ürün başlıkları ve kategorileri ile metin sınıflandırma modeli oluşturulmuştur. İlk olarak kelime kalıplama yöntemi ile başlıkların özellik vektörleri çıkartılmıştır. Elde edilen özellik vektörleri yinelemeli sinir ağı modeli ile eğitilmiştir. Oluşturulan yinelemeli sinir ağı modelinin son katmanı, giriş metninin kategori bazlı olasılık dağılımını verir. Olasılık dağılımında herhangi bir kategoriye ait olma olasılığı 0,75 değerinden küçük olan başlıklar işaretlenerek veri kümesinden çıkartılmıştır. Metin verilerini filtrelemek için oluşturulan model Şekil

3.3'te gösterilmiştir. Görüntü ve metin sınıflandırma modeli ile kirli verilerin elenmesi işlemi sağlanmıştır.



Şekil 3.3. Metin filtrelemek için oluşturulan sınıflandırıcı mimarisi(x: başlık, y_1, y_2, y_3, .., y_n : kategori, P(Y | x): : x giriş verisinde Y gelme olasılığı)

Veri filtreleme aşamalarından sonra elde edilen veri kümesi detayları Tablo 3.1'de verilmiştir.

Tablo 3.1. Veri kümesi detayları

Başlıktaki Ortalama Kelime Sayısı	**Filtrelenmemiş Eşsiz Kelime Sayısı**	**Filtrelenmiş Eşsiz Kelime Sayısı**	**Filtrelenmemiş görüntü-başlık sayısı**	**Filtrelenmiş görüntü-başlık sayısı**
9.4	232025	5000	5591956	1816146

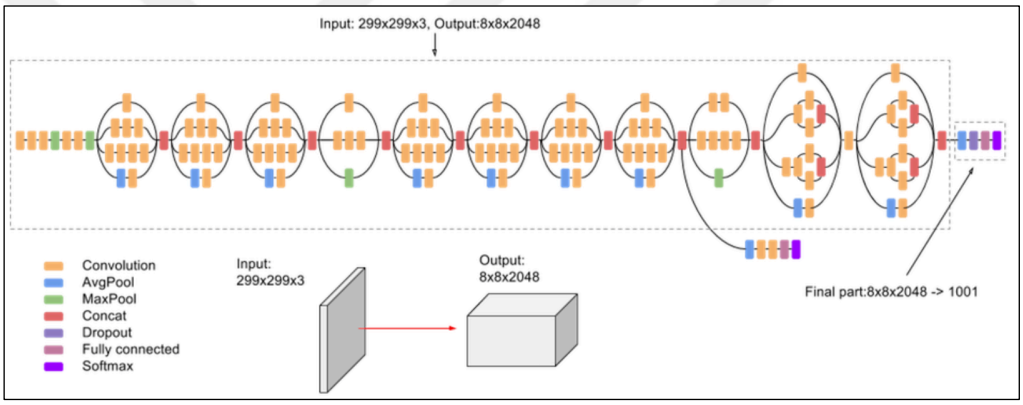
3.2. Derin Öğrenme Model Tasarımı

Derin öğrenme çalışmalarında model tasarımı için genellikle Tensorflow, PyTorch, Caffe gibi farklı derin öğrenme kütüphaneleri kullanılmaktadır. Bu kütüphaneler temelde derin öğrenme katmanlarını birbirine bağlayarak basit bir şekilde modelin eğitilmesini ve test edilmesini sağlar. Tez kapsamında yapılan çalışmada Tensorflow derin öğrenme kütüphanesi kullanılmıştır. Tensorflow ilk sürümü 2015 yılında Google tarafından açık kaynaklı olarak yayınlanmıştır. Python, C++ ve R gibi farklı programlama dillerini destekleyen Tensorflow kütüphanesi, doğal dil işleme, metin sınıflandırma, görüntü tanımlama gibi birçok farklı çalışma alanlarında yaygın olarak kullanılmaktadır. Google tarafından desteklenmesi, geniş dökümantasyonu ve kullanım kolaylığı nedeniyle derin öğrenme modelini oluşturmak için Tensorflow kütüphanesi tercih edilmiştir.

Derin öğrenme modelini oluşturmak için görüntü başlıklama çalışmalarında son zamanlarda oldukça başarılı bir mimari olan dikkat mekanizmalı kodlayıcı – kod çözücü mimarisi kullanılmıştır. Model tasarımı için görüntü başlıklama çalışmaları için hazırlanan Tensorflow dokümantasyonundan [137] yaralanılmıştır.

3.2.1. Ürün görüntülerinden özellik vektörü çıkartımı

Ürün görüntülerinden özellik vektörünü çıkartmak için transfer öğrenme tekniği uygulanmıştır. Transfer öğrenme, önceden eğitilmiş bir modelin katman bilgilerinin yeni modele aktarılması ile yapılmaktadır. Bunun için Tensorflow kütüphanesinde ImageNet veri kümesi ile eğitilen InceptionV3 modeli kullanılmıştır. InceptionV3 mimarisi Şekil 3.4'te verilmiştir.



Şekil 3.4. InceptionV3 mimarisi [138]

InceptionV3 modeli sınıflandırma çalışmalarında kullanılmak üzere evrişimli sinir ağı ile eğitilmiştir. Bu modelin son katmanı görüntü özellik vektöründen sınıflandırma yapmaktadır. Bu çalışmada sadece görüntülerin özellik vektörü gerekli olduğundan InceptionV3 modelinin son katmanı atılarak görüntülerin özellik vektörünün çıkartılması için hazır hale getirilmiştir.

```
1 from tensorflow.keras.applications import InceptionV3
2 from tensorflow.keras import Model
3
4 inception_v3 = InceptionV3(include_top=False, weights='imagenet')
5
6 new_input = inception_v3.input
7 hidden_layer = inception_v3.layers[-1].output
8
9 image_features_extract_model = Model(new_input, hidden_layer)
```

Şekil 3.5. Tensorflow üzerinde InceptionV3 modeli ile transfer öğrenme modelinin kurulması

Tensorflow üzerinde InceptionV3 modelinin transfer öğrenme tekniği ile gerçekleştirilmesi Şekil 3.5'teki kod parçacığında gösterilmiştir. Bu kod parçacığında ilk olarak InceptionV3 modelinin giriş katmanı alınır. Daha sonra gizli katmanlarından sonuncusu hariç diğerleri alınarak yeni bir model oluşturulur. Alınan bu katmanlarla birlikte katmanların önceden eğitilmiş ağırlıkları da alınarak tekrar eğitime gerek kalmadan görüntülerin özellik vektörlerini elde edebileceğimiz modeli bize sağlamış olur.

3.2.2. Ürün başlıkların ön işlemesi ve vektör çıkartımı

Ürün görüntülerinin özellik vektörleri ile birlikle başlıkların da modelin tanımlayabileceği bir hale çevirilmesi gerekmektedir. Bunun için başlıkları ilk olarak kelimelerine ayırıp bir sözlük oluşturulur. Sonrasında sözlük içerisinde her kelimeye farklı bir indeks atanır. Başlıktaki her bir kelime sözlükteki indeksi ile değiştirilerek sayısal vektöre çevrilir. Şekil 3.6'daki kod parçacığında başlıklar üzerindeki ön işleme adımı ve vektöre çevirme işlemini gösterilmiştir.

```
1 from tensorflow.keras.preprocessing.text import Tokenizer
2 from tensorflow.keras.preprocessing.sequence import pad_sequences
3
4 top_k = 5000
5 tokenizer = Tokenizer(num_words=top_k, oov_token="<UNK>", filters='!"#$%&()*+.,-/:;=?@[\]^_`{|}~ ')
6
7 tokenizer.word_index['<PAD>'] = 0
8 tokenizer.index_word[0] = '<PAD>'
9
10 tokenizer.fit_on_texts(train_titles)
11
12 train_sequences = tokenizer.texts_to_sequences(train_titles)
13
14 cap_vector = pad_sequences(train_sequences, padding='post')
```

Şekil 3.6. Başlıkların ön işlenmesi ve vektöre çevrilmesi

Şekil 3.6'da verilen kod parçacığını açıklayacak olursak, ilk olarak "Tokenizer" sınıfını ve "pad_sequences" fonksiyonunu kullanmak üzere yüklüyoruz. Tokenizer metin verileri üzerinde çalışan ve metinleri sayısal vektöre çevirmeyi sağlayan yardımcı bir sınıftır. 5. satırda başlıklar içerisinde en fazla geçme sırasına göre ilk 5000 kelime ile sözlük oluşturan Tokenizer sınıfından bir değişken oluşturulmuştur. "filters" parametresi metinlerin içerisinde hangi karakterlerin silineceğini, "oov_token" parametresi ise sözlük içerisinde bulunmayan kelimelere hangi değerin atanacağını belirtmektedir. 10. satırda veri kümesindeki başlıklar ile Tokenizer'ı eğitmek için

“fit_on_texts” fonksiyonu çağrılmıştır. 12. satırda ise eğitilen Tokenizer ile başlıkların sayısal vektöre çevirim işlemi gerçekleştirilmiştir.

