



**EVRIŞİMLİ VE YİNELEMELİ SİNİR AĞLARI İLE GÖRÜNTÜLERE
BAŞLIK ATAMA**

Tunahan KUNT

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ŞUBAT 2021

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Tunahan KUNT

05/02/2021

EVİRİŞİMLİ VE YİNELEMELİ SINIR AĞLARI İLE GÖRÜNTÜLERE BAŞLIK
(Yüksek Lisans Tezi)

Tunahan KUNT

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Şubat 2021

ÖZET

Görüntülere başlık atama, insan yardımı olmaksızın bilgisayar tarafından görüntü içeriğini betimleyen cümle kurma olarak tanımlanabilir. Görüntülere başlık atama problemi ile sadece görüntüdeki objeleri anlamak değil aynı zamanda objelerin birbiri ile ilişkilerini anlayarak bunları bir cümle haline getirmek amaçlanmıştır. Bu bağlamda görüntülere başlık atama problemi, bilgisayarla görü ve doğal dil işleme alanlarının kesişiminde bulunur ve her iki alanın da zorluklarını bünyesinde barındırır. Karşılaşılan zorlukları çözmek adına farklı mimarilere sahip çalışmalar ortaya konulmuştur. Fakat literatürdeki çalışmaların birçoğu Cermen dil ailesine mensup dillerde özellikle İngilizce dilinde yoğunlaşmıştır. Kendi ana dilimiz olan Türkçe dilinde ise görüntülere başlık atama alanında yapılan çalışma sayısı oldukça azdır. Bu bağlamda bu tez, Türkçe dilinde görüntülere başlık atama alanında yapılmıştır. Önerilen çalışma kapsamında, Türkçe dilinin sondan eklemeli yapısı gözetilerek işlenen veri kümesi ile halihazırda İngilizce dilinde görüntülere başlık atama problemini çözmek için kullanılan modeller ve bu modeller üzerine yapılan birtakım iyileştirmeler ile oluşturulan yeni modeller eğitilmiştir. Bazı modellerde topluluk öğrenmesinin kullanıldığı, bazı modellerde de dikkat mekanizmasının eklendiği çalışmada evrişimli ve yinelemeli sinir ağlarından oluşan kodlayıcı-kod çözücü mimariye sahip modellerin çıktıları açgözlü arama veya ışın araması yöntemleriyle cümle haline getirilmiştir ve oluşturulan cümleler iki dilli değerlendirme ölçeği yöntemiyle değerlendirilmiştir. Yapılan deneylerin sonucunda en yüksek 0.424 BLEU-1 puanı elde edilmiştir. Elde edilen puan görüntülere başlık atama alanındaki Türkçe veri kümesi ile yapılan araştırmalar arasında en yüksek puan olmuştur.

Bilim Kodu : 92432

Anahtar Kelimeler : Derin Öğrenme, Evrişimli Sinir Ağları, Yinelemeli Sinir Ağları, Dikkat Mekanizması, Topluluk Öğrenmesi, Görüntülere Başlık, Atama

Sayfa Adedi : 86

Danışman : Prof. Dr. Necaattin BARIŞCI

IMAGE CAPTIONING WITH CONVOLUTIONAL AND RECURRENT NEURAL NETWORKS

(M. Sc. Thesis)

Tunahan KUNT

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

February 2021

ABSTRACT

Image captioning can be defined as describing a content of the image without human help. With the image captioning, it is aimed not only to understand the objects in the image, but also to understand the relations between the objects then turn it to the meaningful sentence. In this context, the problem of image captioning is located at the intersection of computer vision and natural language processing and it includes the difficulties of both fields. In order to solve the faced difficulties, different studies have been were put forward with different architectures. However, most of the studies in the literature have concentrated on languages belonging to the Germanic language family, especially in English. In the Turkish language, which is our native language, the number of studies in the image captioning field is very few. In this context, this dissertation conducted in the field of image captioning in Turkish language. Within the scope of the proposed study, models that are currently used to solve the problem of assigning captions to images in English language and new models which created with some improvements made on the previous models have been trained with the dataset processed by considering the agglutinative structure of the Turkish language. In this study, some models have been created by ensemble learning and some models have been created by attention mechanism. All models that are created have encoder-decoder architecture and these models consist of convolutional and recurrent neural networks. The outputs of the models were put into sentences by greedy search or beam search techniques. Finally, created sentences were evaluated with the BLEU score method. As a result of the experiments, the highest BLEU-1 score of 0.424 was obtained. The score obtained was the highest score among the Turkish dataset studies in the field of image captioning.

Science Code : 92432

Key Words : Deep Learning, Convolutional Neural Networks, Recurrent Neural Networks, Attention Mechanism, Ensemble Learning, Image Captioning

Page Number : 86

Supervisor : Prof. Dr. Necaattin BARIŞCI

TEŞEKKÜR

Tez çalışmam boyunca her konuda yol gösteren, verdiği bilgiler ile derin öğrenme alanına ilgimin artmasına sağlayan değerli hocam ve danışmanım Prof. Dr. Necaattin BARIŞÇI'ya teşekkür ederim. Kıymetli ve değerli eşime sevgilerle.



İÇİNDEKİLER

	Sayfa
ÖZET.....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	xi
ŞEKİLLERİN LİSTESİ.....	xii
SİMGELER VE KISALTMALAR.....	xv
1. GİRİŞ.....	1
2. LİTERATÜR TARAMASI.....	5
2.1. Şablon Tabanlı Modeller.....	5
2.2. Erişim Tabanlı Modeller.....	6
2.3. Çok Boyutlu Modeller.....	6
3. MATERYAL VE METOTLAR.....	13
3.1. Makine Öğrenimi.....	13
3.1.1. Denetimli öğrenme.....	13
3.1.2. Denetimsiz öğrenme.....	13
3.2. Yapay Sinir Ağları.....	14
3.3. Derin Öğrenme.....	16
3.4. Kelime Gömmeleri.....	17
3.5. Evrişimli Sinir Ağları.....	19
3.5.1. Evrişim katmanı	21
3.5.2. Kaydırma adımı.....	21
3.5.3. Sıfır ekleme.....	22

	Sayfa
3.5.4. Ortaklama katmanı.....	22
3.5.5. Çıktı katmanı	23
3.6. Görüntü Özelliklerinin Çıkarılması.....	24
3.6.1. VGG16 mimarisi	25
3.6.2. VGG19 mimarisi.....	25
3.6.3. ResNet mimarisi.....	26
3.6.4. ResNet v2 mimarisi.....	27
3.6.5. Inception v3 mimarisi.....	28
3.6.6. InceptionResNet v2 mimarisi.....	28
3.6.7. Xception mimarisi.....	29
3.6.8. MobileNet mimarisi.....	29
3.6.9. EfficientNet mimarisi.....	30
3.7. Yinelemeli Sinir Ağları.....	31
3.8. Dil Modeli.....	33
3.8.1. Uzun kısa süreli bellek	33
3.8.2. Kapılı yineleyen birim.....	34
3.8.3. İki yönlü uzun kısa süreli bellek.....	35
3.9. Kodlayıcı-Kod Çözücü Mimarisi.....	36
3.10. Görüntülere Başlık Atama Modelleri.....	37
3.11. Dikkat Mekanizması.....	39
3.11.1. Lokal dikkat mekanizması	41
3.11.2. Global dikkat mekanizması.....	42
3.12. Derin Öğrenme Hiper Parametreleri.....	42
3.12.1. Öğrenme hızı	43

	Sayfa
3.12.2. Mini grup boyutu.....	43
3.12.3. Tur sayısı.....	43
3.12.4. Katman sayısı ve nöron sayısı.....	44
3.12.5. Optimizasyon algoritması.....	44
3.12.6. Başlangıç ağırlıklarının belirlenmesi.....	44
3.12.7. Seyreltme.....	44
3.13. Topluluk Öğrenmesi.....	45
3.14. Veri Kümesi.....	46
4. GÖRÜNTÜLERE BAŞLIK ATAMA İÇİN ÖNERİLEN YAKLAŞIM.	49
4.1. Veri Ön İşleme.....	49
4.2. Veri Dönüşümü.....	50
4.3. Model.....	53
4.3.1. Enjeksiyon modeli	53
4.3.2. Alternatif enjeksiyon modeli.....	54
4.3.3. Birleştirme modeli.....	55
4.3.4. Alternatif birleştirme modeli.....	55
4.3.5. BirEn modeli.....	56
4.3.6. Lokal dikkat mekanizması modeli.....	56
4.3.7. Global dikkat mekanizması modeli.....	57
4.3.8. Topluluk 1 modeli.....	58
4.3.9. Topluluk 2 modeli.....	59
4.3.10. Topluluk 3 modeli.....	59
4.4. Değerlendirme.....	60
5. ARAŞTIRMA BULGULARI.....	63

	Sayfa
5.1. Seçilen Teknolojiler ve Test Ortamı.....	63
5.1.1. Python.....	63
5.1.2. Tensorflow.....	63
5.1.3. Keras.....	63
5.1.4. Test Ortamı.....	64
5.2. Kullanılan Hiper Parametreler.....	64
5.3. Sonuçlar.....	65
6. SONUÇ VE ÖNERİLER.....	75
KAYNAKLAR.....	77
ÖZGEÇMİŞ.....	86

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. TasvirEt veri kümesi istatistik bilgileri.....	47
Çizelge 3.2. TasvirEt veri kümesindeki en sık kullanılan 15 kelime ve frekansları.....	47
Çizelge 4.1. Veri kümesindeki “afış” kelimesi kökünden türeyen kelimelerin miktarı..	51
Çizelge 4.2. Görüntülere başlık atama probleminde modelin girdileri ve çıktısı.....	52
Çizelge 5.1. Model eğitimi sırasında kullanılan hiper parametreler.....	64
Çizelge 5.2. Birleştirme modeli ile elde edilen sonuçlar.....	66
Çizelge 5.3. Alternatif birleştirme, enjeksiyon, alternatif enjeksiyon ve BirEn modelleri ile elde edilen Bleu skorları.....	67
Çizelge 5.4. Dikkat mekanizması kullanılarak geliştirilen modeller ile elde edilen sonuçlar.....	70
Çizelge 5.5. Topluluk öğrenmesi kullanılarak geliştirilen modeller ile elde edilen sonuçlar.....	71
Çizelge 5.6. Karşılaştırmalı sonuçlar.....	72

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Karphaty tarafından önerilen görüntülere başlık atama modeli.....	7
Şekil 2.2. Vinalys ve diğerleri tarafından önerilen görüntülere başlık atama modeli.....	9
Şekil 3.1. Nöron yapısı.....	14
Şekil 3.2. Aktivasyon fonksiyonları.....	15
Şekil 3.3. Yapay sinir ağı.....	16
Şekil 3.4. Derin sinir ağları temsili gösterimi.....	17
Şekil 3.5. Word2vec mimarisi.....	18
Şekil 3.6. SKT (solda) ve skip-gram(sağda) modelleri.....	19
Şekil 3.7. 3 renk kanallı 4x4'lük bir görüntünün gösterimi.....	20
Şekil 3.8. Örnek evrişimli sinir ağı mimarisi.....	20
Şekil 3.9. Evrişim katmanı uygulaması.....	21
Şekil 3.10. Kaydırma adımı (stride) = 2 ile evrişim uygulaması.....	22
Şekil 3.11. Sıfır ekleme (padding) = 2 ile evrişim uygulaması.....	22
Şekil 3.12. Ortaklama katmanı örneği.....	23
Şekil 3.13. Evrişimli sinir ağı mimarisi.....	23
Şekil 3.14. VGG16 mimarisi.....	25
Şekil 3.15. VGG19 mimarisi.....	26
Şekil 3.16. 20 ve 56 boyutlu iki sinir ağının hata oranları. Eğitim hatası (solda), test hatası (sağda).....	26
Şekil 3.17. Artık blok örneği.....	27
Şekil 3.18. ResNet v1 ve ResNet v2 artık blokları.....	27
Şekil 3.19. Inception v3 mimarisi.....	28
Şekil 3.20. InceptionResNet v2 mimarisi.....	29

Şekil	Sayfa
Şekil 3.21. MobileNet mimarisi.....	30
Şekil 3.22. Model ölçeklendirme.....	31
Şekil 3.23. Yinelemeli sinir ağı hücre yapısı.....	32
Şekil 3.24. Yinelemeli sinir ağı mimarisi.....	33
Şekil 3.25. Uzun kısa süreli bellek hücresi.....	34
Şekil 3.26. Kapılı yineleyen birim hücresi.....	35
Şekil 3.27. İki yönlü uzun kısa süreli bellek mimarisi.....	35
Şekil 3.28. Kodlayıcı-kod çözücü mimarisi.....	36
Şekil 3.29. İlk enjeksiyon yaklaşımı.....	37
Şekil 3.30. Ön enjeksiyon yaklaşımı.....	38
Şekil 3.31. Paralel enjeksiyon yaklaşımı.....	38
Şekil 3.32. Birleştirme yaklaşımı.....	39
Şekil 3.33. Dikkat tabanlı başlık oluşturma modelinin genel yapısı.....	40
Şekil 3.34. Dikkat mekanizması kullanılan görüntülere başlık atama mimarisi.....	41
Şekil 3.35. Lokal dikkat mekanizması çalışma akışı.....	42
Şekil 3.36. (a) 2 gizli katmanlı standart sinir ağı (b) Seyreltme uygulanarak üretilmiş sinir ağı.....	45
Şekil 3.37. Topluluk öğrenmesi.....	46
Şekil 4.1. Üst seviye çalışma adımları.....	49
Şekil 4.2. Enjeksiyon modeli.....	54
Şekil 4.3. Alternatif enjeksiyon modeli.....	54
Şekil 4.4. Birleştirme modeli.....	55
Şekil 4.5. Alternatif birleştirme modeli.....	55
Şekil 4.6. BirEn modeli.....	56
Şekil 4.7. Lokal dikkat mekanizması modeli.....	57

Şekil	Sayfa
Şekil 4.8. Global dikkat mekanizması modeli.....	58
Şekil 4.9. Topluluk 1 modeli.....	58
Şekil 4.10. Topluluk 2 modeli.....	59
Şekil 4.11. Topluluk 3 modeli.....	59
Şekil 4.12. Işın araması.....	60
Şekil 5.1. TasvirEt veri kümesi görüntülerinden oluşturulan açıklamalar (üst satır başarılı çıktıları, orta satır orta seviye çıktıları, alt satır ise başarısız çıktıları göstermektedir).....	69



SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar

Açıklamalar

ESA

Evrişimli sinir ağları

İYUKSB

İki yönlü uzun kısa süreli bellek

KYB

Kapılı yineleyen birim

SKT

Sürekli kelime torbası

UKSB

Uzun kısa süreli bellek

YSA

Yinelemeli sinir ağları

1. GİRİŞ

Çevrimiçi platformların giderek büyüyen popülaritesi ile birlikte geçtiğimiz yıllarda ve günümüzde bireylerin günlük internet kullanımında ciddi artışlar yaşanmıştır [1]. Bunun paralelinde gelişen akıllı telefonlarda, kameraların standart donanım olarak sunulması sayesinde multimedya kullanım ve paylaşım hacminde de önemli ölçüde artışlar gözlemlenmiştir. Örneğin sosyal medyada fotoğraf ve video paylaşma uygulaması olan Instagram'da [2], 2018 yılında günlük ortalama 300 milyon hikaye paylaşılırken [3], 2020 yılında bu sayı günlük 500 milyon hikaye paylaşımına çıkmıştır [4]. Ortaya çıkan ve her saniye milyonlarca yeni paylaşımın eklendiği bu devasa görsel veri koleksiyonu, beraberinde hem yönetimini, hem aranmasını hem de efektif şekilde işlenmesini gerektiren bir dizi zorluk ortaya çıkarmıştır. Ortaya çıkan zorlukları çözmek için ise akıllı algoritmaların tasarlanması ve uygulanması ihtiyacı belirlemiştir.

Aynı dönemde bilgisayarların gelişen hesaplama gücü ve artan kapasitesiyle birlikte derin öğrenme ve yapay zeka teknolojilerinde önemli ilerlemeler olmuştur. Bunun sonucunda yapay zekanın alt alanlarından olan bilgisayarlı görü ve doğal dil işleme alanlarında da daha iyi performanslar veren modeller geliştirilmiştir. Artan ihtiyaç ve gelişen teknoloji sonucunda bilgisayarlı görü ve doğal dil işleme alanlarının kesişim noktasında bulunan görüntülerden açıklama oluşturma diğer bir deyişle görüntülere başlık atama alanında başarılı çalışmalar yapılmıştır. Bu çalışmaların ortak noktasında yer alan amaç ise eldeki görüntüden insan yardımı olmaksızın görüntünün içeriğini otomatik olarak tanımlayan bir cümle veya ifade oluşturmaktır [5].

Görüntülere başlık atama problemi hem doğal dil işlemenin hem de bilgisayarlı görünün zorluklarını bünyesinde barındırmaktadır. Bunun yanında bir görselin içeriğini otomatik olarak tanımlama probleminin kendine has başka zorlukları da mevcuttur. Mesela bilgisayarlı görü alanının temel olarak ele aldığı nokta görüntü sınıflandırmadır ve sadece bu konuya odaklanmaktadır [6]. Dolayısıyla sadece bu alandaki problemleri çözmesi gerekmektedir. Fakat görüntüden açıklama oluştururken sadece görüntüdeki nesnelere değil, nesnelerin ortam ve birbirleri ile ilişkilerine ve aksiyonlarına da önem verilmesi gerekmektedir. Aynı zamanda oluşturulan açıklamanın anlamlı olması ve dil biçimsel olarak doğru olması gerekmektedir. Bu bağlamda görüntülere başlık atama konusu sahip

olduğu zorluk ve karmaşıklıktan dolayı hala üzerinde çalışmalar yapılan sıcak bir konu olarak değerlendirilmektedir.

Görüntülere başlık atama probleminin bütün zorluklarının yanında geliştirilen çözümler sadece dijital dünya problemlerinin çözümünde değil gerçek hayata dair farklı problemlerin çözümüne de sağlayacağı faydalar mevcuttur. İlk olarak görme engelli bireylerin hayatında yardımcı teknoloji olarak yer alarak görme engelli bireylerin hayatını kolaylaştırabilecektir [7]. Görme engelli bireylerin sahip olacağı bir cihaz görüntüyü önce metne çevirip, ardından metni sese çevirerek bireylere rehberlik edebilmesi sağlanabilecektir. Bu konu üzerine hali hazırda nVidia ve IBM firmaları partnerliğinde yürütülen çalışmalar mevcuttur [8-9]. Bunun yanında görüntülere başlık atama konusundaki yapılacak çalışmalar ile birlikte çocukların eğitiminden yaşlıların hayatlarını idame ettirmesine kadar toplumun çeşitli kesimlerine muazzam kolaylıklar sağlanabilecektir [10]. Bu yararların dışında da sistemin gelişmesiyle şu an öngörülemeyen daha farklı yararları olacağına dair hiç şüphe yoktur.

Görüntülere başlık atamanın iki ayağından biri olan bilgisayarlı görü; görüntüdeki yansımalar, dokunun belirsizliği gibi görüntüdeki gürültülerden etkilenebilmektedir ve bu gürültü görüntü işlemedeki temel zorluğa sebep olmaktadır. Görüntülere başlık atamanın ikinci ayağı olan doğal dil işleme alanında ise bilgisayarlı görü alanına benzer şekilde metin içindeki olası karakter hataları, gramer hataları gibi gürültüler metin işlemede zorluklara sebep olabilmektedir. Bunun yanı sıra günlük dilin bütünüyle gramer kurallarına uygun olması gibi bir durum olmadığı için doğal dilin kelime ve cümle yapısını anlamlandırmak ve sonuç çıkarmak doğal dil işlemeyi zorlayıcı kılmaktadır [11]. Bu durumların haricinde, daha çok yapısal olarak var olan bir zorluk ise cümle sınırını netleştirme olarak tanımlanabilecek metinleri daha küçük parçalara ayırarak değerlendirmedir. Burada metinde var olan yapılandırılmış verileri elimine etmeden, yapılandırılmamış veriyi anlamlı hale getirmek zorlu bir iş olarak tanımlanabilmektedir.

Doğal dil işlemenin kendi içindeki zorluklarının yanı sıra Türkçe dilinin kendine has zorlukları ve farklı özellikleri vardır. Türkçe sondan eklemeli bir dildir ve kökler sonradan eklenen ekler ile birlikte farklı formlarda bulunabilir [12]. Öyle ki Türkçe dilinde bir kök, eklenebilecek potansiyel üç farklı ekle birlikte onlarca farklı forma girebilmektedir. Bunun yanı sıra Türkçe, ünsüz uyumu özelliğine sahip olduğundan köke göre eklenen hecelerin

sertleşmesi veya yumuşaması mümkün olmaktadır. Türkçe dilinin bu zenginlikleri günlük hayatta ana dili Türkçe olan bireyler için avantajlar taşısa da doğal dil işleme açısından Türkçe dili diğer dillere göre daha zorlu bir dil olarak nitelendirilebilir.

Tez çalışmasının literatür katkısı

Literatürde görüntülere başlık atama alanında birçok çalışma vardır fakat bu çalışmalar incelendiğinde birçoğunun İngilizce dilinde kümelendiği gözlemlenmektedir [5]. Bunun yanında bizim kendi ana dilimiz olan Türkçe dili kullanılarak yapılan görüntülere başlık çalışma sayısı oldukça azdır. Bu sebeple literatüre kazandırılmış görüntülere başlık atama modellerinin başarımlar oranı Türkçe dilinde yeterli seviyede ölçülememiştir. Bu tez kapsamında yapılan çalışmalar şu şekilde özetlenebilir:

- Halihazırda geliştirilmiş görüntülere başlık atama modelleri Türkçe dilindeki veri kümesine uygulanmıştır ve modellerin başarımlar oranları ölçümlenmiştir.
- Türkçe dilinin sondan eklemeli yapısı göz önüne alınarak Türkçeye özgü olarak doğal dil işleme teknikleri uygulanmıştır.
- Görüntülere başlık atama konusundaki son çalışmalarda kullanılan modeller üzerine birtakım geliştirmeler yapılarak yeni modeller oluşturulmuştur ve yeni oluşturulan modellerin Türkçe dilindeki başarımlar oranları ölçümlenmiştir.
- Topluluk öğrenmesi tekniği ile birden fazla model beraber kullanılarak başarımlar oranları ölçümlemesi yapılmıştır.

Tez çalışmasının bölümleri

Bu tez çalışması toplam 6 bölümden oluşmaktadır. Bölüm 2’de görüntülere başlık atama konusu ile ilgili literatür taraması sunulmuştur. Bölüm 3’te evrişimli sinir ağları, yinelemeli sinir ağları ve dikkat mekanizması gibi görüntülere başlık atama modellerini anlamak için gerekli olan materyal ve metotlar sunulmuştur. Bölüm 4’te çalışma aşamaları, model gibi çalışmaya özgü önerilen yaklaşım sunulmuştur. Bölüm 5’te araştırma bulgularından bahsedilmiş ve Bölüm 6’da tezin kısa bir özeti sağlanarak gelecekteki olası araştırma konuları aktarılmıştır.



2. LİTERATÜR TARAMASI

Bu bölümde literatürde var olan görüntülere başlık atama modelleri incelenecektir. Görüntülere başlık atama çalışmalarında sinir ağları dominasyonuna kadar geçen süreçte iki temel yaklaşım hakim olmuştur. Bunların ilki şablon tabanlı (template based) yaklaşımdır. İkinci yaklaşım ise benzer görüntülere ait tanımlamaların getirilerek (retrieval based) , bu tanımlamaların mevcut görüntüye uyacak şekilde değiştirilmesi yaklaşımıdır. Bu iki yaklaşımın ortak özelliği özel isim gibi yalnızca mevcut görüntüyle alakalı bilgileri kaldırıp, tanımlamaları genelleme aşamasından geçirmeleridir [13].

2.1. Şablon Tabanlı Modeller

Şablon tabanlı modeller genelde daha önceden elle tanımlanmış cümle çerçevelerini kullanarak cümledeki nesne, özne, öznitelik açık alanları doldurma yaklaşımına sahiptir [14]. Yang ve diğerleri nesneleri, yüklemeleri, fiilleri ve edatları Gizli Markov Modeli [15] ile seçerek cümle şablonları doldurma çalışması yapmışlardır [16]. Mitchell ve diğerleri söz dizimsel olarak iyi oluşturulmuş cümle parçalarını türetme amacıyla ağaç ikame grameri (tree substitution grammar) kullanarak birleştirme çalışması yapmışlardır [17]. Benzer bir çalışmada Kuznetsova ve diğerleri, ağaç parçalarının eğitim kümesi içindeki açıklamalardan öğrenildiği ve bu parçaların yeni açıklamalar için test anında bir araya getirildiği bir çalışma yapmışlardır [18]. Gupta ve diğerleri görüntülerin etiketlenmesi olarak tanımlanan görüntü notlarından görüntü açıklaması oluşturma çalışması yapmışlardır [19]. Çalışmalarında basit bir açıklamanın nesne, öznitelik, özne, fiil gibi biz dizi temel unsurla tanımlanabileceğini göstermişlerdir. Örneğin çalışmalarında “beyaz bir kedi yeşil çimenin üstünde uyuyor” açıklaması iki nesne, “kedi” ve “çimen”, iki öznitelik, “beyaz” ve “yeşil”, iki özellik-nesne çifti “beyaz kedi” ve “yeşil çimen” gösterilmiştir. Çalışmalarında kullandıkları model, açıklamaları sadece nesneler ve özellikleri olmak üzere iki temel unsur kullanarak üretmektedir. Bu model ile anlamlı açıklamalar üretilmesine rağmen cümle yapısı elle tasarlanmaktadır. Daha karmaşık ve farklı yapılarda cümleler tasarlayabilmek için de açıklama yapısı bağımsız olmalıdır.

2.2. Eriřim Tabanlı Modeller

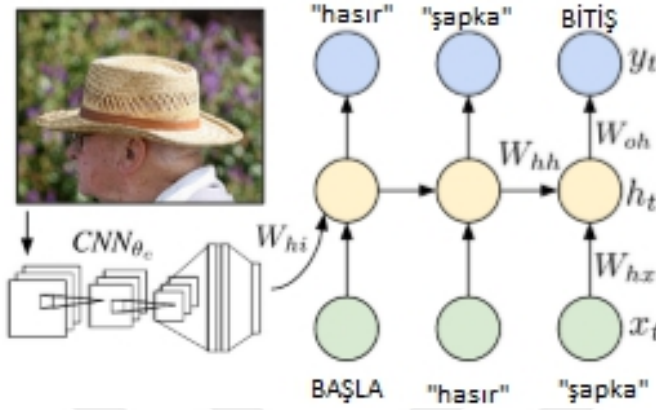
Eriřim tabanlı modeller, yeni tanımlanacak görüntünün açıklamasını oluşturmak için veri kümesindeki benzer görüntüleri alıp benzerliklerden yararlanarak yeni tanımlama oluřturmaya dayanmaktadır [20]. Ordonzer ve diğerkleri [21] geliřtirdikleri Im2Text modelinde yeni tanımlanacak görüntüyü temsil etmek için küçük görüntü tanımlayıcıları kullanılmıştır. Eriřim tabanlı modellerin ekseriyeti bu çalışmanın sonuçlarını referans çalışma olarak değerlendirmişlerdir. Farhadi ve diğerklerine göre görüntü ve açıklamaları arasındaki ilişkinin nesne, eylem ve sahne üçlüsü içinde temsil edilebilir [22]. Bu sebeple her görüntü ve açıklama bu alanda eşlenir. Bir cümle ile görüntü arasındaki yüksek puan, benzerliğin yüksek olduğuna delalettir. Bu modelin problemi benzersiz cümleler oluřturmaya olanak tanımamasıdır. Li ve diğerkleri de Farhadi ve arkadaşlarına benzer bir şekilde üçlü bir dizi ile temsil tabanlı bir model önermiştir. Fakat bu çalışmada n-gram kullanılarak aday cümleler elde edilmektedir [23]. Bu sebeple daha benzersiz cümleler oluřturabilme imkanı tanısa da, aday cümleler görüntünün açıklamalarının özgünlüğünü sınırlandırmaktadır.

Eriřim tabanlı modellerin handikaplarından birisi, oluřturulan başlıkların yakın görüntülerin tanımlamaları ile sınırlanmasıdır. Bu konuda çeřitli yaklaşımlar mevcuttur. Mason ve diğerklerinin [24] çalışmasında bu problemin çözümüne yönelik yalnızca metinsel bilgiler ele alınmıştır ve nihai tanımlama çıkarımsal özetleme teknikleri kullanılarak belirlenmiştir.

2.3. Çok Boyutlu Modeller

Hem şablon tabanlı hem de erişim tabanlı yaklaşımlar belirli bir noktaya kadar başarı sağlamış olsalar da dizayn olarak ağır yapıda olmaları ve metin oluřturma kabiliyeti bakımından kural bazlı esnek olmayan yapıda olmaları sebebiyle görüntüyü betimleme noktasında önceden belirmeyen nesneleri tanımlayamıyorlardır. Sinir ağlarının gelişimi ile birlikte mevcuttaki iki yaklaşım yerlerini sinir ağlarını kullanarak geliştirilen görüntülere başlık atama yaklaşımlarına bırakmıştır. Günümüzde literatüre giren görüntülere başlık atama çalışmalarının neredeyse tamamı sinir ağları kullanılarak geliştirilmektedir.

Karpathy'nin çalışmasında [25] görüntüler ve ilişkili tanımlamaların çift yönlü olarak sisteme alınması için çok boyutlu model kullanmayı önermiştir. Önerdikleri modelin temel mimarisi Şekil 2.1'de gösterilmektedir.



Şekil 2.1. Karpathy tarafından önerilen görüntülere başlık atama modeli [25]

Karpathy ImageNet üzerinde önceden eğitilmiş ve ince ayar yapılmış bir ESA kullanmıştır. Bununla birlikte başlık oluşturucusu olarak YSA kullanmıştır. Modelin eğitimi için her bir görüntünün özellikleri, YSA'nın bir önceki zamandaki gizli durumu (hidden state) ve bir önceki tahmin edilen kelime sistemin girdisini oluşturur. Sinir ağının çıktısı olarak sıradaki kelime tahmin edilir. Bu süreç cümle sonuna gelene kadar devam eder. Başlangıçta ise önceki zamanın gizli durumu olmadığından sistemin girdisi sıfır olur. Önceki kelime olmadığından daha önceden belirlenmiş -BAŞLA- kelimesi anahtar kelime olarak sistemin girdisi olur. Yeterli örnekle eğitim tamamlandıktan sonra YSA, kullanılan dili öğrenir ve ağırlık güncellemeleri ile bir önceki kelime ve görüntü özellikleri girdileri ile bir sonraki kelimeyi tahmin eder.

Karpathy'nin çalışmasında eğitim ve test sürecinden önce kelimeler ön işlemeden geçirilerek kelime vektörüne dönüştürülür. ESA'dan çıkarılan görüntü vektörü ile kelime vektörü nokta çarpımı yoluyla aynı vektör uzayında eşlenir. Bu işlem, Girshick ve diğerlerinin önerdiği [26] bölgesel evrişimli sinir ağı (regional CNN) aracılığıyla elde edilir. Bölgesel evrişimli sinir ağıyla görüntüdeki en üst 19 nesne parçası tespit edilir. Devamında çift yönlü YSA ile kelimeyi iki yönde çevreleyen bilgiler ve kelimeye karşılık gelen görüntü özellik vektörüne dayalı olarak her kelime, görüntü özellik vektörüyle aynı vektör uzayına eşlenir. Burada tahmin edilen cümle parçası ve görüntü özellik vektörü

çiftinin iyi bir şekilde ilişkilendirilmesi amacıyla görüntü ve cümle puanının en yüksek olması hedeflenmektedir.

Kiros ve diğerleri [27] görüntülerden başlık oluşturmak için görüntü - başlık çok boyutlu sinirsel dil modeli kullanmayı önermiştir. Çalışmalarında modifiye edilmiş log-çift doğrusal dil modeli [28], kullanılmıştır. Doğal dil işleme probleminde aşağıda gösterilen önceden oluşturulmuş $k_1, \dots, k_{t-1}$ kelimelerine koşullandırılmış bir k_t kelimesi oluşturma olasılığını tahmin etmek için bir dil modeli kullanılır.

$$P(k_t | k_1, \dots, k_{t-1}). \quad (2.1)$$

Kiros ve diğerleri bu dil modelini iki farklı yoldan görüntülere bağımlı hale getirmişlerdir. İlk olarak görüntü özelliklerini bir sonraki tahmin edilecek kelimeye ekleyerek bağımlı hale getirmişlerdir. İkinci olarak ise kelime temsil matrislerini, görüntü özellikleri aracılığıyla geçitleyerek bağımlı hale getirmişlerdir. Yani Kiros ve diğerlerinin çalışmasında bir k_t kelimesi üretme olasılığı aşağıdaki gibidir:

$$P(k_t | k_1, \dots, k_{t-1}, G). \quad (2.2)$$

Burada G görüntü özelliklerini ifade etmektedir.

Bu çalışmada görüntülerin özellikleri derin evrişimli sinir ağı ile çıkarılmıştır. Ortak görüntü-açıklama özellikleri, çok boyutlu sinir ağı modeli aracılığıyla öğrenilmiştir. Önerilen model ile önceden oluşturulan kelimeler ve görüntü özelliğine bağlı olarak kelime kelime görüntü başlığı oluşturulabilmektedir.

Google araştırmacıları olan Vinyals ve diğerleri tarafından oluşturulmuş [29] Sinirsel Görüntü Alt Yazısı Oluşturucu modeli, 2014 yılı MS COCO Başlık Atama Mücadelesi'nde ilk sırada yer almıştır ve bu çalışma görüntülere başlık atama alanında temel alınan, gelişmiş çalışmalardan biri olarak nitelendirilir. Çalışmada makine çevirisindeki son gelişmeleri görüntülere başlık atama alanı için uygulanmaktadır. Çalışmada kodlayıcı-kod çözücü mimarisi kullanılmıştır. Çalışmanın kodlayıcı kısmı için ESA ve kod çözücü kısmı için YSA kullanılmıştır. Model, bir girdi görüntüsü verildiğinde hedef kelime dizisi üretme

çıkarmak için ESA kullanır. Çıkarılan vektörlerin her biri görüntünün bir kısmına karşılık gelmektedir.

Modelin kod çözücü kısmı kelime seviyesinde çalışan UKSB ağıdır ve görüntünün ilgili kısmının dinamik bir temsili olan bir içerik vektörü oluşturur. Çalışmada içerik vektörünü hesaplayan iki farklı dikkat mekanizması uygulanmıştır. Birincisi sert dikkat mekanizmasıdır. Bu mekanizma vektördeki değerleri karşılık gelen konumun odaklanılacak doğru yer olma olasılığı olarak yorumlar. İkinci mekanizma yumuşak dikkat mekanizmasıdır ve tüm vektörlerin değerlerini bir araya getirerek konumun göreceli önemini verir. Bu çalışma dikkat mekanizmasının nesne olmayan bölgelere bile katılabileceğini göstermektedir. Çalışmadaki model insan sezgisine karşılık gelen hizalamaları bile öğrenebilir.