Veri kümesindeki başlıklar farklı sayıda kelimelerden oluştuğu için bu verilerin sabit vektör boyutuna çevirilmesi gerekmektedir. 14. satırda sayısal hale getirilen vektörlerin “pad_sequences” fonksiyonu ile sabit boyuta çevirilme işlemi yapılmaktadır. Burada “padding” parametresi vektörün önüne mi yoksa sonuna mı boş verilerin ekleneceğini belirtmektedir. Bu işlemler sonucunda veri kümesindeki başlıklar sabit uzunlukta sayısal vektörler olarak temsil edilmiştir.

3.2.3. Tensorflow modellerinin oluşturulması

Tez çalışması kapsamında Tensorflow modeli oluşturulurken alt sınıflandırma yöntemi kullanılmıştır. Alt sınıflandırma, Tensorflow kütüphanesi içerisinde “fit”, “evaluate” ve “predict” gibi bazı fonksiyonların ön tanımlı gelen “tensorflow.keras.Model” sınıfından kalıtım alınarak yeni model oluşturma yöntemidir.

Dikkat mekanizmalı kodlayıcı – kod çözücü mimarisi için 3 farklı Tensorflow modeli oluşturulmuştur. Bu modellerden ilki olan dikkat mekanizması modeli Şekil 3.7’deki kod parçacığında gösterilmiştir.

```
1  from tensorflow.keras import Model
2  from tensorflow.keras.layers import Dense
3  from tensorflow import expand_dims, reduce_sum
4  from tensorflow.nn import tanh, softmax
5
6  class Attention(Model):
7    def __init__(self, units):
8      super(Attention, self).__init__()
9      self.W1 = Dense(units)
10     self.W2 = Dense(units)
11     self.V = Dense(1)
12
13   def call(self, features, hidden):
14     hidden_with_time_axis = expand_dims(hidden, 1)
15
16     attention_hidden_layer = (tanh(self.W1(features) + self.W2(hidden_with_time_axis)))
17
18     score = self.V(attention_hidden_layer)
19
20     attention_weights = softmax(score, axis=1)
21
22     context_vector = attention_weights * features
23     context_vector = reduce_sum(context_vector, axis=1)
24
25     return context_vector, attention_weights
```

Şekil 3.7. Dikkat mekanizması modeli

Oluşturulan ikinci model ise kodlayıcı modelidir. Tanımlanan mimarinin kodlayıcı kısmında görüntülerin özellik çıkartımı yapılmaktadır. Ancak görüntülerin özellik vektörünü çıkartmak oldukça uzun zaman alacağı için bu aşama transfer öğrenme tekniği ile yapılmıştır. Dolayısıyla oluşturulan kodlayıcı modeli sadece transfer öğrenme ile çıkartılan görüntü özelliklerini tam bağlanım katmanına aktarmayı sağlar. Şekil 3.8'de Tensorflow'da oluşturulan kodlayıcı modelinin kod parçacığı, Şekil 3.9'da ise kodlayıcı modelinin "model.summary" fonksiyonundan elde edilmiş özet bilgilerine yer verilmiştir.

```
1 from tensorflow.keras import Model
2 from tensorflow.keras.layers import Dense
3 from tensorflow.nn import relu
4
5 class Encoder(Model):
6
7     def __init__(self, embedding_dim):
8         super(Encoder, self).__init__()
9
10        self.fc = Dense(embedding_dim)
11
12    def call(self, x):
13        x = self.fc(x)
14        x = relu(x)
15        return x
```

Şekil 3.8. Kodlayıcı modelinin Tensorflow üzerinde oluşturulması

```
Model: "encoder"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                multiple                  524544
=================================================================
Total params: 524,544
Trainable params: 524,544
Non-trainable params: 0
_________________________________________________________________
```

Şekil 3.9. Kodlayıcı modelinin özet bilgisi

Tensorflow üzerinde oluşturulan son model ise kod çözücü modelidir. Şekil 3.10'da verilen kod çözücü modelinde, tekrarlayan sinir ağı olarak GRU mimarisi kullanılmıştır. Bunun dışında bir kelime kalıpma katmanı, iki tam bağlanım katmanı

ve önceden oluşturulan dikkat mekanizması kod çözücü modelinin katmanları olarak tanımlanmıştır. Şekil 3.11'de kod çözücü modelinin özet bilgileri gösterilmiştir.

```
from tensorflow.keras import Model
from tensorflow.keras.layers import Embedding
from tensorflow.keras.layers import GRU
from tensorflow.keras.layers import Dense
from tensorflow import expand_dims, reshape, zeros

class Decoder(Model):
  def __init__(self, embedding_dim, units, vocab_size):
    super(Decoder, self).__init__()
    self.units = units

    self.embedding = Embedding(vocab_size, embedding_dim)

    self.gru = GRU(self.units)

    self.fc1 = Dense(self.units)
    self.fc2 = Dense(vocab_size)

    self.attention = Attention(self.units)

  def call(self, x, features, hidden):
    context_vector, attention_weights = self.attention(features, hidden)

    x = self.embedding(x)
    x = tf.concat([expand_dims(context_vector, 1), x], axis=-1)

    output, state = self.gru(x)

    x = self.fc1(output)
    x = reshape(x, (-1, x.shape[2]))
    x = self.fc2(x)

    return x, state, attention_weights

  def reset_state(self, batch_size):
    return zeros((batch_size, self.units))
```

Şekil 3.10. Kod çözücü modelinin Tensorflow üzerinde oluşturulması

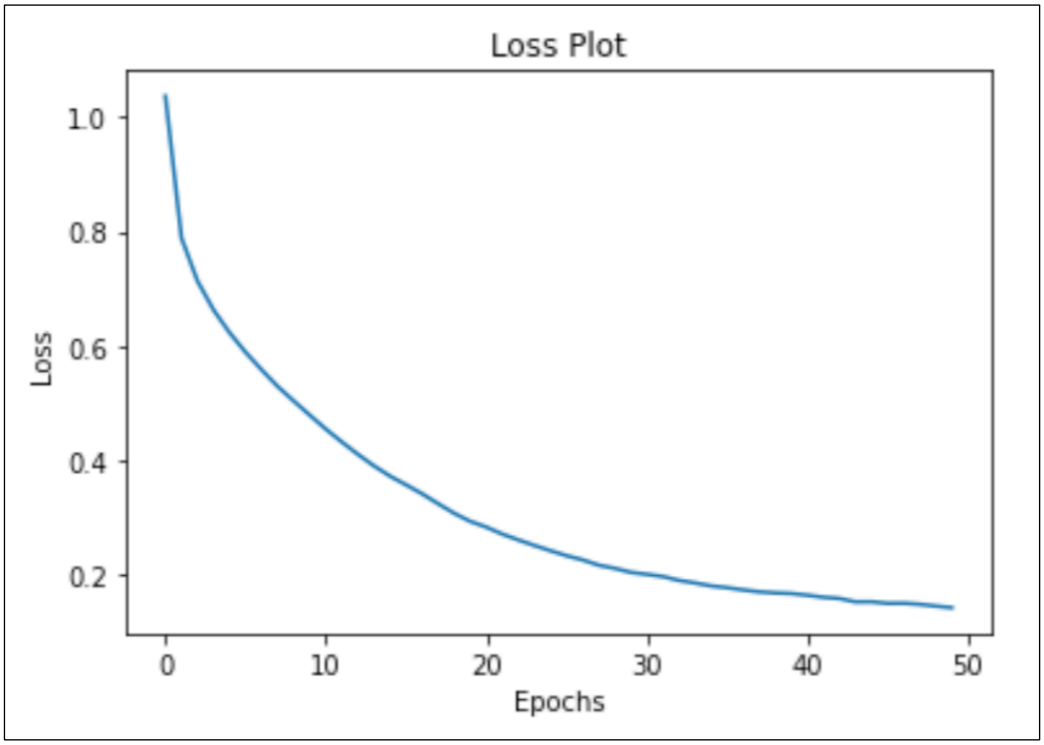
```
Model: "decoder"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        multiple                  1280256
_________________________________________________________________
gru (GRU)                    multiple                  1575936
_________________________________________________________________
dense_1 (Dense)              multiple                  262656
_________________________________________________________________
dense_2 (Dense)              multiple                  2565513
_________________________________________________________________
bahdanau_attention (Bahdanau multiple                  394753
=================================================================
Total params: 6,079,114
Trainable params: 6,079,114
Non-trainable params: 0
_________________________________________________________________
```