Çalışmada ImageNet üzerinde eğitilmiş bir ESA kullanılmıştır. Model eğitimi sırasında modelin Flickr8k [34] veri kümesi için RMSProp [35] optimizasyon algoritmasıyla en iyi sonucu verdiğini, Flickr30k [36] ve MSCOCO [37] veri kümelerinde ise Adam [38] algoritmasıyla en iyi sonucu verdiğini belirtmişlerdir. Eğitim sırasındaki performansın ve hızın aynı uzunluktaki cümleler ile mini gruplar oluşturulduğu zaman arttığı gözlemlenmiştir.

Yukarıdaki çalışmaların hepsi İngilizce dilindedir. Sayıca oldukça az da olsa Türkçe dili kullanılarak yapılan görüntülere başlık atama çalışmaları mevcuttur. Ünal ve diğerleri tarafından [39] geliştirilen çalışma Türkçe dilindeki ilk görüntülere başlık atama çalışmasıdır. Bu çalışma kapsamında öncelikle Türkçe görüntülere başlık atama veri kümesi oluşturulmuştur. Ünal ve diğerleri tarafından oluşturulan TasvirEt isimli görüntülere başlık atama veri kümesi Türkçe için geliştirilmiş ilk görüntülere başlık atama veri kümesidir ve halen bu alanda TasvirEt'den başka Türkçe veri kümesi bulunmamaktadır.

Çalışmada TasvirEt veri kümesi kullanılarak iki farklı görüntülere başlık atama yaklaşımı önerilmiştir. İlk yaklaşım en yakın görüntü başlığının transferidir. Bu kapsamda görüntünün vektörel görünümü ESA kullanılarak çıkarılmış ve bu görüntüye en yakın N^2 aday görüntü veri kümesinden getirilmiştir. Bu görüntülerden en yakın görüntünün başlığının transfer edilmesi ile test görüntüsünün başlığı elde edilmektedir. Çalışmadaki

diğer bir yaklaşım ise konsensus başlığın bulunması şeklinde tanımlanmaktadır. Önerilen yaklaşıma göre en yakın komşu görüntü kümesi bulunduğundan sonra bu görüntülere ait başlıklar köklerine ayrılmaktadır ve başlık kümesinden en merkezi başlık, test görüntüsüne en uygun başlık olarak değerlendirilmektedir.

Samet ve diğerlerinin çalışmasında [40] yeni bir veri kümesi oluşturmak veya literatürdeki tek Türkçe veri kümesi ile çalışmak yerine İngilizce veri kümelerinin çeviri yazılımı aracılığıyla Türkçeye çevrilerek elde edilen veri kümesi üzerine görüntülere başlık atama modeli eğitimi gerçekleştirilmiştir. Çalışmada kodlayıcı-kod çözücü tabanlı bir model kullanılmıştır. Modelin kodlayıcı tarafında ESA kullanılmıştır. Kod çözücü tarafında ise YSA kullanılmıştır. Önerilen modelde ESA olarak en son katmanı çıkarılmış VGG16 [41] ağı kullanılmıştır. YSA olarak ise UKSB kullanılmıştır. Çalışmada MSCOCO ve Flickr30k veri kümeleri Google Translate API [42] kullanılarak Türkçeye çevrilmiştir. Çalışmada optimizasyon yöntemi olarak Adam kullanılmıştır. Çalışmanın başarımlarını ölçmek için BLEU [43], METEOR [44], Rouge-L [45] ve CIDEr [46] yöntemleri kullanılmıştır.

Kuyu ve diğerlerinin çalışmasında [47] bir önceki çalışmada olduğu gibi İngilizce veri kümeleri çeviri yazılımı aracılığıyla Türkçeye çevrilerek çevirisi yapılmış veri kümeleri üzerine çalışılmıştır. Çalışmada Türkçenin sondan eklemeli yapısından dolayı kelimeler yerine alt sözcük öğeleri kullanan bir model önerilmiştir. Önerilen modelde eğitim kümesindeki sözcükler bayt çifti kodlaması (byte pair encoding) [48] kullanılarak alt sözcüklere ayrıştırılmaktadır ve sonuçlar alt sözcük sözlüğü kullanılarak elde edilmektedir.



3. MATERYAL VE METOTLAR

Bu bölümde sinir ağı hakkında gerekli teknik temel ve görüntülere başlık atama çalışmamızda kullanılan yaklaşımların teknik alt yapısı sağlanmaktadır.

3.1. Makine Öğrenimi

Makine öğrenimi çeşitli bilgisayar algoritmalarının, önceki deneyimlere dayanarak veriler arasındaki gizli bilgilerin çıkarılmasını sağlayan bir bilim dalıdır. Aynı zamanda makine öğrenimi hesaplama açısından maliyetli işlemlerin kolaylıkla yapılabilmesine olanak sağlamaktadır. Makine öğrenimi algoritmaları öğrenme yöntemlerine göre denetimli öğrenme, denetimsiz öğrenme, pekiştirmeli öğrenme, yarı denetimli öğrenme gibi sınıflara ayrılabilir.

3.1.1. Denetimli öğrenme

Denetimli öğrenme veri kümesindeki girdilerden istenilen çıktılar bilindiği yani sonucun zaten etiketlendiği veri kümeleri üzerine uygulanan bir öğrenme makine öğrenimi yöntemidir [49]. Denetimli öğrenmede eğitim kümesinin belirlenen algoritma yoluyla eğitimi ile girdi ve çıktılar arasında bir ilişki kurulur. Kurulan ilişki kullanılarak test zamanında veri kümesindeki girdinin karşılığı olacak çıktı tahmin edilir.

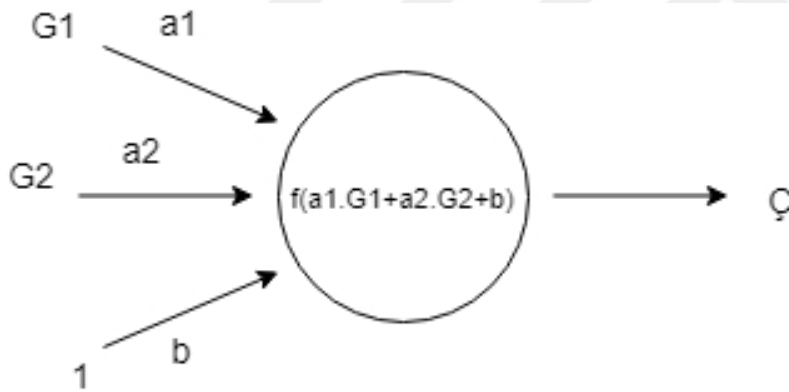
3.1.2. Denetimsiz öğrenme

Denetimsiz öğrenme veri kümesindeki girdilerden istenilen çıktılar bilinmediği sonucun etiketlenmediği veri kümeleri üzerine uygulanan bir öğrenme makine öğrenimi yöntemidir. Denetimsiz öğrenmede amaç veri kümesi içindeki ilişkinin ve yapıların öğrenilmesidir. Kümeleme, benzer girdileri alakalı kümelere ayıran bir teknik olarak denetimsiz öğrenmeye örnektir [11].

3.2. Yapay Sinir Ağları

II.Dünya Savaşı'ndan sonra ortaya çıkan yapay sinir ağları, insan beynindeki nöronların matematiksel modelinden esinlenilerek ortaya çıkmış bir teknolojidir [50]. Yapay sinir ağlarının en temel parçası olan ve lineer fonksiyonla ifade edilebilen düğüm veya nöron, 1958'de Frank Rosenblatt [51] tarafından oluşturulmuştur. Yapay sinir ağlarının önem kazanması, geri yayılım algoritması [52] ile birleştikten sonra olmuştur. Fakat o dönemde sinir ağları makinelerin işlem gücündeki zayıflık ve ayrıca veri kümelerinin eksikliği nedeniyle bugünkü popülerliğine ulaşamamıştır.

Nöron veya diğer adıyla düğüm bir sinir ağının temel hesaplama birimi olarak tanımlanır. Bir nöron diğer bazı nöronlardan veya harici bir kaynaktan girdi alır ve bir çıktıyı hesaplar [53]. Düğüm, girişlerinin ağırlıklı toplamına Şekil 3.1'de gösterilen bir f fonksiyonu uygular.



Şekil 3.1. Nöron yapısı

Yukarıdaki nöronda G_1 ve G_2 girdileri, a_1 ve a_2 bu girdilerle ilişkili ağırlıkları temsil etmektedir. Ayrıca b yanlılık (bias) olarak adlandırılan başka bir girdiyi temsil etmektedir. Yapay sinir ağlarında bütün girdiler ilişkili ağırlıklarıyla çarpılır, yanlılık değeri ile toplanır ve bir sonraki nöronun değerini belirlemek için doğrusal olmayan bir aktivasyon fonksiyonu uygulanır. Aktivasyon fonksiyonu uygulanarak nöronun çıktısının doğrusal olması engellenmiş olur. Bunun amacı gerçek dünya verilerinin neredeyse hepsi doğrusal olmadığı için nöronların doğrusal olmayan verileri öğrenmesidir.

Aktivasyon fonksiyonları bir sayı alır ve üzerinde sabit bir matematiksel işlem uygular.

Günümüzde sıkça kullanılan birkaç aktivasyon fonksiyonu şunlardır:

Sigmoid: Girdinin değerini 0 ile 1 arasında olacak şekilde sıkıştırır.

$$\sigma(x) = 1 / (1 + \exp(-x)) \quad (3.1)$$

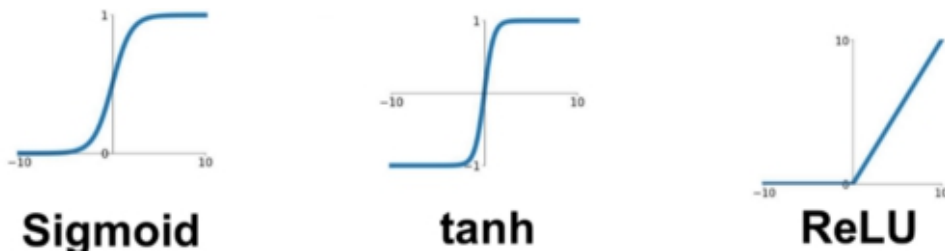
Tanh: Girdinin değerini -1 ile 1 arasında olacak şekilde sıkıştırır.

$$\tanh(x) = 2\sigma(2x) - 1 \quad (3.2)$$

ReLU: Doğrultulmuş Doğrusal Birim (Rectified Linear Unit) anlamına gelir. Girdinin değeri eğer negatif ise girdinin değerini sıfırla değiştirir.

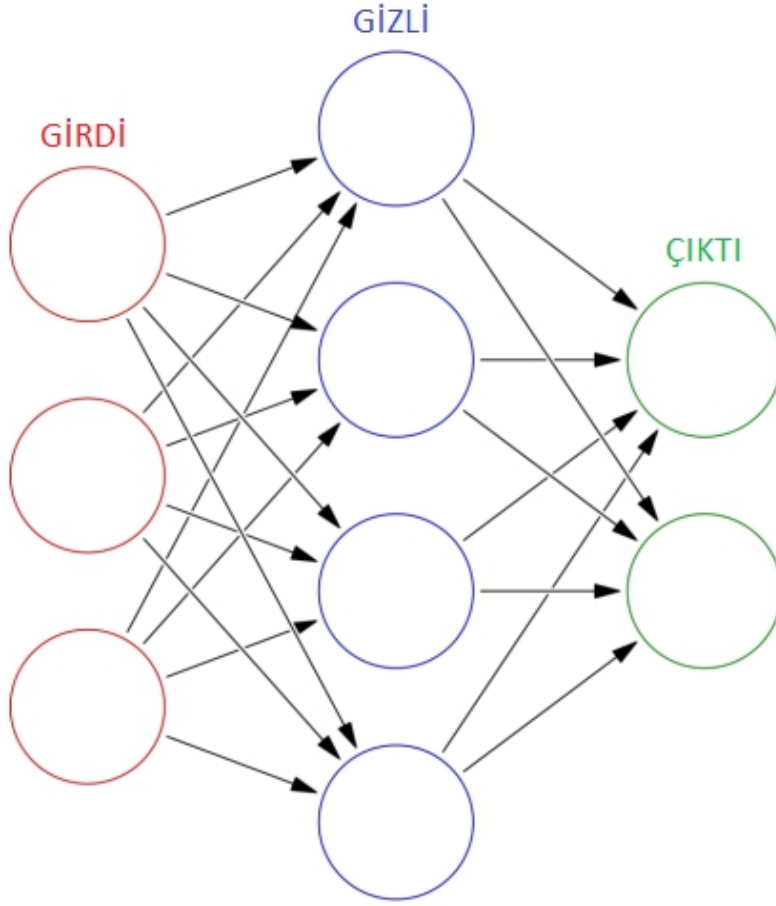
$$f(x) = \max(0, x) \quad (3.3)$$

Şekil 3.2’de yukarıdaki aktivasyon fonksiyonları gösterilmiştir.



Şekil 3.2. Aktivasyon fonksiyonları

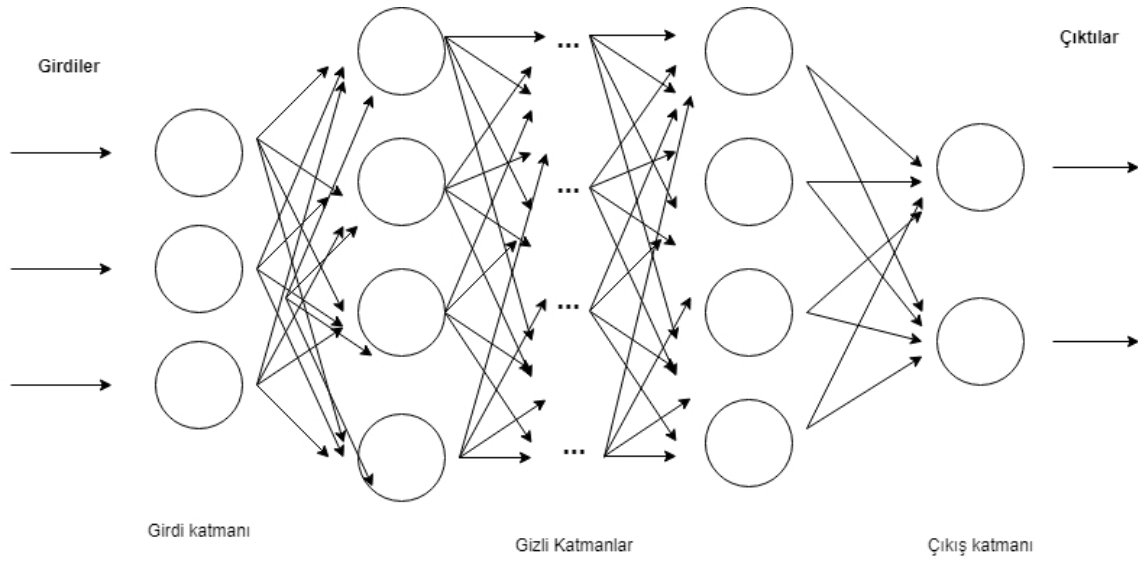
Şekil 3.3’de görülebilen temsili bir yapay sinir ağı giriş katmanı, çıkış katmanı ve gizli katmandan oluşur. Girdiler giriş katmanına verilir ve gizli katmanlardaki birçok geçişten sonra sonucu dış katmana döndürür. Yapay sinir ağlarının çıktığı tahmin etmek için belirli algoritması yoktur ve sürekli olarak girdi verilerinden öğrenen bir yapıdadır. Yapay sinir ağlarında nöronların arasında bulunan kenarlar üzerinden sinyaller nöronlar arasında aktarılır. Kenarların ağırlıkları vardır ve sürekli öğrenme ile kenarların ağırlıkları yeniden hesaplanarak modelin optimizasyonu sağlanır. Bu ağırlıklar, bir kenardan geçen sinyalin gücünü belirler.



Şekil 3.3. Yapay sinir ağı [54]

3.3. Derin Öğrenme

Makine öğrenimi alt alanı olarak tanımlanabilen derin öğrenme [55], temel olarak çok katmanlı yapay sinir ağı yapılarıdır. Derin öğrenme modelleri makine öğrenimi modellerinin aksine hem özellik çıkarımını hem de sınıflandırmayı kendi kendine yapmaktadır. Derin öğrenmedeki “derin” tabiri yapay sinir ağındaki katman sayısının büyüklüğünden gelmektedir [56]. Yani ağıdaki katman sayısı arttıkça ağı derinliği artmaktadır. Klasik yapay sinir ağlarında katman sayısı oldukça sınırlı iken, derin sinir ağları onlarca katmandan oluşabilmektedir. Bu kadar büyük yapıdaki bir ağı üzerinde işlem yapmak için yüksek işlem gücüne sahip cihazlara ihtiyaç duyulmaktadır. Bu noktada derin öğrenme modellerinin son dönemde gelişmesinin temel sebebi ihtiyaç duyulan işlem gücündeki artıştan dolayıdır. Derin öğrenme modellerini kullanabilmek için farklı yapılar mevcuttur. Günümüzde yapılan çalışmaların ekseriyeti bu yapılardan Keras [57], TensorFlow [58], Torch [59] ve Caffé [60]’yi kullanmaktadır. Şekil 3.4’te örnek bir derin sinir ağı yapısı gösterilmektedir.