Şekil 3.11. Kod çözücü modelinin özet bilgisi

3.3. Model Eğitimi

Tensorflow üzerinde oluşturulan dikkat mekanizmalı kodlayıcı–kod çözücü mimarisinde ürün görüntülerinden elde edilen özellik vektörü kodlayıcı mimarisine giriş bilgisi olarak verilir. Kodlayıcıdan elde edilen çıkış verisi, gizli katman bilgisi ve ürün başlık vektörü, kod çözücü modeline giriş olarak verilir. Kod çözücü, aldığı giriş verileri ile sözlükteki hangi kelimenin sonraki kelime olacağı bilgisinin olasılık değerlerini verir. Kodlayıcıdan sonraki kelime için çıkan olasılık değeri ve gizli katman bilgileri modelin kayıp değerini hesaplamak için kullanılır. Son adım olarak kayıp fonksiyonundan çıkan sonuç neticesinde en iyileme algoritması yardımıyla geri yayılım çalıştırılarak modelin eğitimi gerçekleştirilir.

Oluşturulan model 30 CPU, 120 GB Ram ve 2 Nvidia Tesla P100 grafik kartlı makinede eğitilmiştir. Model eğitimi 50 iterasyon boyunca devam ettirilmiştir ve her bir eğitim ortalama 5-6 saat arasında sürmüştür. İlk iterasyonda 1.9784 olan model kaybı 50 iterasyon sonucunda 0.1438'e düşmüştür. 50 iterasyon sonrasında model aşırı öğrenme eğiliminde olduğu için eğitim 50 iterasyonda bırakılmıştır. Model eğitimi esnasında elde edilen kayıp grafiği Şekil 3.12'de gösterilmiştir.



Şekil 3.12. Model eğitimi sonucunda ortaya çıkan kayıp grafiği

3.4. Deneysel Çalışma Sonuçları

Görüntü başlıklama çalışmalarında ortaya çıkan sonuçları değerlendirmek için BLUE, METEOR ve CIDEr gibi farklı değerlendirme metrikleri kullanılmaktadır. Bu değerlendirme metrikleri, test verisi üzerinde çalıştırılarak modelin ürettiği başlık ile gerçek başlığı karşılaştırarak modelin hata oranını hesaplar.

Tez kapsamında SGD, Adam ve RMSProp en iyileme algoritması kullanılarak model eğitimi gerçekleştirilmiştir. Her bir eğitim esnasında model parametreleri, iterasyon ve veri kümesi sabit tutulmuştur. Tablo 3.2'de farklı en iyileme algoritmaları ile yapılan deneysel çalışmanın sonuçları verilmiştir.

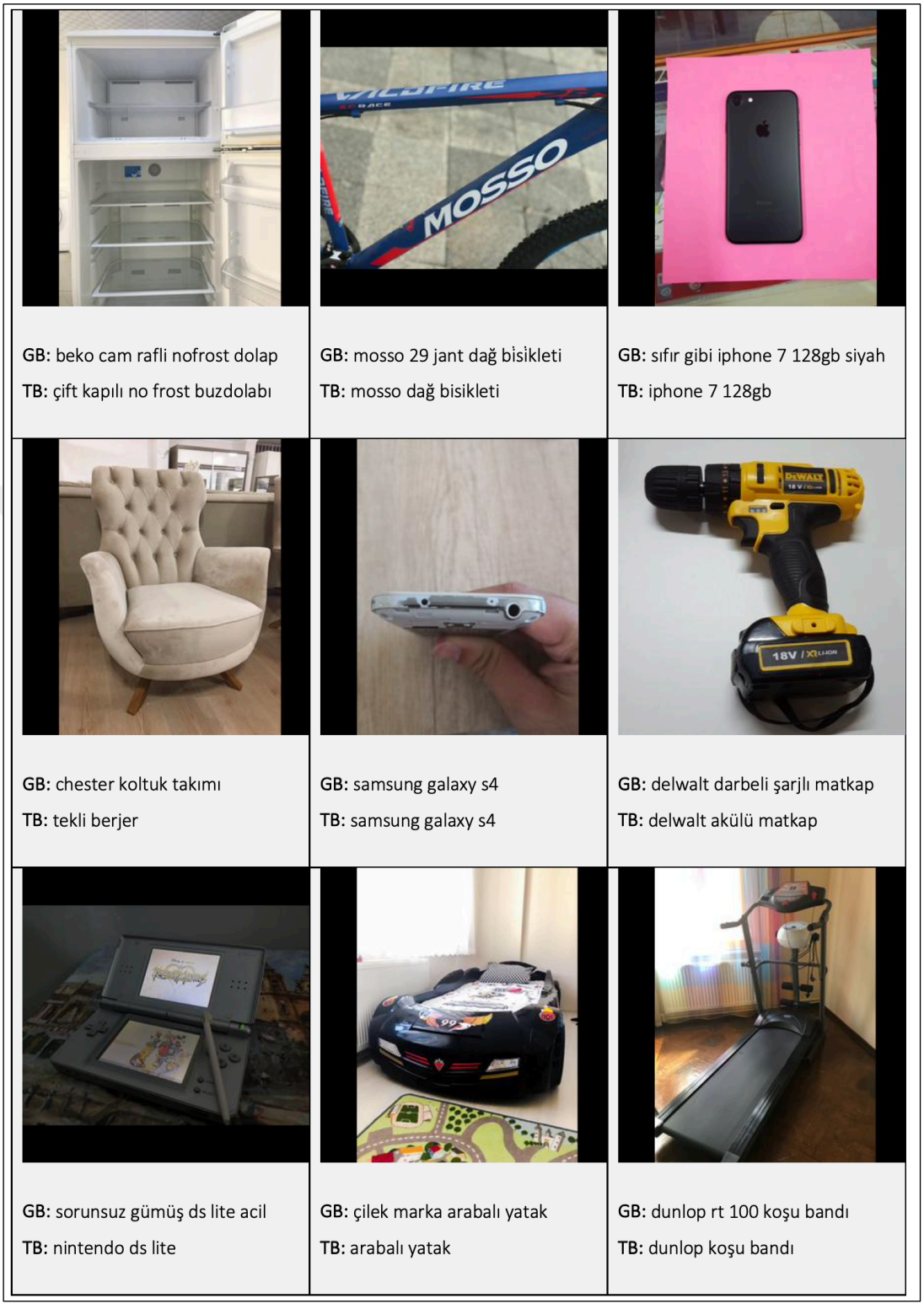
Tablo 3.2. Deneysel çalışma hata oranları

En İyileme Algoritması	**B@1**	**B@2**	**B@3**	**B@4**	**Meteor**	**Cider**
SGD	0,567	0,440	0,330	0,231	0,114	0,452
Adam	**0,687**	**0,623**	**0,564**	**0,494**	**0,130**	**0,634**
RMSProp	0,442	0,374	0,320	0,258	0,125	0,372

Tablo 3.2'de verilen değerlendirme skorlarında, ölçülen başarı oranları arasında en iyi sonucu veren en iyileme algoritmasının Adam olduğu görülmüştür. Adam algoritması, Momentum ve RMSProp en iyileme algoritmalarının iyi özelliklerini birleştirerek, kirli verilerde daha iyi sonuç elde ettiğini söyleyebiliriz.