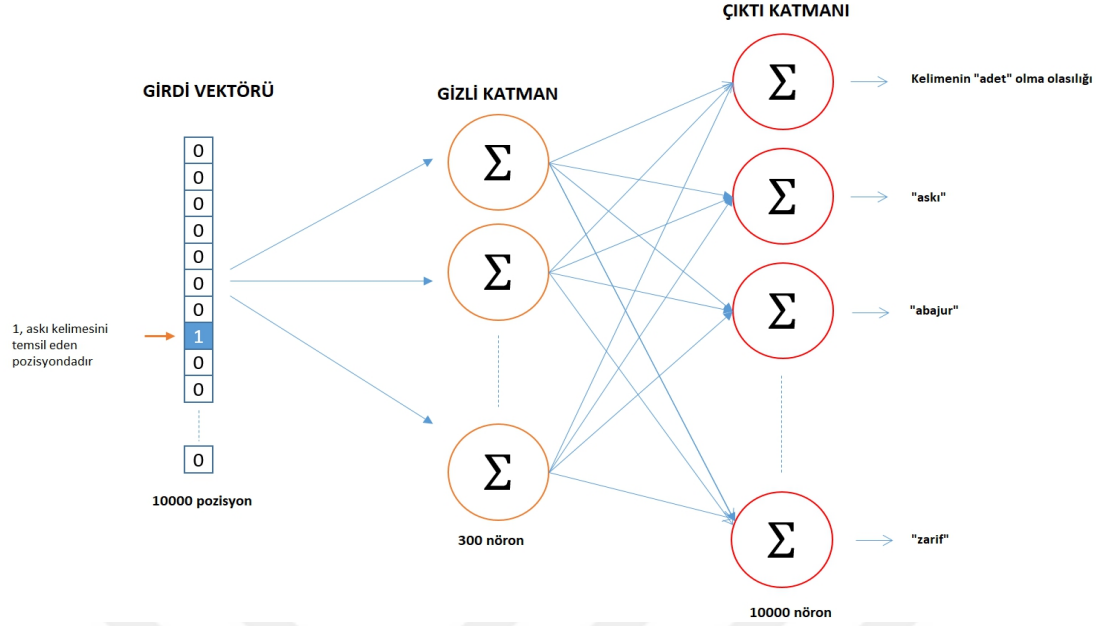


Şekil 3.4. Derin sinir ağları temsili gösterimi

3.4. Kelime Gömmeleri

Kelime gömmeleri (word embeddings) anlamsal olarak birbirine yakın kelimelerin benzer şekilde temsil edilmesine olanak tanıyan bir kelime temsil yöntemidir [61]. Kelime gömmelerini çıkarmak için farklı yöntemler mevcuttur. Bunlardan en popüler ve en yaygın kullanıma sahip olanı, Google'da çalışan Mikolov ve diğerleri tarafından 2013 yılında geliştirilen Word2vec'tir [62]. Word2vec kelimeleri vektör uzayında matematiksel olarak ifade etmeye çalışan denetimsiz ve iki katmanlı bir sinir ağı modelidir [63].

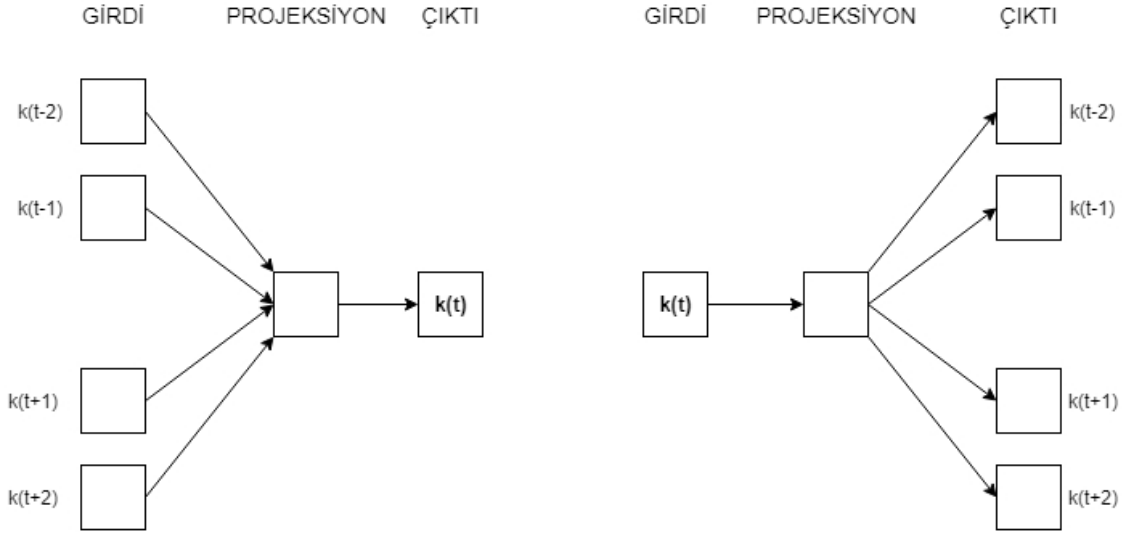
Mikolov ve diğerleri Word2vec çalışmalarıyla çok büyük boyutlu veri kümeleri üzerinden doğruluğu yüksek kelime vektörleri elde etmeyi amaçlamışlardır. Çalışma temel olarak, benzer kelimelerin vektör uzayında kosinüs uzaklığı hesaplandığında birbirine yakın konumda bulunması ve kelimelerin birden fazla benzerlik derecesine sahip olabileceği ilkesine dayanmaktadır [64]. Word2vec temel olarak giriş katmanı, gizli katman ve çıkış katmanı olmak üzere üç katmandan oluşan yapay bir sinir ağı yapısıdır. Bu ağda gizli katmandaki nöron sayısı kelimenin kaç boyutlu vektör ile temsil edildiğini gösterir. Ağın girdisi, veri kümesindeki kelimelerden oluşan sözlükteki her bir kelimeyi temsil eden bir sıfır vektörleridir (one-hot vector). Ağın eğitimi sırasında stokastik gradyan inişi ve geri yayılımı kullanılmıştır. [62]. Şekil 3.5'te Word2vec mimarisi gösterilmektedir.



Şekil 3.5. Word2vec mimarisi [65]

Kelimelerin vektörel temsilini çıkarmak için kullanılan iki ana Word2vec mimarisi vardır. Bunlar Sürekli Kelime Torbası modeli (Continuous Bag Of Words) ve skip-gram modelidir. Algoritmik olarak, bu modeller birbirine benzerdir ve her ikisi de kelimeleri hedef değişkenle eşleştiren sinir ağlarıdır [66]. Her iki modelde kelimelerin vektörel olarak tanımlamasını sağlayan sinir ağındaki ağırlıkları öğrenir.

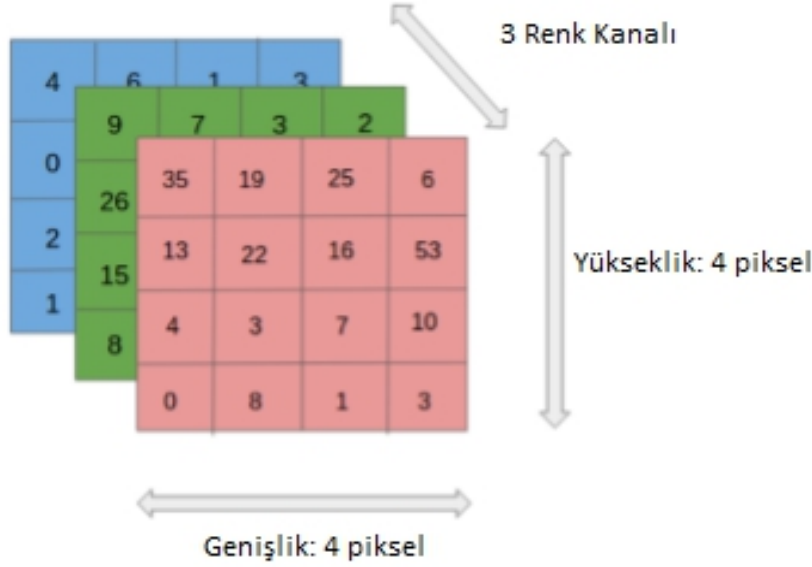
SKT modeli, mevcut kelimeyi, çevresindeki kelimelerden oluşan bir pencereden tahmin eder. Skip-gram modeli, SKT modeline göre ters mantıkta çalışır. Yani mevcut sözcüğü kullanarak çevreleyen penceredeki kelimeleri tahmin eder. Örneğin “bir adam çimenlerin üstündeki köpeğine bakıyor” cümlesinde SKT: “çimlerin” kelimesini tahmin etmek için pencere boyutu iki iken: “bir”, “adam”, “üstündeki”, “köpeğine” kelimelerini girdi olarak alırken; skip-gram tam tersini şekilde “çimlerin” kelimesinden pencere boyutu iki iken “bir”, “adam”, “üstündeki”, “köpeğine” kelimelerini tahmin eder. SKT skip-gram'a göre daha hızlıdır ancak skip-gram sık geçmeyen kelimeler için daha başarılı sonuç verir. bu sebeple küçük veri setlerinde daha iyi performansa sahiptir. Şekil 3.6’da SKT ve skip-gram yöntemleri gösterilmektedir.



Şekil 3.6. SKT (solda) ve skip-gram(sağda) modelleri

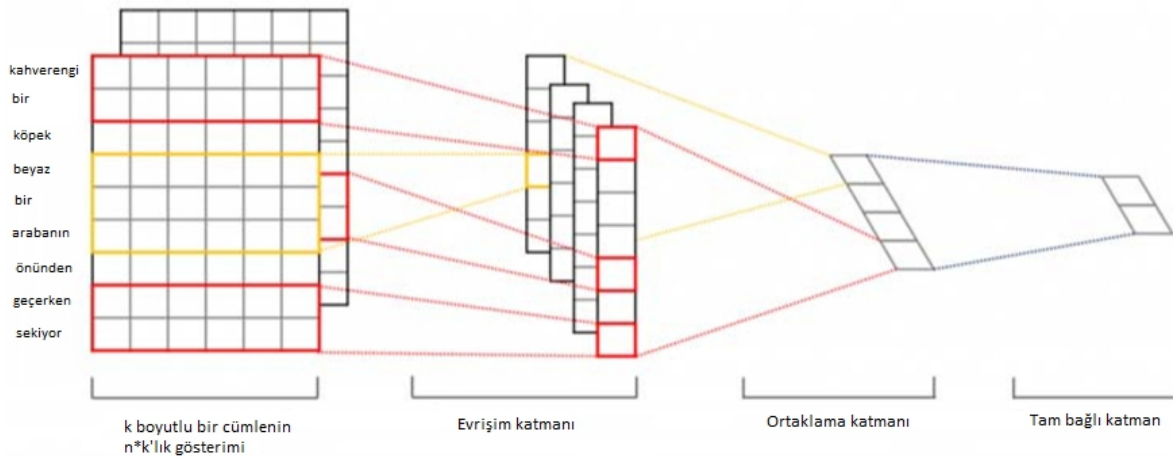
3.5. Evrişimli Sinir Ağları

Evrişimli sinir ağları, LeCun ve diğerleri tarafından 1989 yılında önerilen ileri beslemeli bir yapay sinir ağı ve derin öğrenme mimarisidir [67]. İnsanlar ve makineler arasındaki algılama farkını kapatmaya yardımcı olmak için canlıların görsel algı mekanizmalarından ilham alınarak tasarlanan ESA'nın temel amacı, resimler içinde bilgiyi anlamak, derin öğrenme yöntemleriyle yeni resim oluşturmak gibi resimler üzerine derin öğrenme çalışmaları yapmak için resim içindeki karmaşık özellikleri anlamaktır. ESA, bir görüntü üzerine çalışabilecek, görüntüdeki nesneleri algılayacak ve birbirini ayırt edebilecek şekilde tasarlanmıştır [68]. Fakat ESA ile sadece görüntüler üzerine çalışmalar yapılmaz; videolar, ses verileri, metinler gibi çok çeşitli konular üzerine ESA ile çalışmalar yapılmıştır ve yapılmaya aktif bir şekilde devam edilmektedir.



Şekil 3.7. 3 renk kanallı 4x4'lük bir görüntünün gösterimi [69]

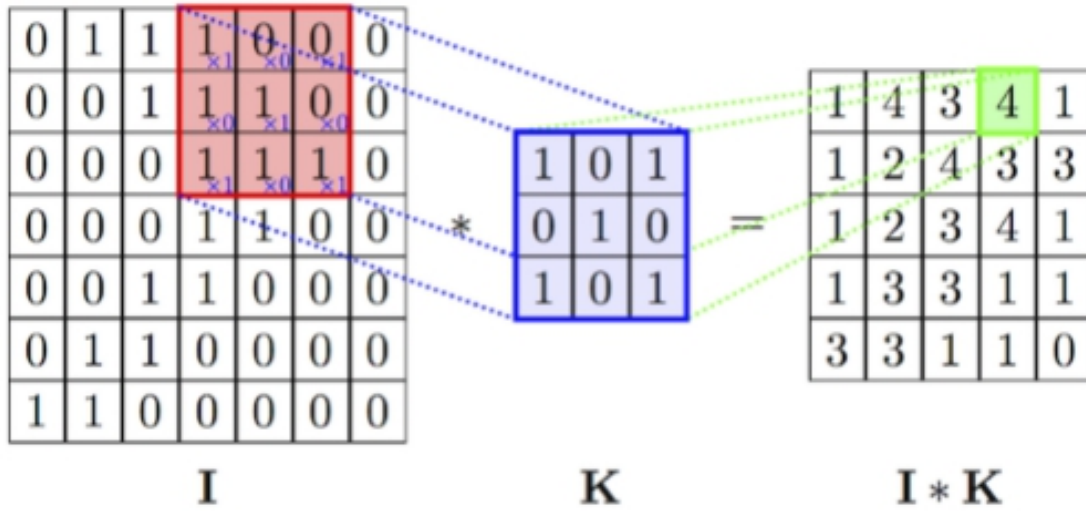
Bu çalışmaların çoğunda girdi çok boyutlu bir dizidir. Örneğin Şekil 3.7'deki gibi 256x256 piksellik bir renkli görüntü, 3 renk kanalına sahiptir (kırmızı, yeşil ve mavi) ve 256x256x3 boyutludur. Girdi boyutunun yüksekliği sebebiyle basit sinir ağlarındaki gibi bağlı katmanları kullanmanın çok fayda sağlamadığı deneyimlenmiştir. Bu sebeple girdi boyutunun yüksek olduğu görüntü, video ve ses gibi veri kümeleri üzerine yapılan çalışmalarda, girdinin mekansal düzeninin farkında olabilen ve anlamlı parametre paylaşımları yapabilen ESA'nın daha başarılı olduğu gözlemlenmiştir. ESA temel olarak üç katmandan oluşur. Bunlar giriş katmanı, gizli katman ve çıktı katmanıdır. Ancak gizli katmanda basit sinir ağlarının aksine, birbirini takip eden evrişim ve ortalama katmanları vardır. Şekil 3.8'de, cümle sınıflandırma için basit bir ESA mimarisi gösterilmektedir.



Şekil 3.8. Örnek evrişimli sinir ağı mimarisi [70]

3.5.1. Evrişim katmanı

Evrişim katmanı ESA'nın en önemli kısmıdır. Evrişim katmanında önceki katmandaki giriş verileri dönüştürülür. Yani, giriş verilerinden girdinin özellik haritası (feature map) çıkarılır. Bunu yapmak için girdiye farklı boyut ve ağırlıklarda filtreler, matris çarpımı yöntemi ile uygulanır. Bunu işlem Şekil 3.9'da gösterildiği gibi her bir seferde, filtre bir adım hareket ettirilerek gerçekleştirilir. Şekil 3.9'da, I görüntünün piksellere dönüştürülmüş halini, K ise uygulanan filtreyi temsil etmektedir.

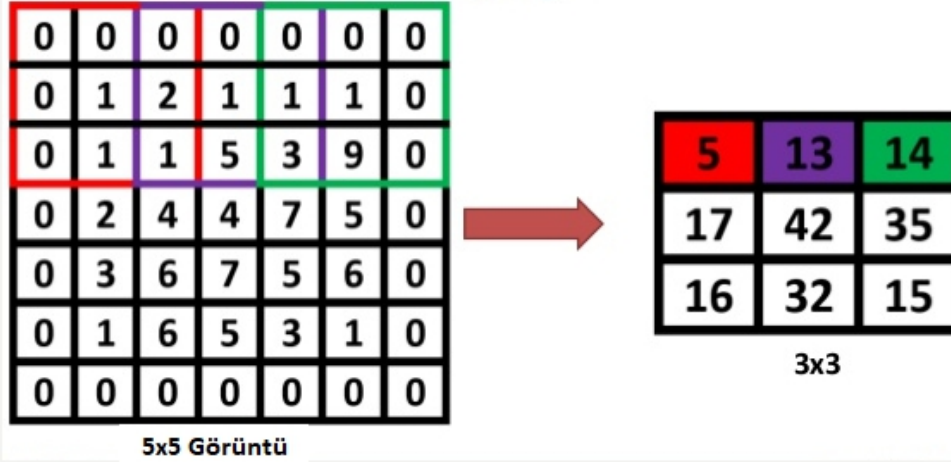


Şekil 3.9. Evrişim katmanı uygulaması [71]

Evrişim katmanı farklı ağırlık vektörlerine sahip birden çok özellik haritasından oluşur. Böylece girdinin her konumundan girdiye ait birden çok yüksek seviyeli ve düşük seviyeli özellik elde edilebilir [67].