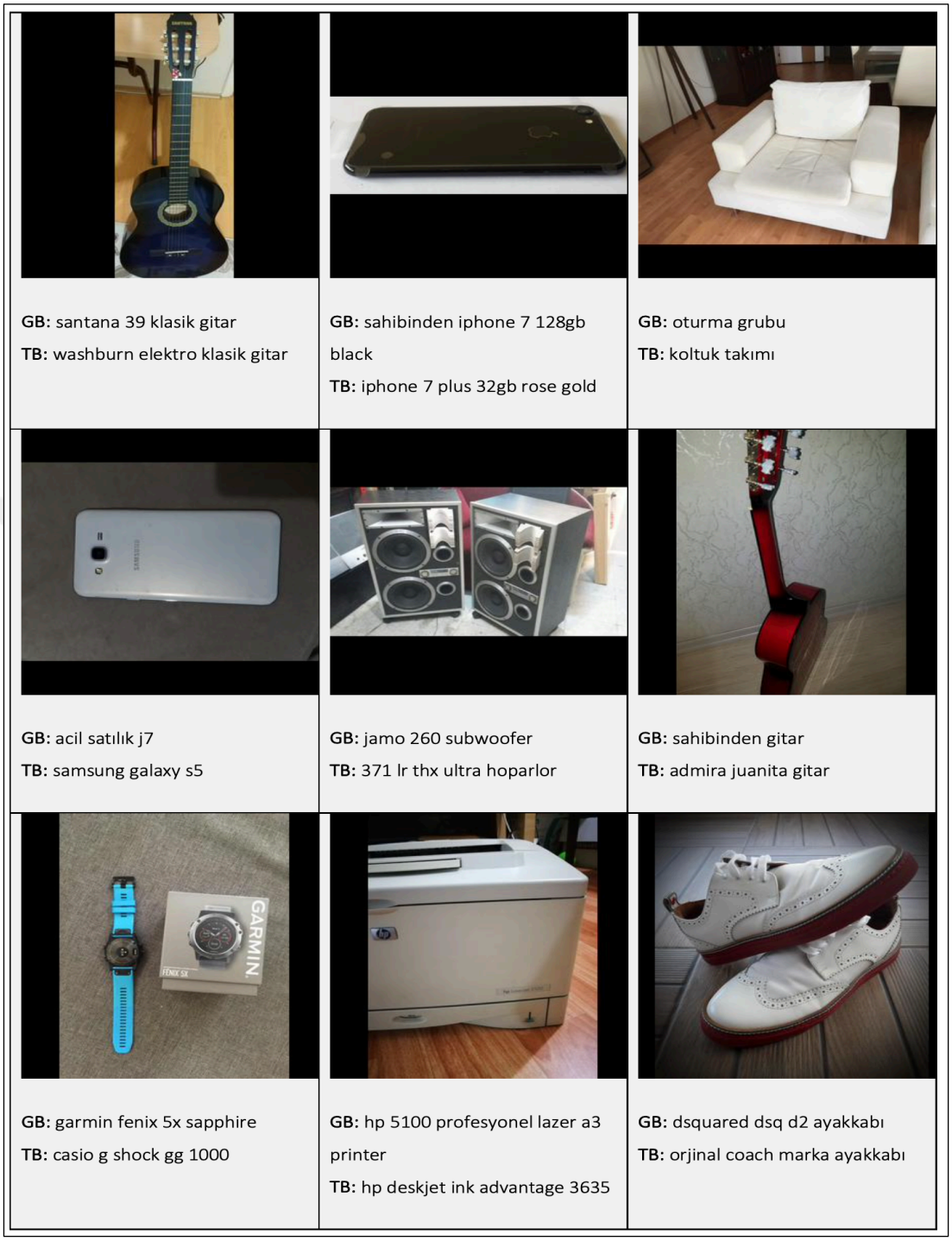
Elde edilen en yüksek BLUE-1 (B@1) skoru 0,687'dir. Bu skor da bize görüntülerin gerçek başlıkları ile model tarafından üretilen başlıklarının ortalama %69 oranında aynı kelimeler ile başlık üretimi gerçekleştirdiğini göstermektedir.

Deneysel çalışma sonucunda elde edilen; başarılı, kısmen başarılı ve başarısız örnekler sırasıyla Şekil 3.13, 3.14 ve 3.15'te verilmiştir.



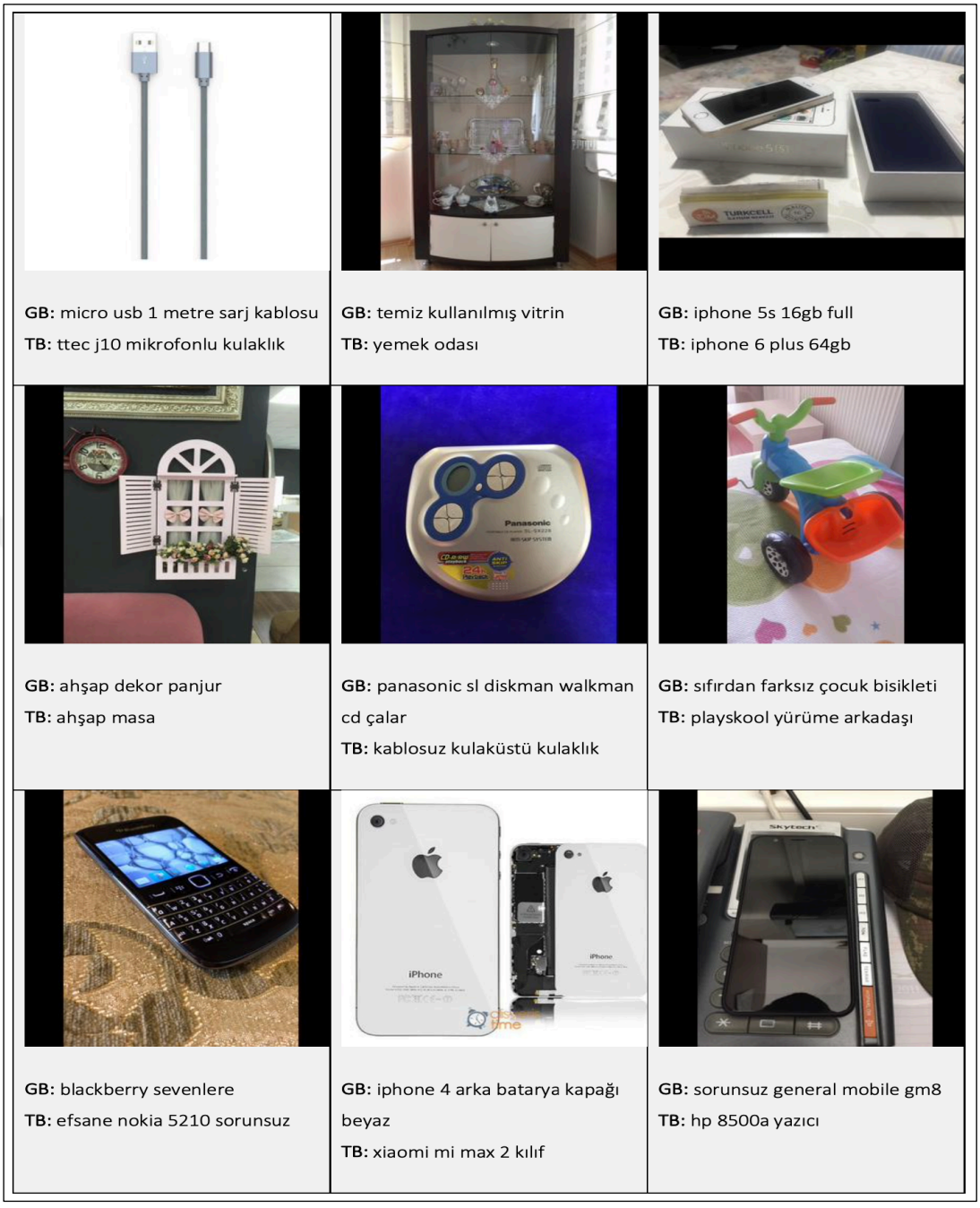
Şekil 3.13. Deneysel çalışmada elde edilen başarılı sonuçlar (GB: gerçek başlık, TB: tahmin edilen başlık)

Şekil 3.13'te verilen sonuçlara baktığımızda görüntü içerisindeki ürünlerin tam olarak ne olduğu tanımlanarak, bu ürünlere uygun başlıkları başarılı bir şekilde model tarafından üretildiği görülmektedir.