3.5.2. Kaydırma adımı

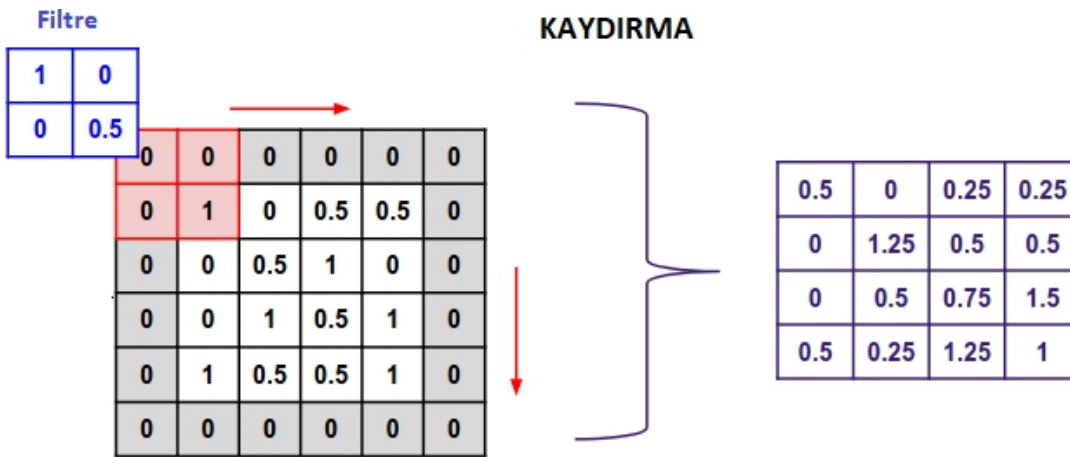
Kaydırma adımı (stride), filtrelerin giriş verileri üzerinde kaç birim hareket ettiğini ifade etmektedir. Şekil 3.10, 7x7 boyutundaki bir veriye 3x3 boyutundaki bir filtrenin uygulanmasını göstermektedir. Şekil 3.10'daki örnekte kaydırma adımı iki olarak belirlenmiştir. Filtre, kaydırma adımı iki olarak belirlendiğinde x ve y yönlerinde iki birim hareket eder.



Şekil 3.10. Kaydırma adımı (stride) = 2 ile evrişim uygulaması [72]

3.5.3. Sıfır ekleme

Uygulanan filtre ile giriş verisi arasındaki boyut farkına göre girdi ile çıktı arasındaki boyut farkını azaltmak için girdi matrisine sıfır ekleme (padding) uygulaması yapılabilmektedir. Böylece sonraki katmanlara aktarılan özellik sayısı artırılabilir. Bu işlemi sağlayabilmek için giriş verisinin çevresine ekstra pikseller ile sıfır eklenir. Şekil 3.11’de bir kere sıfır eklenmiş girdi matrisine filtre uygulaması örneği gösterilmektedir.

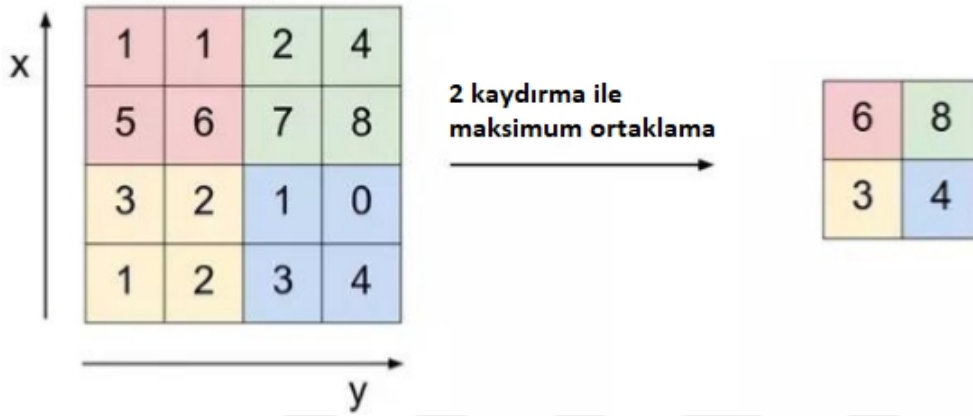


Şekil 3.11. Sıfır ekleme (padding) = 2 ile evrişim uygulaması [73]

3.5.4. Ortaklama katmanı

Ortaklama (pooling) katmanında, maksimum ortaklama, minimum ortaklama ve ortalama ortaklama gibi farklı yöntemler uygulanarak, seçilen yöntemle göre evrişim katmanının

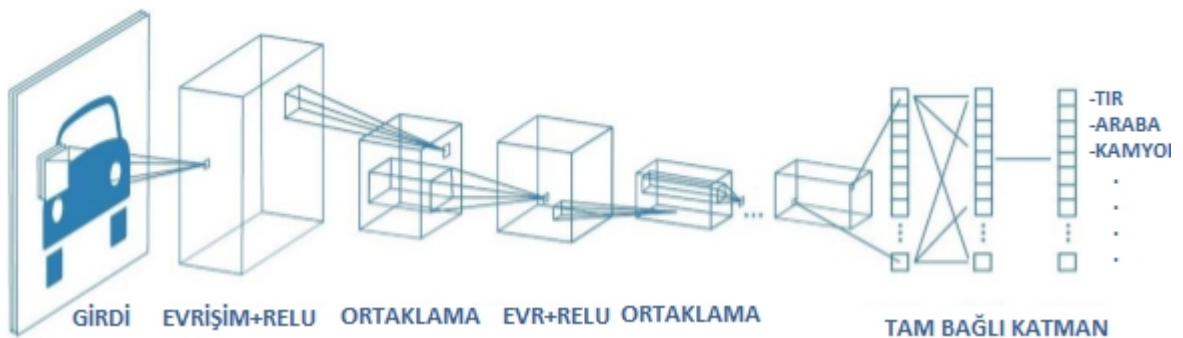
çıktıları üzerinden alt örnekleme yapılır. Ortaklama katmanındaki amaç baskın özellikleri kaybetmeden özellik vektörünün boyutunu küçültmektir. Maksimum ortaklamada seçilen boyut aralığındaki maksimum değer döndürülür, ortalama ortaklamada ise seçilen boyut aralığındaki tüm değerlerin ortalaması döndürülür. Maksimum ortaklama gürültülü verilerde daha iyi çalıştığı için ve gürültüyü ortadan kaldırdığı için daha sık kullanılabilmektedir [68]. Şekil 3.12’de maksimum ortaklama örneği gösterilmektedir.



Şekil 3.12. Ortaklama katmanı örneği [74]

3.5.5. Çıktı katmanı

ESA’daki son katman çıktı katmanıdır. Özellikleri çıkarılmış girdinin ortaklama katmanındaki çıktısı düzeltilirdikten sonra tam bağlı sinir ağına verilir. Tam bağlı sinir ağında özellikler arasındaki doğrusal olmayan ilişkiler öğrenilir ve sonucunda softmax sınıflandırması kullanılarak sınıflandırma yapılır. Softmax sınıflandırması ile her bir girdinin hangi sınıfa ait olduğunun olasılık değerleri hesaplanır. Şekil 3.13’te örnek bir ESA mimarisindeki bütün veri akışları gösterilmektedir.



Şekil 3.13. Evrişimli sinir ağı mimarisi [69]

Bu kısımda mevcut görüntülere başlık modelleri ve bu tez çalışması dahilinde yeni oluşturulan modellerin işleyişinden bahsedilecektir. Görüntü özelliklerinin çıkarılması ve dil modeli aşamaları her modelde kullanılırken, dikkat mekanizması sadece dikkat mekanizması ile çalışan modellerde kullanılmaktadır.

3.6. Görüntü Özelliklerinin Çıkarılması

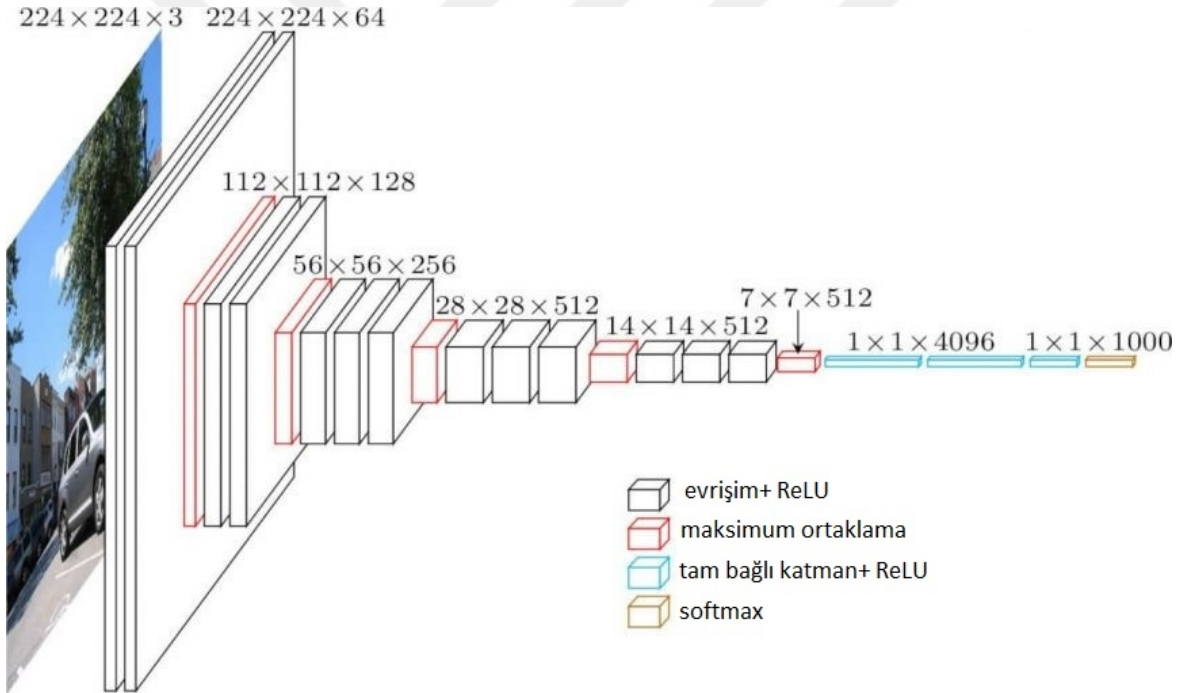
Görüntü özelliklerinin çıkarılması aşamasında ESA mimarileri kullanılmıştır. Bu tez çalışmasında kullanılan ESA mimarileri:

- VGG16
- VGG19
- ResNet
- ResNet v2
- Inception v3
- InceptionResNetv2
- Xception
- EfficientNet
- MobileNet'dir.

Kullanılan mimarilerin eğitilmesi hem eldeki veri miktarının azlığı hem de mevcut bilgisayarın işlem gücünün sınırı nedeniyle çok zordur. Bununla birlikte kullanılan ESA mimarilerinin hepsi hali hazırda ImageNet [75] gibi binlerce kategoride milyonlarca görüntüden oluşan bir görüntü veri kümesi üstünde eğitilmiştir. Bu noktada literatürdeki hemen hemen bütün çalışmalar gibi transfer öğrenmesi yardımıyla mevcuttaki eğitilmiş mimarilerin ağırlıkları kullanılarak görüntülerin özellikleri çıkarılmıştır. Bunun yanında bizim amacımız görüntülerin özellik vektörünü kullanarak görüntülere başlık atama problemini çözmek olduğu için kullanılan mimarilerdeki son iki katman çıkarılmış ve mimarilerin çıktısı olarak bir tahmin dönmesi yerine görüntülerin özellik vektörünü çıktı olarak dönmesi sağlanmıştır.

3.6.1. VGG16 mimarisi

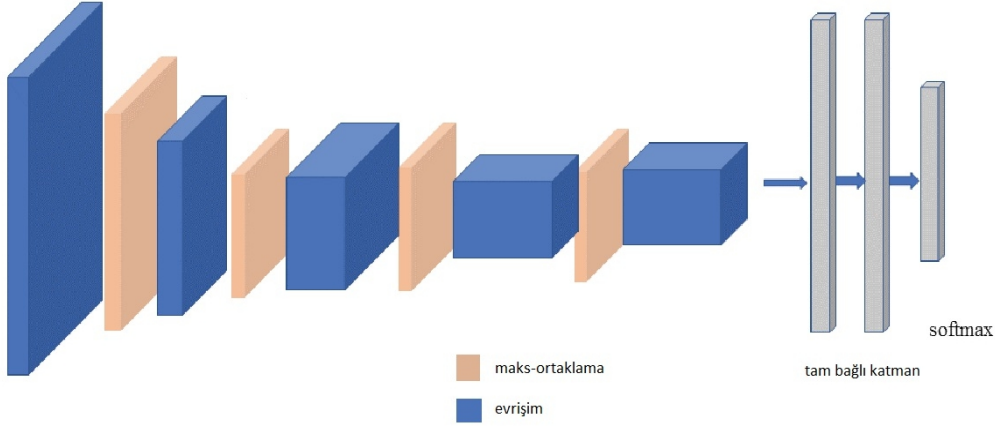
Simonyan ve Zisserman tarafından [41] önerilen VGG16 mimarisi, 2014'teki ILSVRC yarışmasında [76] en iyi 5 modelden biri olmuştur. VGG16, parametre sayısını kısararak eğitim zamanını azaltmak için tasarlanmıştır. Kendinden bir önceki en başarılı ESA mimarisi olan AlexNet'den [77] farklı olarak 3x3'lük filtre kullanmıştır ve VGG16 mimarisinde arka arkaya 2 veya 3 evrişim katmanı bulunmaktadır. VGG16 mimarisinde toplam 13 evrişim katmanı ve 5 ortaklama/ havuzlama katmanı bulunmaktadır. VGG16 mimarisinin en son aşamasında 4096 nörondan oluşan iki tane tam bağlı katman vardır ve bu katmanların sonunda hedef sınıf boyutunda (1000 nöron) içeren softmax katmanı bulunmaktadır [78]. Şekil 3.14'te VGG16 mimarisi gösterilmektedir.



Şekil 3.14. VGG16 mimarisi [79]

3.6.2. VGG19 mimarisi

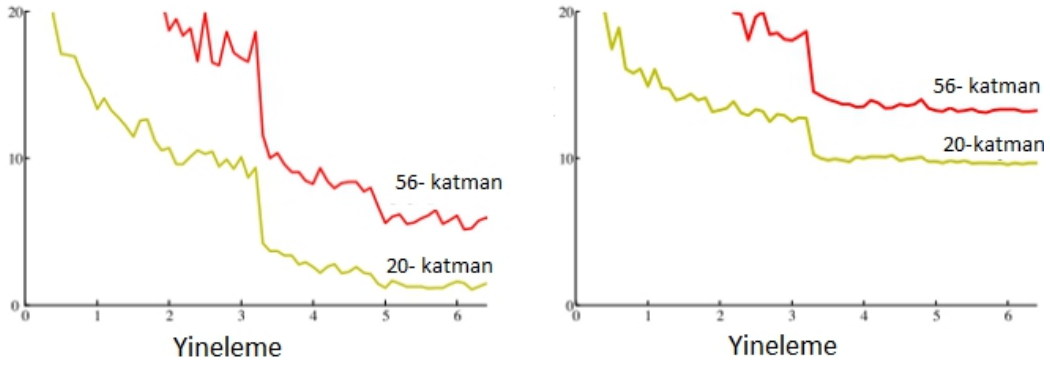
VGG16 mimarisi ile benzer bir mimariye sahip olan VGG19 mimarisinde [41] 19 tane katman bulunmaktadır. Şekil 3.15'de VGG19 mimarisi gösterilmektedir.



Şekil 3.15. VGG19 mimarisi [80]

3.6.3. ResNet mimarisi

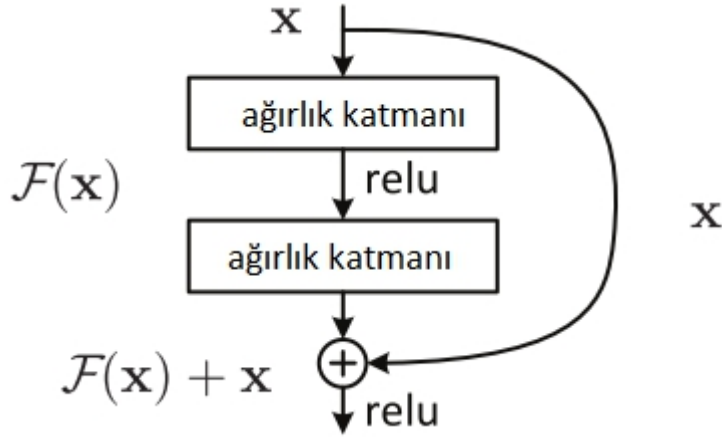
ESA mimarileri arasında ResNet [81], getirdiği yenilikle önemli bir konumda yer almıştır. ResNet'e kadar geliştirilen mimarilerde temel yaklaşım katman sayısını arttırarak derinliği arttırmak yönündedir. Fakat ağ derinliğini arttırmak bir noktadan sonra gradyan yok olması probleminden dolayı işe yaramamaktadır. Hatta birçok çalışmada ağın derinliğini çok fazla artırmanın başarımlar oranını azalttığı gözlemlenmiştir. Örnek olarak Şekil 3.16'da 20 ve 56 katmanlı iki ağın hata oranları gösterilmektedir.



Şekil 3.16. 20 ve 56 boyutlu iki sinir ağının hata oranları. Eğitim hatası (solda), test hatası (sağ) [82]

ResNet mimarisinin ana fikri, Şekil 3.17'de gösterildiği gibi artık bloğun (residual block) bir veya daha fazla katmanı atlayarak artık değerleri sonraki katmanlara aktarımını sağlamaktır. Bu sayede ResNet, önceki derin öğrenme mimarileri gibi düz bir akış içinde

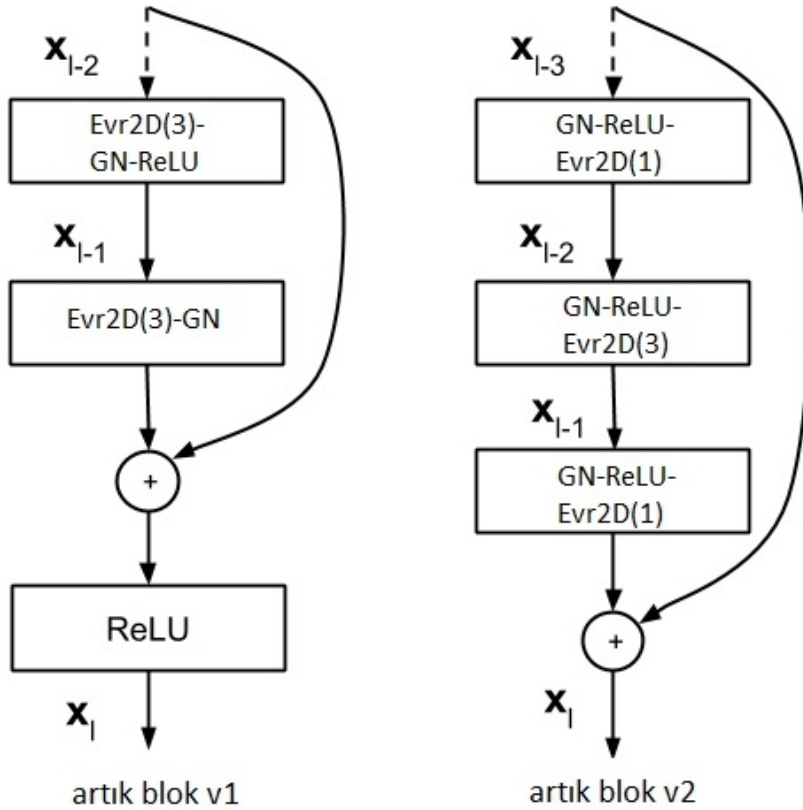
ilerlememektedir ve artık değerlerin aktarılması sayesinde ResNet mimarisinde gradyanların yok olması problemi çözülmektedir.



Şekil 3.17. Artık blok örneği [82]

3.6.4. ResNet v2 mimarisi

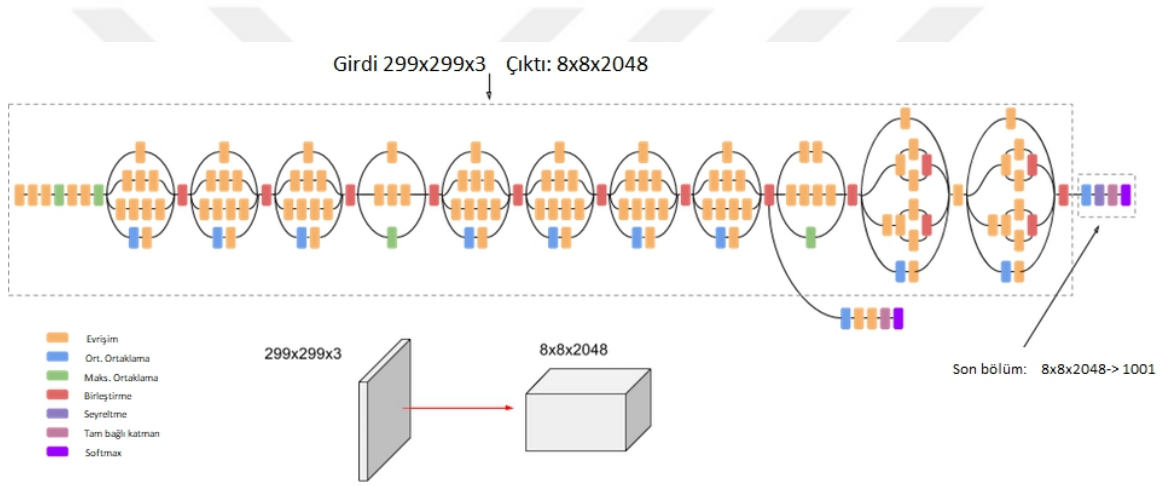
ResNet mimarisinin gelişmiş bir versiyonu olan ResNet v2’de temel iyileştirme Şekil 3.18’de gösterildiği gibi artık bloktaki katmanların düzenlenmesi üzerine yapılmıştır [83].



Şekil 3.18. ResNet v1 ve ResNet v2 artık blokları [84]

3.6.5. Inception v3 mimarisi

ILSVRC 2015 [77] yarışmasında en başarılı seçilen, Google araştırmacıları tarafından geliştirilen Inception mimarisi [85], ismini Inception filminden almaktadır. Inception mimarisi temel olarak farklı farklı Inception modüllerinden oluşmaktadır. Her bir Inception modülü ise farklı boyutlardaki evrişim ve ortaklama katmanlarından oluşmaktadır [85]. Bu tez çalışmasında kullanılan Inception v3 ise aynı araştırmacılar tarafından hazırlanmış Inception mimarisinin gelişmiş bir versiyonudur. Inception v3'te önceki versiyondan farklı olarak ağı daha düzenli hale getirmeye yarayan grup normalizasyon katmanı ve tam bağlı katman eklenmiştir. Şekil 3.19'da Inception v3 mimarisi gösterilmektedir.

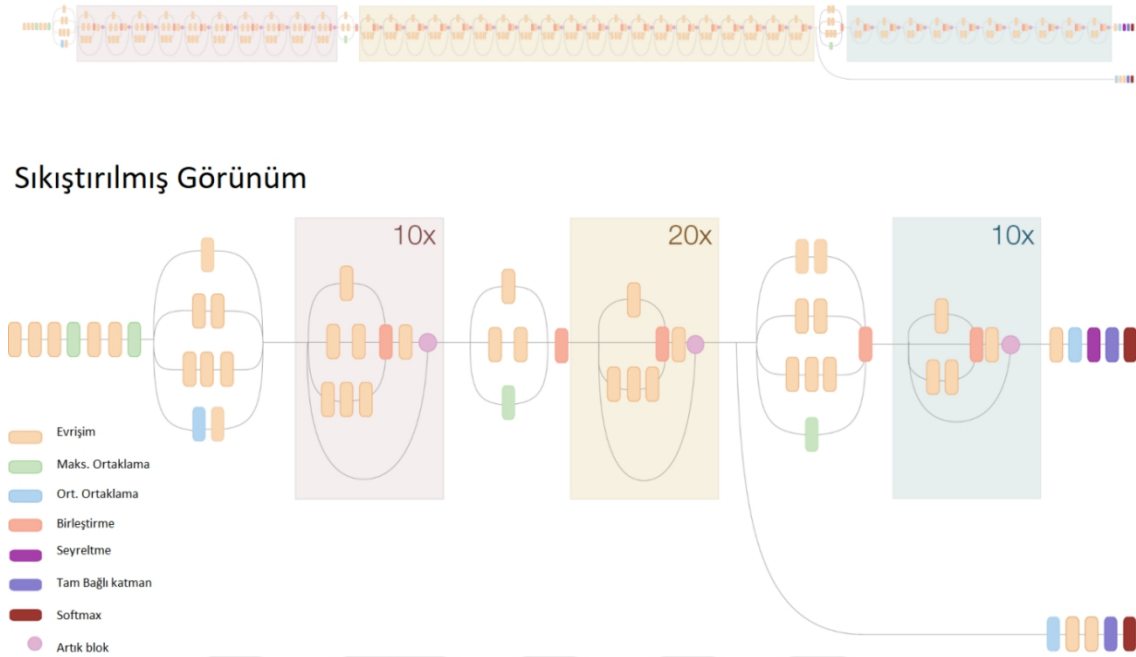


Şekil 3.19. Inception v3 mimarisi [85]

3.6.6. InceptionResNet v2 mimarisi

InceptionResNet v2 [86] mimarisi, Inception mimarisiyle ResNet mimarisinin birleşiminden oluşmuştur ve performansı artırmak hedeflenmiştir. InceptionResNet v2 mimarisi Inception v3 mimarisinden daha derindir ve bu sebeple hesaplama maliyeti daha büyüktür. Fakat doğruluk oranı daha yüksektir. Şekil 3.20'de InceptionResNet v2 mimarisi gösterilmektedir.

Inception Resnet V2



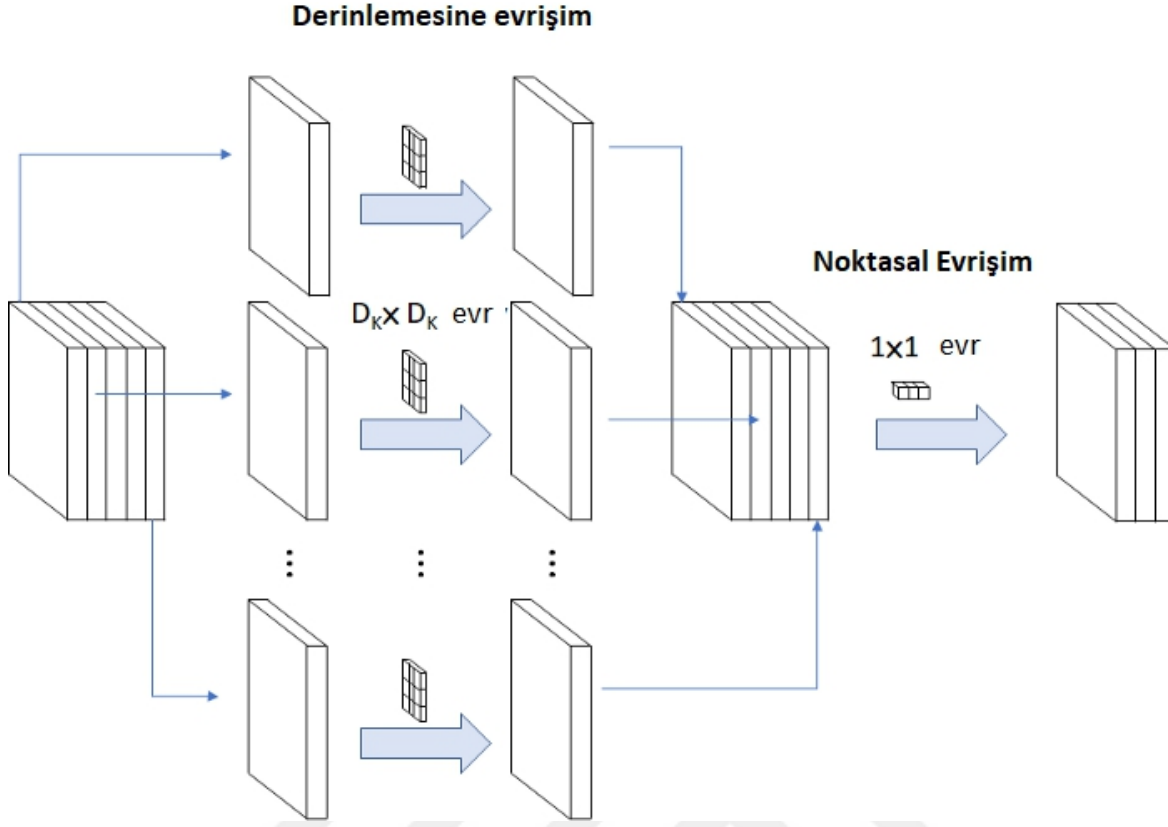
Şekil 3.20. InceptionResNet v2 mimarisi [87]

3.6.7. Xception mimarisi

Google araştırmacısı Chollet tarafından geliştirilen Xception mimarisi [88] Inception mimarisinin daha ileri versiyonu olarak ortaya çıkmıştır ve Inception v3'ten daha iyi sonuçlar elde etmiştir. Xception mimarisinde standart Inception modülleri, derinlemesine ayrılabilir evrişimler ile değiştirilmiştir [88].

3.6.8. MobileNet mimarisi

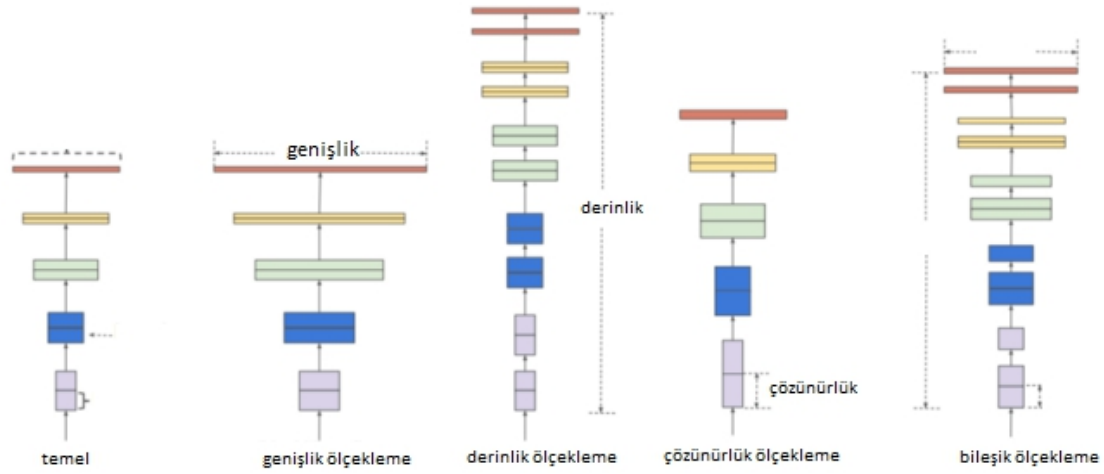
MobileNet [89], Google araştırmacıları tarafından mobil uygulamalar için geliştirilmiş bir ESA mimarisidir. MobileNet mimarisinde derinlemesine ayrılabilir sinir ağları kullanılmaktadır. Klasik ESA ağlarına göre parametre sayısı daha azdır. Böylece işlem maliyeti azalmaktadır. Şekil 3.21'de MobileNet mimarisi gösterilmektedir.



řekil 3.21. MobileNet mimarisi [90]

3.6.9. EfficientNet mimarisi

Bu alıřma dahilinde en son kullanılan ESA mimarisi EfficientNet'dir [91]. EfficientNet mimarisi kendinden nce geliřtirilen mimarilerin aksine sadece bařarım oranını artırma odaklı deęil; aynı zamanda daha efektif bir mimari sunmayı amalamıřtır. nk daha efektif bir mimarinin aynı zamanda modelin doęruluk yzdesini artırdıęı gzlemlenmiřtir. Bu baęlamda Efficient mimarisi geliřtirilirken mimarinin efektiflięini etkileyen derinlik, geniřlik ve znrlk boyutlarında leklendirme alıřması yapılmıřtır. EfficientNet mimarisinde leklendirme řekil 3.22'de gsterilmektedir.

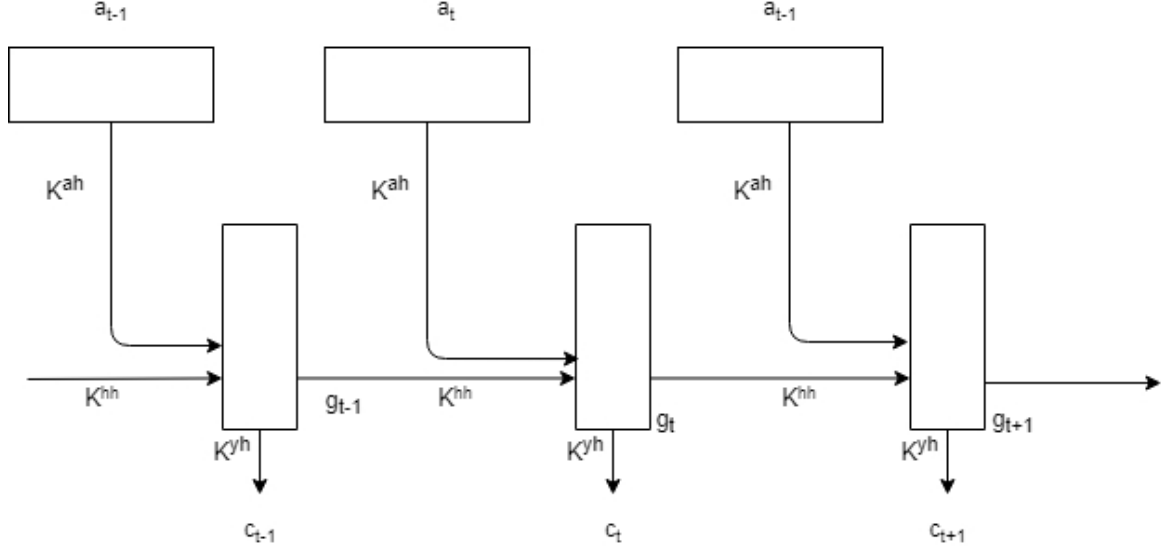


Şekil 3.22. Model ölçeklendirme [91]

3.7. Yinelemeli Sinir Ağları

Yinelemeli sinir ağları ileri beslemeli sinir ağlarının değiştirilmiş bir versiyonudur ve sıralı bilgi kullanır [92]. Geleneksel yapay sinir ağlarında her bir t zamanında sadece o anki mevcut girdi sınıflandırmak için kabul edilir. Önceki girdilerin sonuçları veya sonraki girdiler hesaba katılmaz, sadece o anki ağa beslenen girdi değerlendirilir. Fakat gerçek hayattaki uygulamaların birçoğunda hem girdiler hem de çıktılar dizi halindedir. Örneğin videolar birer görüntü dizisidir, cümleler kelime dizisidir veya hava durumu tahmininde önceki günlerin ve sonraki günlerin hava tahminleri bir dizinin parçasıdır.

YSA geleneksel sinir ağlarının tersine her bir t zamanında sadece mevcut girdiyi hesaba katmaz; bunun yerine mevcut girdi ve bir önceki girdinin ağda geçmesi sonucu elde edilen gizli durumun birleşimi güncel girdiyi oluşturur. YSA mimarisinin temelini oluşturan önceki girdiden elde edilen gizli durumu hesaba katma özelliği ileri beslemeli ağların çözemediği problemleri çözmek için kullanılır. Şekil 3.23'te YSA hücre yapısı gösterilmektedir.