Şekil 3.14. Deneysel çalışmada elde edilen kısmen başarılı sonuçlar (GB: gerçek başlık, TB: tahmin edilen başlık)

Şekil 3.14'te verilen kısmen başarılı sonuçlarda, model görüntü üzerindeki ürünün ne olduğunu anlayabilmiş ancak bunlara uygun başlıkları tam olarak üretememiş. Bunun en büyük nedeni ise başlık üzerinde yer alan marka bilgileridir. Örneğin Şekil 3.14'te verilen ilk örnekte, model ürünün gitar olduğunu tanımlayabilmiş ancak Santana markası yerine Washburn markası olarak başlık üretmiştir.



Şekil 3.15. Deneysel çalışmada elde edilen başarısız sonuçlar (GB: gerçek başlık, TB: tahmin edilen başlık

Şekil 3.15'te ise modelin ürettiği başarısız sonuçlar verilmiştir. Burada da yine marka bilgileri ve ürün görüntülerinin birbirlerine benzemesi ya da görüntünün çekildiği açı, modeli yanıltarak yanlış sonuçlar üretmesine neden olmuştur.

Elde edilen sonuçları incelediğimizde, çalışma kapsamında oluşturulan derin öğrenme modeli, başarılı sayılabilecek ürün başlıkları üretmesine karşılık geliştirilecek yönleri de bulunmaktadır.

4. SONUÇLAR VE ÖNERİLER

Son yıllarda derin öğrenme kavramı araştırmacıların ve şirketlerin dikkatini çeken aynı zamanda ilgi odağı haline gelen bir çalışma alanıdır. Derin öğrenmenin temelleri 1960'lı yıllara dayansa da gerçek anlamda tanınması 2010'ların başlarında olmuştur. Bu sürecin uzun sürmesindeki temel nedenler olarak; hesaplama gücündeki yetersizlik, derin öğrenme için geliştirilen algoritmaların yetersiz kalması ve veri kümesi eksikliğini öne sürebiliriz.

Oyun sektöründeki ekran kartları için yapılan yatırımlar sonucunda 2007 yılında Nvidia firması tarafından CUDA teknolojisi geliştirilmiştir. Bu teknoloji grafik kartları üzerinde oldukça hızlı ve paralel hesaplama yeteneği sağlamıştır. Aynı zamanda sunulan API (Application Programming Interface) üzerinden araştırmacıların da grafik kartı üzerinde hesaplama yapmalarına olanak sağlamıştır.

2010'lu yıllarda derin sinir ağı hesaplanması için aktivasyon fonksiyonları, en iyileme algoritmaları, başlangıç ağırlığı tayin etme yöntemleri gibi basit ancak oldukça faydalı çalışmalar yayınlanmıştır. Bu çalışmalar ile birlikte derin sinir ağı eğitimi için bir kısıt olan algoritma sorunları da kısmen çözüme kavuşmuştur.

İnternet kullanım oranının dünya üzerindeki artışı ile birlikte internet kullanıcılarının ürettiği veriler de gün geçtikçe artmıştır ve üretilen bu veriler Wikipedia, Flick, Youtube gibi platformlar üzerinden yayınlanmıştır. Bu sayede derin öğrenme çalışmaları için diğer bir kısıt olan veri kümeleri de bu sayade çözümlenmiştir.

Derin öğrenme alanındaki gelişmeler ile birlikte geleneksel makine öğrenmesi algoritmaları ile çözülemeyen ya da başarısı düşük olan çalışma alanları da araştırmacıları derin öğrenme çalışmalarına doğru yöneltmiştir. Bu çalışma alanlarından biri de görüntü başlıklama olmuştur. Görüntü başlıklama çalışmaları, görüntülerin içeriğini tanımlayıp, dil kurallarına uygun ve anlamlı bir şekilde otomatik başlık oluşturmayı sağlar. Bu çalışmalar hem görüntü işleme hem de dil işleme tekniklerinin birlikte kullanılmasını gerektirdiği için literatürde oldukça zorlu bir çalışma alanı olarak görülmektedir.

Bu tez kapsamında derin öğrenme yöntemleri ile görüntü başlıklama mimarisinin ikinci el ürün görüntülerine uygulanarak başlıkların otomatik çıkartılması amaçlanmıştır. Derin öğrenme mimarisi olarak, Xu ve diğerlerinin çalışmasındaki [13] dikkat mekanizmalı kodlayıcı – kod çözücü mimarisi kullanılmıştır. Sahibinden.com sitesi üzerinden 5,5 milyondan fazla ham veri ön işleme adımları ile filtrelenerek yaklaşık 1,8 milyon görüntü – başlık ikili verisi elde edilmiştir. Derin öğrenme modelinin tasarımı için Google tarafından desteklenen Tensorflow derin öğrenme kütüphanesi kullanılmıştır. Geliştirilen derin öğrenme mimarisi yüksek hesaplama kapasitesi olan makine ile Adam, RMSProp ve SGD en iyileme algoritmaları ile ayrı ayrı 50 iterasyonda eğitilmiştir. BLUE, METEOR ve CIDEr gibi farklı skorlama teknikleri ile değerlendirilen deneysel çalışma sonucunda en iyi başarımı B@1 skoru 0.687 olan Adam en iyileme algoritması ile alındığı görülmüştür. Aynı zamanda test veri kümesi üzerinde çalıştırılan modelin sonuçları sırasıyla başarılı, kısmen başarılı ve başarısız sonuçlar Şekil 3.13, 3.14 ve 3.15'te gösterilmiştir. Sonuçları incelediğimizde oldukça başarılı sonuçlar aldığımız gibi başarısız ve geliştirmeye açık olan sonuçlarda alınmıştır.

Gelecek çalışmalar için ilk olarak önerimiz veri kümesinin daha hatasız ve daha az kirli veriden oluşturulması amaçlanmalıdır. Sonrasında veri ön işleme ve filtreleme aşamaları için geliştirilen sınıflandırma modelinin görüntü – başlık ikili verisiyle birlikte eğitilip buna göre veri filtreleme işlemi yapılabilir. Mimarinin kodlayıcı kısmında farklı CNN mimariler denenebilir aynı zaman farklı kelime kalıplama yöntemleri ile eğitim tekrarlanabilir. Son olarak ise yakın zamanlaarda görüntü başlıklama için geliştirilen GAN (Generative Adversarial Network) tabanlı mimariler ile eğitim tekrarlanıp elde edilen sonuçlar karşılaştırılabilir.

KAYNAKLAR

[1] Hossain M. Z., Sohel F., Shiratuddin M. F., Laga H. A comprehensive survey of deep learning for image captioning, *ACM Computing Surveys*, 2019, **51**(6), 1-36.