Şekil 3.23. Yinelemeli sinir ağı hücre yapısı

Şekil 3.23'te g_t t zamanında gizli durumdur, a_t t zamanındaki girdidir ve g_t 3.4 ile hesaplanır.

$$g_t = K^{hh}g_{t-1} + K^{ah}a_t \quad (3.4)$$

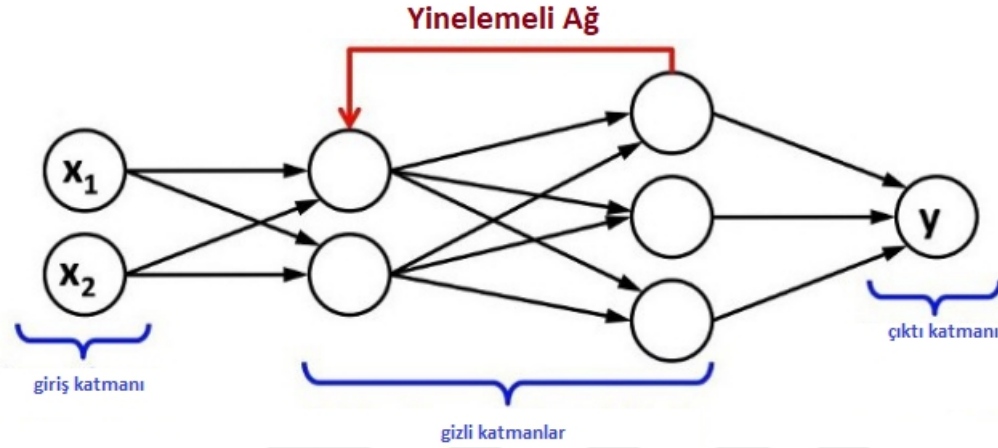
Burada K^{hh} önceki durum g_{t-1} yi gösteren ağırlık matrisidir. K^{ah} , a_t girdisini temsil için kullanılan ağırlık matrisidir ve c_t , t zamanında olasılık dağılımının çıktısıdır. Çıktı denklem 3.5 ile hesaplanır.

$$c_t = \text{softmax}(K^{yh}g_t) \quad (3.5)$$

Burada K^{yh} , t zamanındaki ağırlık matrisidir.

YSA önceki hücrenin gizli durumunu tutmak için bir hafızaya sahiptir. YSA'daki her bir hücrede önceki hücrenin gizli durumu aktivasyon fonksiyonu üzerinden geçirilir ve ardından mevcut hücrenin girdisi ile birleştirilir. Elde edilen değerde bir sonraki hücreye aktarılacak gizli durumu oluşturmaktadır. Fakat geleneksel YSA kısa süreli hafıza problemlerine sahiptir. Yani geleneksel YSA geçmişe ait bilgiyi güncel girdiyi tahmin etmek için kullanırken sadece birkaç adım geriye gidebilir, bu sebeple uzun dizilerde ağırlık başlangıcında öğrenilen bilgileri unutmaya meyillidirler [68]. Bunun yanında bir katmanda hesaplanan hata değerinin geriye yayılımı sırasında önceki katmanlara hata değerinin

küçük bir kısmı aktarılabilir. Bu durum literatürde kaybolan gradyan problemi (vanishing gradient problem) olarak tanımlanır. Geleneksel YSA, kaybolan gradyan problemini çözemez. YSA'nın bu konudaki eksikliklerinden ötürü UKSB, kapılı yineleyen birim gibi çeşitli YSA mimarileri geliştirilmiştir. Şekil 3.24'de örnek bir YSA mimarisi gösterilmektedir.



Şekil 3.24. Yinelemeli sinir ağ mimarisi [93]

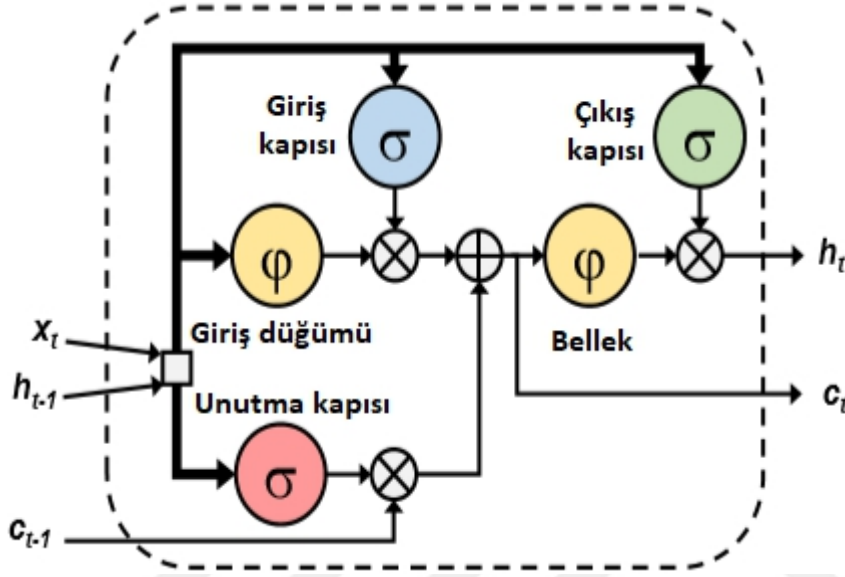
3.8. Dil Modeli

Dil modeli, sıralı olarak önceki kelimelere bağlı olarak sonraki kelimeyi tahmin ederek cümle oluşturabilir. YSA sıralı verilerdeki başarısından ötürü dil modelinin oluşturulmasında sıkça kullanılmaktadır. Bu tez çalışması dahilinde dil modeli oluşturmak için 3 farklı YSA mimarisi kullanılmıştır. Bunlar; UKSB, KYB ve iki yönlü uzun kısa süreli bellektir.

3.8.1. Uzun kısa süreli bellek

YSA bölümünde bahsedildiği gibi, YSA'nın temel problemlerinden birisi gradyan yok olması problemi. Bu problemin çözümü için UKSB geliştirilmiştir. UKSB içinde barındırdığı birçok kapı nedeniyle girdiler arasındaki bağımlılıkları yakalamada YSA'dan daha iyi sonuçlar verir. UKSB hücresinin klasik bir YSA'dan farklı olarak üç kapısı daha vardır: giriş kapısı, unutma kapısı ve çıkış kapısı. Bunların yanında YSA'daki gibi birde içinde bellek bulundurmaktadır. UKSB mimarisinde gizli katmanlarda normal nöronlar yerine UKSB hücreleri bulunur.

Giriş kapısı mevcut hücre durumuna bağlı olarak yeni girdiden hangi bilgilerin gireceğini kontrol eder. Unutma kapısı, önceki durumun mevcut duruma etkisini kontrol ederek hangi bilgilerin unutulması gerektiğine karar verir. Son olarak çıkış kapısı hücrenin mevcut durumuna göre bir sonraki gizli durumun ne olacağına karar verir. Şekil 3.25’de tipik bir UKSB hücresi gösterilmektedir.

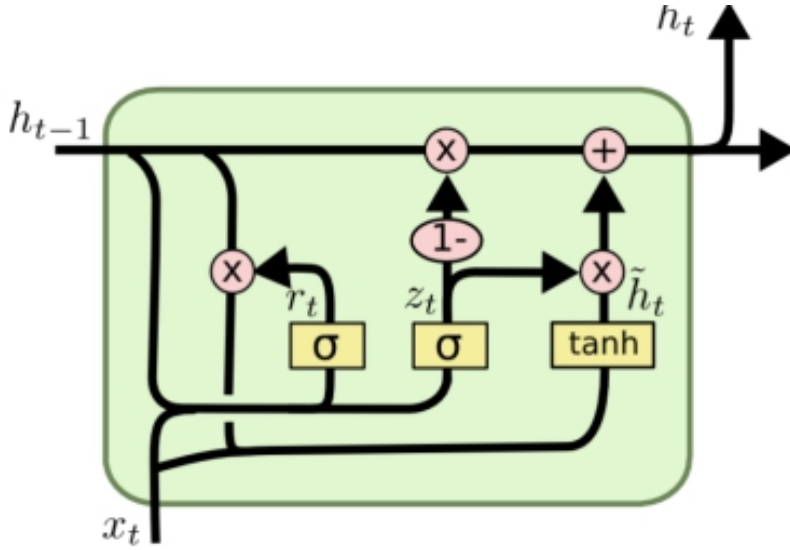


Şekil 3.25. Uzun kısa süreli bellek hücresi [94]

Yukarıdaki UKSB hücresinde x_t girdiyi, h_{t-1} bir önceki gizli durumu, c_{t-1} hücrenin bir önceki durumunu, h_t çıktığı ve c_t hücrenin durumunu temsil etmektedir.

3.8.2. Kapılı yineleyen birim

Kapılı yineleyen birim (Gated recurrent unit) [95], UKSB ile oldukça benzeyen bir mimaridir. KYB’de UKSB’den farklı olarak hücre durumu yoktur ve bilgi aktarımı için gizli durum kullanmaktadır. Bunun yanında KYB’de güncelleme ve sıfırlama kapıları bulunmaktadır. Güncelleme kapısı, UKSB’de ki giriş ve unutma kapılarına benzemektedir. Sıfırlama kapısı ise hangi geçmiş bilgilerin ileri aktarılacağına karar vermektedir [93]. KYB, UKSB’den daha hızlı çalışmaktadır. KYB örneği Şekil 3.26’da mevcuttur.

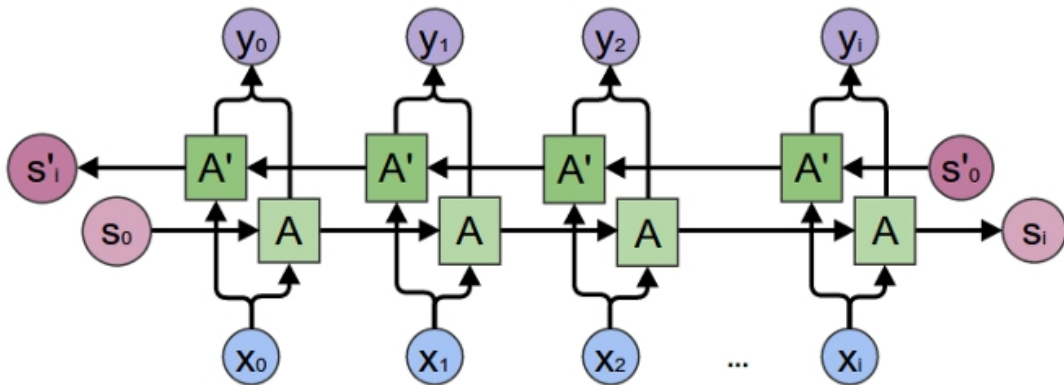


Şekil 3.26. Kapılı yineleyen birim hücresi [96]

Yukarıdaki KYB hücresinde x_t girdiyi, h_{t-1} bir önceki gizli durumu, h_t çıktıyı temsil etmektedir.

3.8.3. İki yönlü uzun kısa süreli bellek

İki yönlü uzun kısa süreli bellek temel olarak birbirinden bağımsız iki tane UKSB'nin bir arada kullanılmasıdır. Böylece ağda hem ileri hem de geri yönlü bilgi akışına olanak sağlanır. Diğer bir tabirle bilgi hem geleceğe doğru akarken hem de gelecekteki bilgi akışı sağlanır. İYUKSB mimarisi Şekil 3.27'de gösterilmektedir.

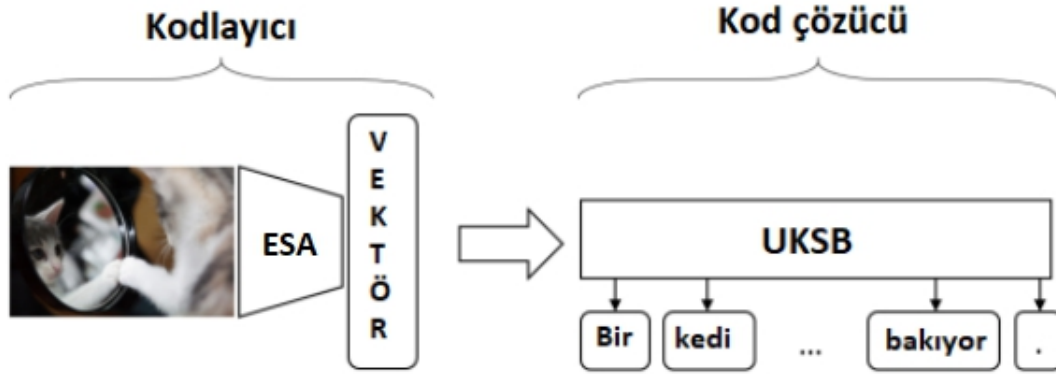


Şekil 3.27. İki yönlü uzun kısa süreli bellek mimarisi [97]

3.9. Kodlayıcı-Kod çözücü Mimarisi

Geleneksel derin yapay sinir ağlarında girdinin ve çıktının boyutunun önceden bilinmesi ve sabit olması gereklidir [98]. Fakat karşılaşılan problemlerin birçoğunda böyle bir durumla karşılaşmaz. Örneğin görüntülere başlık atama probleminde görüntüyü tasvir edecek cümlelerin kaç kelimeden oluşacağı sabit değildir veya bir dilden başka bir dile tercüme probleminde her iki dildeki kelime sayısı aynı olmayabilir. Mesela Türkçede 4 kelimeden oluşan "Güzel şeyler olacağını hissediyorum" cümlesi İngilizceye çevrildiğinde "I feel that good things will happen" şeklinde çevrilir ve 7 kelimeden oluşur. Bu ve benzeri problemleri çözmek için kodlayıcı-kod çözücü (encoder-decoder) mimarisi geliştirilmiştir.

Kodlayıcı-kod çözücü mimarisine sahip klasik bir ağ, girdi görüntüsünün özelliklerini ESA ile çıkarır. Çıkarılan özellikler görüntünün başlığının oluşturulması için YSA'ya beslenir. Şekil 3.28'de bu akışın gösterimi sağlanmaktadır.



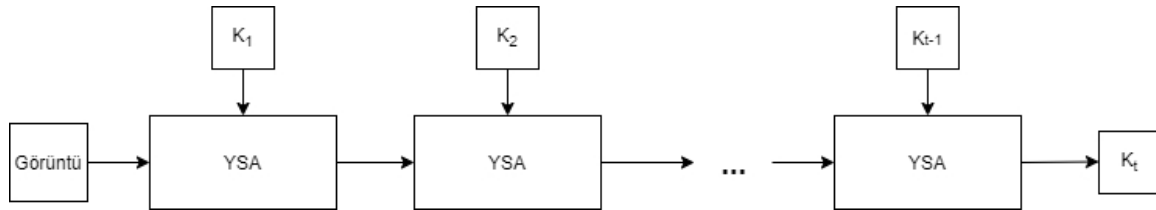
Şekil 3.28. Kodlayıcı-kod çözücü mimarisi [64]

Kodlayıcı bu yaklaşımda görüntünün görüntü vektörüne ve tercihen daha düşük boyutlu bir vektöre dönüştürülme sürecini ifade etmektedir. Burada görüntüye ait vektörün görüntüyü maksimum seviyede temsil etmesi ve maksimum bilgiyi içermesi beklenmektedir [64]. Kod çözme aşaması ise görüntüden çıkarılan vektörün tanımlamaya dönüştürülmesini ifade etmektedir.

3.10. Görüntülere Başlık Atama Modelleri

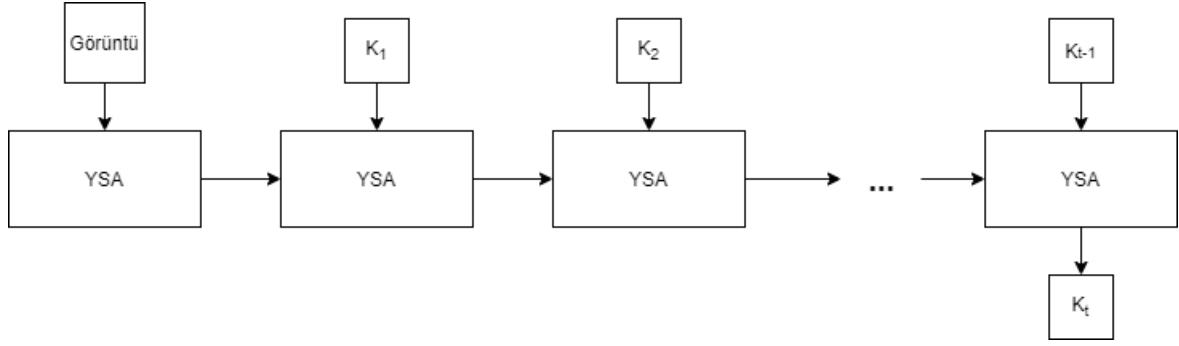
Görüntülere başlık atama alanındaki problemlerin çözümünde, kodlayıcı-kod çözücü mimarisine sahip çalışmaların başarılı olduğu gözlemlenmiştir. Kodlayıcı-kod çözücü mimarisinin uygulanması enjeksiyon (inject) ve birleştirme (merge) yaklaşımlarına ayrılabilir. Enjeksiyon yaklaşımında temel olarak girdi görüntüsüne ait özellik vektörü ile başlığın o ana kadar oluşturulmuş kelimeleri birleştirilir ve ardından YSA'dan geçirilerek tanımlamadaki bir sonraki kelime oluşturulur. Enjeksiyon yaklaşımının kendi içinde 3 farklı varyasyonu vardır. Bunlar ilk enjeksiyon (init-inject), ön enjeksiyon (pre-inject) ve paralel enjeksiyon (par-inject) yaklaşımlarıdır.