[2] Farhadi A., Hejrati M., Sadeghi M. A., Young P., Rashtchian C., Hockenmaier J., Forsyth D., Every picture tells a story: Generating sentences from images, *Lecture Notes in Computer Science, vol 6314*, DOI: 10.1007/978-3-642-15561-1_2

[3] Li S., Kulkarni G., Berg T., Berg A., Choi Y., Composing simple image descriptions using web-scale n-grams, *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, Oregon, USA, 23-24 June 2011.

[4] Ordonez V., Kulkarni G., Berg T. L., Im2text: Describing images using 1 million captioned photographs, *Lecture Notes in Computer Science, vol 8692*, DOI: 10.1007/978-3-319-10593-2_35

[5] Yunchao G., Liwei W., Micah H., Julia H., Svetlana L., Improving imagesentence embeddings using large weakly annotated photo collections, *Lecture Notes in Computer Science, vol 8692*, DOI: 10.1007/978-3-319-10593-2_35

[6] Micah H., Peter Y., Julia H., Framing image description as a ranking task: Data, models and evaluation metrics, *Journal of Artificial Intelligence Research*, 2013 **47**(1), 853–899.

[7] Chen S., Chuang G., Ram N., Automatic concept discovery from parallel text and visual corpora, *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 7-13 December 2015.

[8] Kiros R., Salakhutdinov R., Zemel R., Multimodal neural language models. *Proceedings of the International Conference on Machine Learning*, Beijing, China, 21–26 June 2014, 595–603.

[9] Mao J., Xu W., Yang Y., Wang J., Yuille A. L., Explain images with multimodal recurrent neural networks, *arXiv preprint arXiv*, 2014, 1410.1090.

[10] Kiros R., Salakhutdinov R., Zemel R. S., Unifying visual-semantic embeddings with multimodal neural language models, *arXiv preprint arXiv*, 2014, 1411.2539.

[11] Karpathy A., Fei-Fei L., Deep visual-semantic alignments for generating image descriptions, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, **39**(4), 3128-3137.

[12] Vinyals O., Toshev A., Bengio S., Erhan D., Show and tell: A neural image caption generator, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, USA, 7-12 June 2015.

[13] Xu K., Ba J., Kiros R., Cho K., Courville A., Salakhudinov R., Bengio Y., Show, attend and tell: Neural image caption generation with visual attention, *International conference on machine learning*, Lille, France, 07-09 July 2015.

[14] Karpathy A., Joulin A., Fei-Fei L., Deep fragment embeddings for bidirectional image sentence mapping, *arXiv preprint arXiv*, 2014, 1406.5679.

[15] Mao J., Xu W., Yang Y., Wang J., Huang Z., Yuille A., Deep captioning with multimodal recurrent neural networks (m-rnn), *arXiv preprint arXiv*, 2014, 1412.6632.

[16] Chen X., Zitnick C. L., Learning a recurrent visual representation for image caption generation, *arXiv preprint arXiv*, 2014, 1411.5654.

[17] Fang H., Gupta S., Iandola F., Srivastava R. K., Deng L., Dollár P., Zweig G. From captions to visual concepts and back, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, USA, 7-12 June 2015.

[18] Jia X., Gavves E., Fernando B., Tuytelaars T., Guiding the long-short term memory model for image caption generation, *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile, 7-13 December 2015.

[19] Sugano Y., Bulling A., Seeing with humans: Gaze-assisted neural image captioning, *arXiv preprint arXiv*, 2016, 1608.05203.

[20] Mathews A., Xie L., He X., Senticap: Generating image descriptions with sentiments, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016, **30**(1), 3574-3580.

[21] Wang C., Yang H., Bartz C., Meinel C., Image captioning with deep bidirectional LSTMs, *Proceedings of the 24th ACM international conference on Multimedia*, 2016, **14**(2), 988-997.

[22] Johnson J., Karpathy A., Fei-Fei L., Densecap: Fully convolutional localization networks for dense captioning, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, USA, 27-30 June 2016.

[23] Mao J., Huang J., Toshev A., Camburu O., Yuille A. L., Murphy K., Generation and comprehension of unambiguous object descriptions, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, USA, 27-30 June 2016.

[24] Wang M., Song L., Yang X., Luo, C., A parallel-fusion RNN-LSTM architecture for image caption generation, *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, USA, 25-28 September 2016.

[25] Tran K., He X., Zhang L., Sun J., Carapcea C., Thrasher C., Sienkiewicz C., Rich image captioning in the wild, *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, Las Vegas, USA, 26 June-1 July 2016.

[26] Ma S., Han Y., Describing images by feeding LSTM with structural words, *2016 IEEE International Conference on Multimedia and Expo (ICME)*, Seattle, USA, 11-15 July 2016.

[27] You Q., Jin H., Wang Z., Fang C., Luo J., Image captioning with semantic attention, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, USA, 27-30 June 2016.

[28] Hendricks L. A., Venugopalan S., Rohrbach M., Mooney R., Saenko K., Darrell T., Deep compositional captioning: Describing novel object categories without paired training data, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, USA, 27-30 June 2016.

[29] Yao T., Pan Y., Li Y., Qiu Z., Mei T., Boosting image captioning with attributes, *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 22-29 Oct. 2017.

[30] Lu J., Xiong C., Parikh D., Socher R., Knowing when to look: Adaptive attention via a visual sentinel for image captioning, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, USA, 21-26 July 2017.

[31] Chen L., Zhang H., Xiao J., Nie L., Shao J., Liu W., Chua T. S., Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, USA, 21-26 July 2017.

[32] Gan Z., Gan C., He X., Pu Y., Tran K., Gao J., Deng L., Semantic compositional networks for visual captioning, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, USA, 21-26 July 2017.

[33] Ren Z., Wang X., Zhang N., Lv X., Li L. J., Deep reinforcement learning-based image captioning with embedding reward, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, USA, 21-26 July 2017.

[34] Chunseong P. C., Kim B., Kim G., Attend to you: Personalized image captioning with context sequence memory networks, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, USA, 21-26 July 2017.

[35] Wang Y., Lin Z., Shen X., Cohen S., Cottrell G. W., Skeleton key: Image captioning by skeleton-attribute decomposition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, USA, 21-26 July 2017.

[36] Tavakoli H. R., Shetty R., Borji A., Laaksonen J., Paying attention to descriptions generated by image captioning models, *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 22-29 Oct. 2017.

[37] Liu C., Mao J., Sha F., Yuille A., Attention correctness in neural image captioning, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017, **31**(1), 4176-4182.

[38] Gan C., Gan Z., He X., Gao J., Deng L., Stylenet: Generating attractive visual captions with styles, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, 21-26 July 2017.

[39] Dai B., Fidler S., Urtasun R., Lin, D., Towards diverse and natural image descriptions via a conditional gan, *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 22-29 Oct. 2017.

[40] Shetty R., Rohrbach M., Anne H. L., Fritz M., Schiele B., Speaking the same language: Matching machine to human captions by adversarial training, *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 22-29 Oct. 2017.

[41] Liu S., Zhu Z., Ye N., Guadarrama S., Murphy K., Improved image captioning via policy gradient optimization of spider, *Proceedings of the IEEE international conference on computer vision*, Venice, 22-29 Oct. 2017.

[42] Gu J., Wang G., Cai J., Chen T., An empirical study of language cnn for image captioning, *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 22-29 Oct. 2017.

[43] Yao T., Pan Y., Li Y., Mei T., Incorporating copying mechanism in image captioning for learning novel objects, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, USA, 21-26 July 2017.

[44] Rennie S. J., Marcheret E., Mroueh Y., Ross J., Goel V., Self-critical sequence training for image captioning, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, 21-26 July 2017.

[45] Venugopalan S., Anne H. L., Rohrbach M., Mooney R., Darrell T., Saenko K., Captioning images with diverse objects, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, USA, 21-26 July 2017.

[46] Zhang L., Sung F., Liu F., Xiang T., Gong S., Yang, Y., Hospedales T. M., Actor-critic sequence training for image captioning, *arXiv preprint arXiv*, 2017, 1706.09601.

[47] Wu Q., Shen C., Wang P., Dick A., Van D. H. A., Image captioning and visual question answering based on attributes and external knowledge, *IEEE transactions on pattern analysis and machine intelligence*, 2017, **40**(6), 1367-1381.

[48] Aneja J., Deshpande A., Schwing A. G., Convolutional image captioning, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City, UT, 18-23 June 2018.

[49] Wang Q., Chan A. B., Cnn+ cnn: Convolutional decoders for image captioning, *arXiv preprint arXiv*, 2018, 1805.09019.

[50] Feng Y., Ma L., Liu W., Luo J., Unsupervised image captioning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[51] Shuster K., Humeau S., Hu H., Bordes A., Weston J., Engaging image captioning via personality, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[52] Li Y., Yao T., Pan Y., Chao H., Mei T., Pointing novel objects in image captioning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[53] Yang X., Tang K., Zhang H., Cai J., Auto-encoding scene graphs for image captioning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[54] Yin G., Sheng L., Liu B., Yu N., Wang X., Shao J., Context and attribute grounded dense captioning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[55] Qin Y., Du J., Zhang Y., Lu H., Look back and predict forward in image captioning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[56] Gao J., Wang S., Wang S., Ma S., Gao W., Self-critical n-step training for image captioning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[57] Zheng Y., Li Y., Wang S., Intention oriented image captions with guiding objects, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[58] Wang Q., Chan A. B., Describing like humans: on diversity in image captioning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[59] Dognin P., Melnyk I., Mroueh Y., Ross J., Sercu T., Adversarial semantic alignment for improved image captions, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[60] Guo L., Liu J., Yao P., Li J., Lu, H., Mscap: Multi-style image captioning with unpaired stylized text, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[61] Deshpande A., Aneja J., Wang L., Schwing A. G., Forsyth D., Fast, diverse and accurate image captioning guided by part-of-speech, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[62] Biten A. F., Gomez L., Rusinol M., Karatzas D., Good news, everyone! context driven entity-aware captioning for news images, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[63] Kim D. J., Choi J., Oh T. H., Kweon I. S., Dense relational captioning: Triple-stream networks for relationship-based captioning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[64] Cornia M., Baraldi L., Cucchiara R., Show, control and tell: A framework for generating controllable and grounded captions, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, 15-20 June 2019.

[65] Li X., Jiang S., Han J., Learning object context for dense captioning, *Proceedings of the AAAI conference on artificial intelligence*, 2019, **33**(1), 8650-8657.

[66] Chen C., Mu S., Xiao W., Ye Z., Wu L., Ju Q., Improving image captioning with conditional generative adversarial nets, *Proceedings of the AAAI Conference on Artificial Intelligence,* 2019, **33**(1), 8142-8150.

[67] Fan Z., Wei Z., Wang S., Huang X. J., Bridging by word: Image grounded vocabulary construction for visual captioning, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, 28 July - 2 August 2019.

[68] Demirel B., Cinbis R. G., Ikizler-Cinbis N., Image captioning with unseen objects, *arXiv preprint arXiv*, 2019, 1908.00047.

[69] Sammani F., Elsayed M., Look and modify: Modification networks for image captioning, *arXiv preprint arXiv*, 2019, 1909.03169.

[70] Khare A., Huber M., Show, Infer and Tell: Contextual Inference for Creative Captioning, *British Machine Vision Conference* (*BMVC)*, Cardiff, UK, 9-12 September 2019.

[71] Yan S., Hua Y., Robertson N., SC-RANK: Improving Convolutional Image Captioning with Self-Critical Learning and Ranking Metric-based Reward, *British Machine Vision Conference* (*BMVC)*, Cardiff, UK, 9-12 September 2019.

[72] Yao T., Pan Y., Li Y., Mei T., Hierarchy parsing for image captioning, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Korea (South), 27 Oct.-2 Nov. 2019.

[73] Li G., Zhu L., Liu P., Yang Y., Entangled transformer for image captioning, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Korea (South), 27 Oct.-2 Nov. 2019.

[74] Huang L., Wang W., Chen J., Wei X. Y., Attention on attention for image captioning, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Korea (South), 27 Oct.-2 Nov. 2019.

[75] Ke L., Pei W., Li R., Shen X., Tai Y. W., Reflective decoding network for image captioning, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Korea (South), 27 Oct.-2 Nov. 2019.

[76] Yang X., Zhang H., Cai J., Learning to collocate neural modules for image captioning, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Korea (South), 27 Oct.-2 Nov. 2019.

[77] Herdade S., Kappeler A., Boakye K., Soares J., Image captioning: Transforming objects into words, *arXiv preprint arXiv*, 2019, 1906.05963.

[78] Huang L., Wang W., Xia Y., Chen J., Adaptively aligned image captioning via adaptive attention time, *arXiv preprint arXiv*, 2019, 1909.09060.

[79] Chen F., Ji R., Ji J., Sun X., Zhang B., Ge X., Wang Y., Variational structured semantic inference for diverse image captioning, *33rd Conference on Neural Information Processing Systems*, Vancouver, Canada, 8–14 November 2019.

[80] Liu F., Liu Y., Ren X., He X., Sun X., Aligning visual regions and textual concepts for semantic-grounded image representations, *arXiv preprint arXiv*, 2019, 1905.06139.

[81] Liu F., Ren X., Liu Y., Lei K., Sun X., Exploring and distilling cross-modal information for image captioning, *arXiv preprint arXiv*, 2020, 2002.12585.

[82] Wang H., Wang H., Xu K., Swell-and-Shrink: Decomposing Image Captioning by Transformation and Summarization, *International Joint Conference on Artificial Intelligence*, Macao, China, 10-16 August 2019.

[83] Yan S., Xu J., Liu Y., Xu L., Hornet: A hierarchical offshoot recurrent network for improving person re-id via image captioning, *arXiv preprint arXiv*, 2019, 1908.04915.

[84] Jiang M., Huang Q., Zhang L., Wang X., Zhang P., Gan Z., Gao J., TIGEr: text-to-image grounding for image caption evaluation, *arXiv preprint arXiv*, 2019, 1909.02050.

[85] Zhao W., Wu X., Zhang X., MemCap: Memorizing style knowledge for image captioning, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, **34**(7), 12984-12992.

[86] Zhou L., Palangi H., Zhang L., Hu H., Corso J., Gao J., Unified vision-language pre-training for image captioning and vqa, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, **34**(7), 13041-13049.

[87] Wang L., Bai Z., Zhang Y., Lu H., Show, Recall, and Tell: Image Captioning with Recall Mechanism, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, **34**(7), 12176-12183.

[88] Seo P. H., Sharma P., Levinboim T., Han B., Soricut R., Reinforcing an image caption generator using off-line human feedback, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, **34**(3), 2693-2700.

[89] Liu J., Wang K., Xu C., Zhao Z., Xu R., Shen Y., Yang M., Interactive dual generative adversarial networks for image captioning, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, **34**(7), 11588-11595.

[90] Cao T., Han K., Wang X., Ma L., Fu Y., Jiang Y. G., Xue X, Feature deformation meta-networks in image captioning of novel objects, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, **34**(7), 10494-10501.

[91] Hou J., Wu X., Zhang X., Qi Y., Jia Y., Luo J., Joint Commonsense and Relation Reasoning for Image and Video Captioning, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, **34**(7), 10973-10980.

[92] Zhang W., Ying Y., Lu P., Zha H., Learning Long-and Short-Term User Literal-Preference with Multimodal Hierarchical Transformer Network for Personalized Image Caption, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, **34**(5), 9571-9578.

[93] Guo L., Liu J., Zhu X., Yao P., Lu S., Lu, H., Normalized and Geometry-Aware Self-Attention Network for Image Captioning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, USA, 13-19 June 2020.

[94] Chen S., Jin Q., Wang P., Wu Q., Say as you wish: Fine-grained control of image caption generation with abstract scene graphs, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, USA, 13-19 June 2020.

[95] Shi Z., Zhou X., Qiu X., Zhu X., Improving Image Captioning with Better Use of Captions, *arXiv preprint arXiv*, 2020, 2006.11807.

[96] Alikhani M., Sharma P., Li S., Soricut R., Stone, M., Clue: Cross-modal coherence modeling for caption generation, *arXiv preprint arXiv*, 2020, 2005.00908.

[97] Yi Y., Deng H., Hu J., Improving image captioning evaluation by considering inter references variance, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 6-8 July 2020.

[98] Jin J., Fu K., Cui R., Sha F., Zhang C., Aligning where to see and what to tell: image caption with region-based attention and scene factorization, *arXiv preprint arXiv*, 2015, 1506.06272.

[99] Yang Z., Yuan Y., Wu Y., Salakhutdinov R., Cohen W., Encode, review, and decode: Reviewer module for caption generation, *arXiv preprint arXiv*, 2016, 1605.07912.

[100] Pedersoli M., Lucas T., Schmid C., Verbeek J., Areas of attention for image captioning, *Proceedings of the IEEE international conference on computer vision*, Venice, 22-29 October 2017.

[101] Plummer B. A., Wang L., Cervantes C. M., Caicedo J. C., Hockenmaier J., Lazebnik S., Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models, *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile, 7-13 Dec. 2015.

[102] Lin T. Y., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Zitnick C. L., Microsoft coco: Common objects in context, *European conference on computer vision*, Zurich, Switzerland, 6-12 September 2014.

[103] Yang L., Tang K., Yang J., Li L. J., Dense captioning with joint inference and visual context, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, 21-26 July 2017.

[104] Zhang J., Shih K., Tao A., Catanzaro B., Elgammal A., An interpretable model for scene graph generation, *arXiv preprint arXiv*, 2018, 1811.09543.

[105] Kinghorn P., Zhang L., Shao L., A region-based image caption generator with refined descriptions, *Neurocomputing*, 2018, **272**(C), 416-424.

[106] Fan C., Crandall D. J., Deepdiary: Automatically captioning lifelogging image streams, *European Conference on Computer Vision*, Amsterdam, The Netherlands, 8-10 October 2016.

[107] Staniūtė R., Šešok D., A systematic literature review on image captioning, *Applied Sciences*, 2019, **9**(10), 2024.

[108] Papineni K., Roukos S., Ward T., Zhu W. J., BLEU: a method for automatic evaluation of machine translation, *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, Pennsylvania, USA, July 2002.

[109] Banerjee S., Lavie A., METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, Michigan, USA, June 2005.

[110] Lin C. Y., Rouge: A package for automatic evaluation of summaries, *Text summarization branches out*, Barcelona, Spain, July 2004.

[111] Vedantam R., Zitnick C. L., Parikh D., Cider: Consensus-based image description evaluation, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, USA, 7-12 June 2015.

[112] Anderson P., Fernando B., Johnson M., Gould S., Spice: Semantic propositional image caption evaluation, *European Conference on Computer Vision*, Amsterdam, The Netherlands, 11-14 October 2016.

[113] Jadon S, Introduction to Different Activation Functions for Deep Learning, https://medium.com/@shrutijadon10104776/survey-on-activation-functions-for-deep-learning-9689331ba092/ (Ziyaret Tarihi: 10 Ekim 2020)

[114] Chollet F., *Python ile Derin Öğrenme*, 1. baskı, Buzdağı Yayınevi, Eryaman, Ankara, 2019.

[115] Gazel S. E. R., Bati C. T., Derin Sinir Ağları ile En İyi Modelin Belirlenmesi: Mantar Verileri Üzerine Keras Uygulaması, *Yüzüncü Yıl Üniversitesi Tarım Bilimleri Dergisi*, **29**(3), 406-417.

[116] LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D., Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989, **1**(4), 541-551.

[117] LeCun Y., Bottou L., Bengio Y., Haffner P., Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 1998, **86**(11), 2278-2324.

[118] https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html (Ziyaret Tarihi: 12 Ekim 2020).

[119] Saha S., A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way, https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (Ziyaret Tarihi: 14 Ekim 2020).

[120] Krizhevsky A., Sutskever I., Hinton G. E., Imagenet classification with deep convolutional neural networks, *Communications of the ACM*, 2017, **60**(6), 84-90.

[121] Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Rabinovich A., Going deeper with convolutions, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, USA, 7-12 June 2015.

[122] Simonyan K., Zisserman A., Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv*, 2014, 1409.1556.

[123] He K., Zhang X., Ren S., Sun J., Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, USA, 27-30 June 2016.

[124] Olah C., Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs (Ziyaret Tarihi: 7 Kasım 2020).

[125] Amidi A., Amidi S., Recurrent Neural Networks cheatsheet, https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks (Ziyaret Tarihi: 7 Kasım 2020).

[126] Bengio Y., Simard P., Frasconi P., Learning long-term dependencies with gradient descent is difficult, *IEEE transactions on neural networks*, 1994, **5**(2), 157-166.

[127] Hochreiter S., Schmidhuber J., Long short-term memory. *Neural computation*, 1997, **9**(8), 1735-1780.

[128] Tadvalkar A., IMDB Review Sentiment Classification using RNN LSTM, https://kgptalkie.com/imdb-review-sentiment-classification-using-rnn-lstm/ (Ziyaret Tarihi: 29 Ağustos 2020).

[129] Cho K., Merriënboer V. B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y., Learning phrase representations using RNN encoder-decoder for statistical machine translation, *arXiv preprint arXiv*, 2014, 1406.1078.

[130] Jozefowicz R., Zaremba W., Sutskever I., An empirical exploration of recurrent network architectures, *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, Lille, France, 7-9 July 2015.

[131] Tsai C. F., Bag-of-words representation in image annotation: A review, *International Scholarly Research Notices*, **2012**(1).

[132] Mikolov T., Chen K., Corrado G., Dean J., Efficient estimation of word representations in vector space, *arXiv preprint arXiv*, 2013, 1301.3781.

[133] Word2Vec: TensorFlow Vector Representation Of Words, https://data-flair.training/blogs/tensorflow-word2vec/ (Ziyaret Tarihi: 10 Ekim 2020)

[134] Pennington J., Socher R., Manning C. D., Glove: Global vectors for word representation, *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, Qatar, 25–29 October 2014.

[135] Bojanowski P., Grave E., Joulin A., Mikolov T., Enriching word vectors with subword information, *Transactions of the Association for Computational Linguistics*, 2017, **5**, 135-146.

[136] Bahdanau D., Cho K., Bengio Y., Neural machine translation by jointly learning to align and translate, *arXiv preprint arXiv*, 2014, 1409.0473.

[137] https://www.tensorflow.org/tutorials/text/image_captioning/ (Ziyaret Tarihi: 21 Kasım 2020)

[138] https://cloud.google.com/tpu/docs/inception-v3-advanced (Ziyaret Tarihi: 1 Şubat 2021)

KİŞİSEL YAYIN VE ESERLER

[1] **Atıcı B.**, Omurca S. İ., Ekinci E., Kullanıcı Şikayetlerindeki Ürün Özelliklerinin Gizli Dirichlet Ayırımı ile Saptanması, *2017 Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansı (UBMK)*, Antalya, Türkiye, 5-8 Ekim 2017.

[2] **Atıcı B.**, Omurca S. İ., Generating Classified Ad Product Image Titles with Image Captioning, *2020 Uluslararası Mühendislikte Yapay Zeka ve Uygulamalı Matematik Konferansı (UMYMK 2020)*, Antalya, Türkiye, 9-11 Ekim 2020.

ÖZGEÇMİŞ

Birkan Atıcı, Samsun'da lise öğrenimini tamamladı. Lisans öğrenimine 2013 yılında Kocaeli Üniversitesi Bilgisayar Mühendisliğinde başladı ve 2017 yılında tamamladı. 2017 yılında Kocaeli Üniversitesi Bilgisayar Mühendisliği Ana Bilim Dalı'nda yüksek lisans eğitimine başladı. Yüksek lisans eğitimi süresince derin öğrenme, doğal diş işleme ve görüntü işleme alanları üzerinde çalışmıştır. Yüksek lisans eğitiminin yanı sıra Sahibinden Bilgi Teknolojileri şirketinde Yazılım Uzmanı olarak çalışmaktadır.