İlk enjeksiyon yaklaşımında YSA'nın ilk gizli durum vektörünü girdi görüntüsünün ESA aracılığıyla oluşturulan özellik vektörü oluşturur. Burada oluşturulan özellik vektörünün YSA'nın gizli durum vektörüyle aynı boyutta olması gerekmektedir. İlk enjeksiyon yaklaşımında görüntüye ait tanımlama cümlesi oluştuğunda, her bir kelime için bir önceki kelime ve bir önceki kelimenin gizli durum vektörü, mevcut kelimenin çıkarıldığı YSA'ya girdi olur. Şekil 3.29'da ilk enjeksiyon yaklaşımı gösterilmektedir.



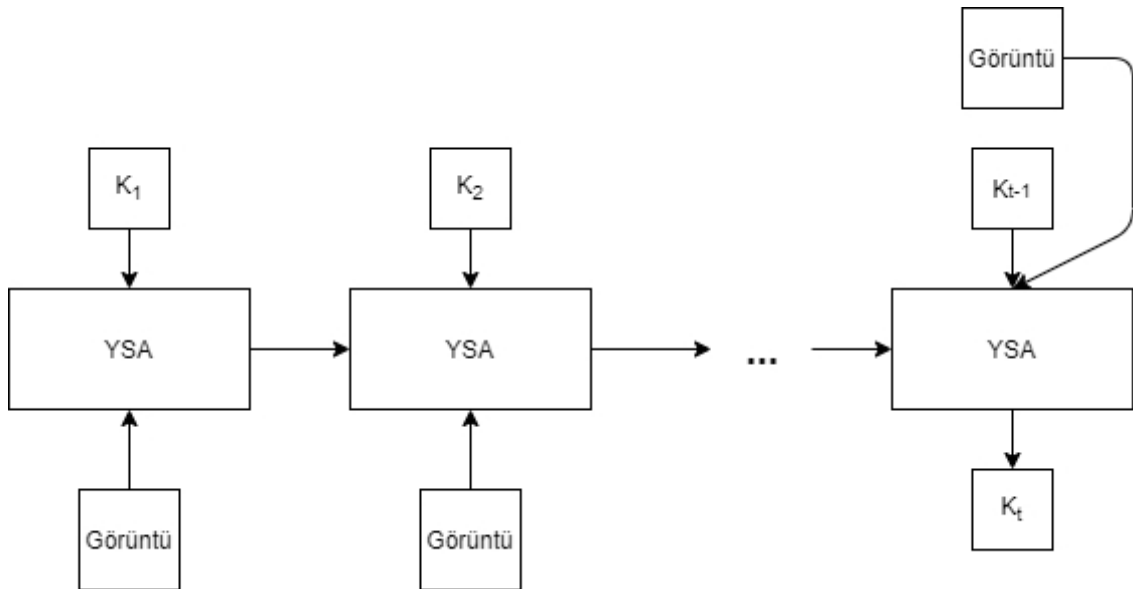
Şekil 3.29. İlk enjeksiyon yaklaşımı (K kelimeleri temsil etmektedir)

Ön enjeksiyon yaklaşımında YSA'nın ilk girdisi görüntü vektörüdür. Bu sebeple görüntü vektörü ilk kelime gibi değerlendirilir. Ön enjeksiyon yaklaşımında da görüntü vektörü ile kelime vektörlerinin aynı boyutta olması gereklidir. Ön enjeksiyon çalışmalarında bazen görüntü vektörleri yerine resmin özelliklerinin kullanılması veya görüntü vektörü ve görüntü özelliklerinin beraber girdi olması gibi varyasyonları vardır. Bununla beraber ilk enjeksiyon modelindeki gibi her bir kelime için bir önceki kelime ve bir önceki kelimenin gizli durum vektörü, mevcut kelimenin çıkarıldığı YSA'ya girdi olur. Şekil 3.30'da ön enjeksiyon yaklaşımı gösterilmektedir.



Şekil 3.30. Ön enjeksiyon yaklaşımı (K kelimeleri temsil etmektedir)

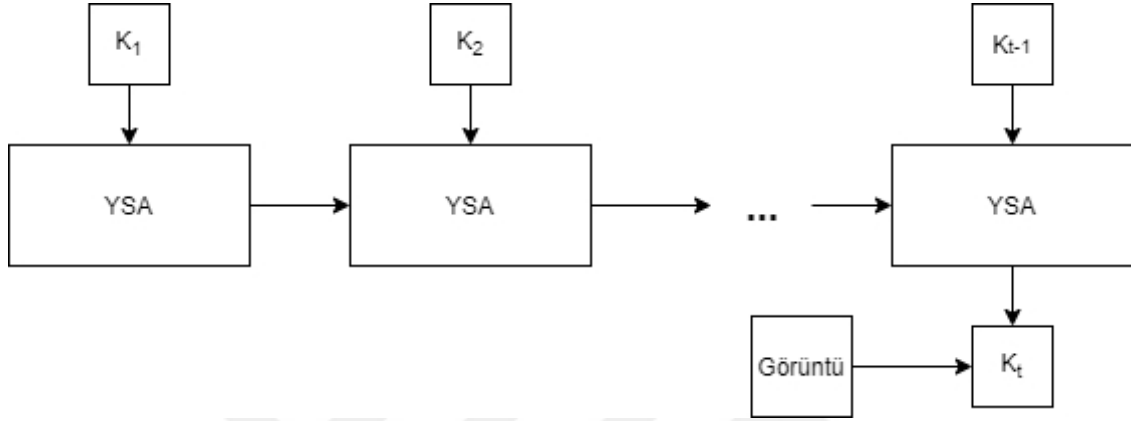
Paralel enjeksiyon yaklaşımında girdi görüntüsüne ait görüntü vektörü, başlık kelimelerine paralel olarak YSA'ya girdi olur. Burada kelime vektörü ile görüntü vektörü YSA'ya girmeden önce birleştirilir. Fakat paralel enjeksiyon yaklaşımında başlığa ait her bir kelimenin tahmininde görüntü vektörünün YSA'ya girmesi mutlak şart değildir. Yani paralel enjeksiyon yaklaşımında görüntü zaman zaman YSA'ya beslenir. Bunun sebebi veri kümesine bağlı olarak değişkenlik göstermekle beraber, her kelimedede görüntü vektörü YSA'ya girdiği zaman YSA aynı zamanda bir önceki kelimenin gizli durum vektörünü girdi olarak aldığı için YSA'nın görüntü vektörüyle şişmesinden kaynaklıdır. Şekil 3.31'de paralel enjeksiyon yaklaşımı gösterilmektedir.



Şekil 3.31. Paralel enjeksiyon yaklaşımı (K kelimeleri temsil etmektedir)

Üç enjeksiyon yaklaşımının yanında bir tane birleştirme (merge) yaklaşımı mevcuttur. Birleştirme yaklaşımında YSA'ya sadece kelime vektörü girer. Görüntü vektörü hiçbir

aşamada YSA'ya girdi olmaz. Bunun yerine YSA'dan geçmiş kelime vektörüyle birleşir ve yeni kelime tahmin edilir. Enjeksiyon modelinde görüntü sürekli olarak YSA'dan geçtiği için her bir adımda görüntünün temsili değişirken ve etkisi azalırken birleştirme modelinde görüntü geç bağlandığı için ve YSA'dan geçmediği için görüntünün temsili değişmez. Şekil 3.32'de birleştirme yaklaşımı gösterilmektedir.



Şekil 3.32. Birleştirme yaklaşımı (K kelimeleri temsil etmektedir)

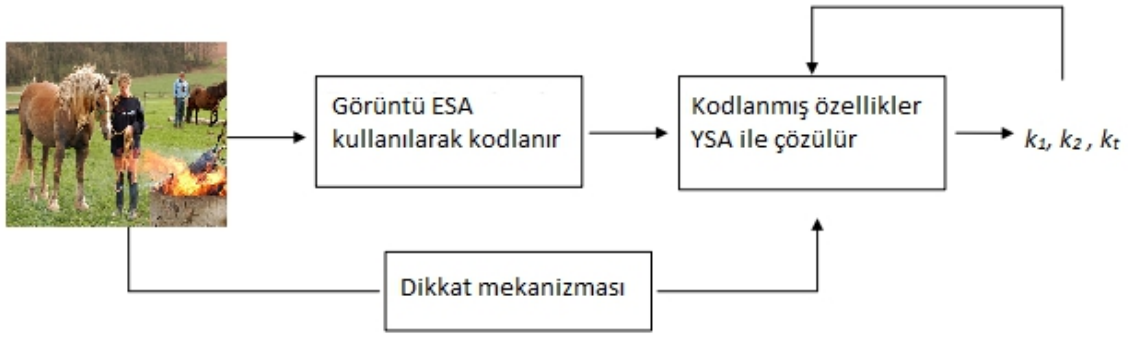
Önceki çalışmalar değerlendirildiğinde üç enjeksiyon yaklaşımı ve birleştirme yaklaşımı arasından birleştirme yaklaşımının en başarılı sonuçlara sahip olduğu gözlemlenmiştir [99]. Ortaya çıkan bu sonuca göre görüntü vektörü ile kelime vektörünün birleştirilmesinin geç aşamaya bırakılması başarımlarını artırmaktadır. Diğer bir çıkarılan sonuç ise kodlayıcı-kod çözücü mimari kullanan görüntülere başlık atama çalışmalarında, YSA'ya çıktı üretmek yerine, girdi kodlamak rolü verildiğinde çıkan sonuçlar daha iyi olmaktadır [99].

3.11. Dikkat Mekanizması

Kodlayıcı-kod çözücü mimarisi ile geliştirilen modellerin görüntülere başlık atama problemlerinin çözümünde de başarılı olduğu gözlemlenmiştir. Fakat bu çözümlerin tanımlama oluşturdukça, görüntüyü analiz ederken performansları azalmaktadır. Ayrıca kodlayıcı-kod çözücü mimariye sahip modeller görüntünün mekansal yönlerini dikkate almamaktadır. Bunun yerine görüntüye ait tanımlamaları görüntünün bütününden oluşturur. Buda oluşturulan başlıkların görüntüye ait bilgileri ifade edebilmesini azaltır.

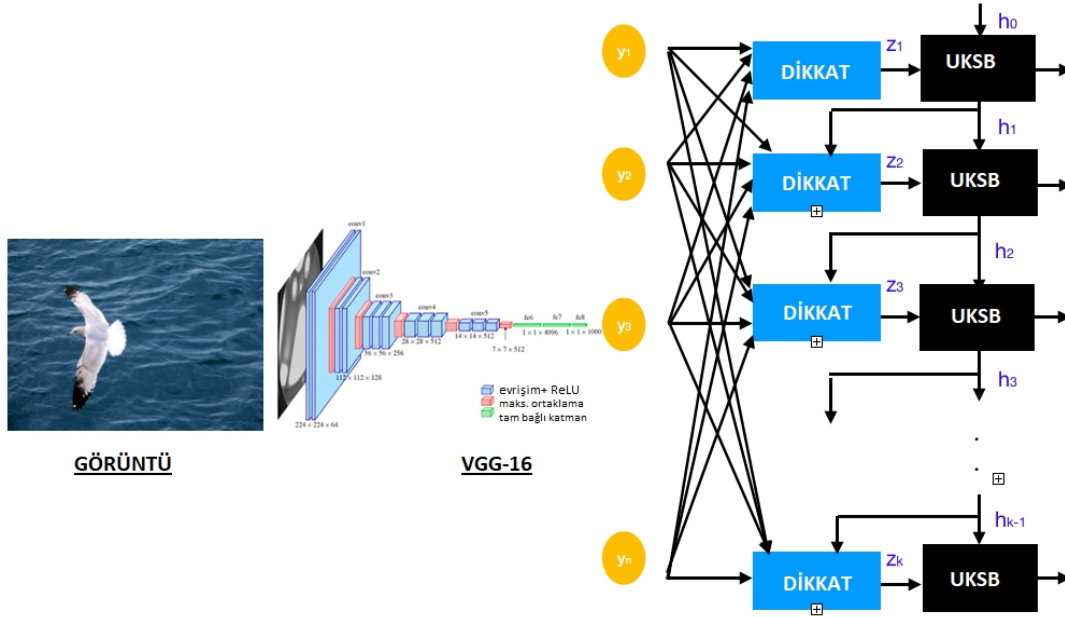
Bunun yanında görüntüler içerikleri açısından oldukça zengin bilgilere sahiptir. Fakat görüntülerden oluşturulan başlıklarda beklenen görüntüye ait en dikkat çekici noktaların

ifade edilmesidir. Bu noktada yapılan çalışmalarda insanların ve hayvanların görsel dikkat mekanizmalarından ilham alınarak dikkat mekanizması geliştirilmiştir [100]. Dikkat mekanizması kullanarak geliştirilen çalışmalar kodlayıcı-kod çözücü yaklaşımla geliştirilmiş çalışmaların gelişmiş sürümü olarak tanımlanabilir. Dikkat tabanlı yaklaşımlarda dikkat mekanizması kodlayıcı-kod çözücü tabanlı görüntülerden başlık oluşturma yaklaşımlarına dahil edilir ve başlıklar dikkat mekanizmasına göre hesaplanan gizli durumlara göre oluşturulmaktadır. Dikkat tabanlı görüntü alt yazılama modellerinin temel yapısı Şekil 3.33’de olduğu gibidir. Şekil 3.33’de k_1 , k_2 kelimeleri ifade etmektedir.



Şekil 3.33. Dikkat tabanlı başlık oluşturma modelinin genel yapısı

Dikkat mekanizmasına dayalı görüntülerden başlık oluşturma modelleri, modelin nereye bakacağını bilme kapasitesine sahip bilgisayarla görme algoritmaları sağladıkları için giderek daha popüler hale gelmektedir [101]. Dikkat mekanizması nereye bakacağını öğrenme işini görüntüye ait farklı bölgelerin içeriğini, önemine göre dinamik bir şekilde farklı ağırlıklandırarak yapmaktadır. Dikkat mekanizması kullanarak geliştirilen bir modelde klasik olarak önce girdi görseline ait özellikler ESA kullanılarak çıkarılır. Devamında YSA, dikkat mekanizmasının görüntüde odaklandığı bölgeye göre ve önceden ürettiği kelimeleri de hesaba katarak görüntüye ait başlığı oluşturur. Dikkat mekanizması kullanılarak oluşturulan görüntülere başlık atama mimarisi Şekil 3.34’deki gibidir.



Şekil 3.34. Dikkat mekanizması kullanılan görüntülere başlık atama mimarisi [102]

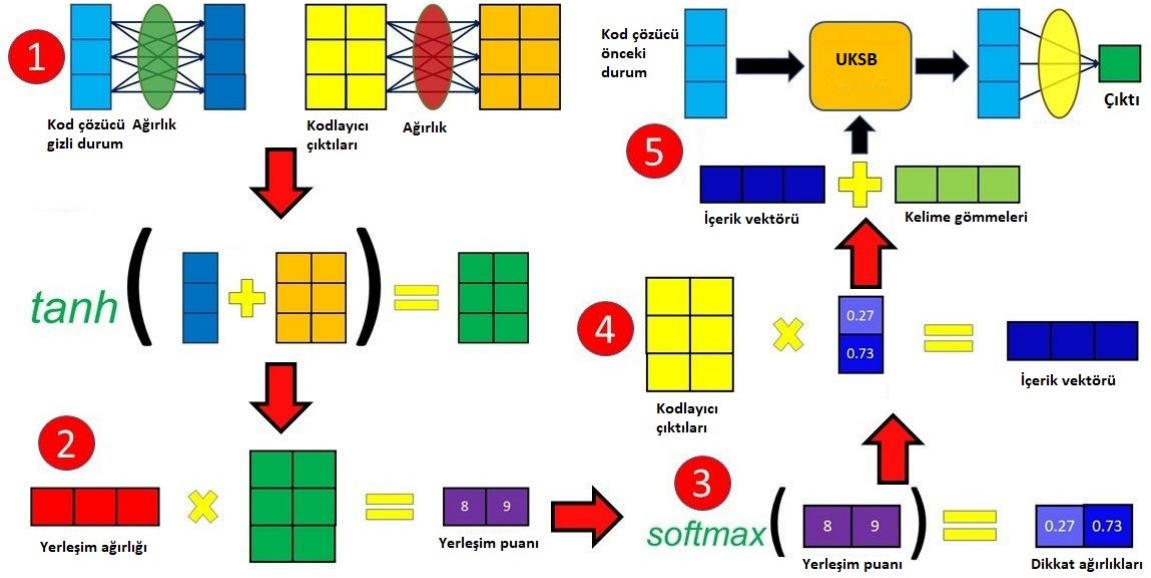
Bu tez çalışması kapsamında lokal ve global dikkat mekanizmaları kullanılmıştır. İki mekanizmanın kullanıldığı modellerde, kullanılmayan modellere göre kod çözücü aşamasında farklılıklar mevcuttur.

3.11.1. Lokal dikkat mekanizması

Bahdanau dikkat mekanizması diye de adlandırılabilen lokal dikkat mekanizması [103], Bahdanau ve diğerleri tarafından tasarlanmıştır. Lokal dikkat mekanizması temel olarak, girdi görüntüsündeki belli bir noktaya odaklanmayı amaçlamaktadır. Lokal dikkat mekanizması 5 adımda gerçekleşmektedir. Bu adımlar sırasıyla şöyledir:

- Kodlayıcı, girdi görüntülerinden gizli durumlar oluşturur.
- Kod çözücünün bir önceki gizli durumu ile bir numaralı adımda oluşan gizli durumlar arasında yerleşim puanı hesaplanır.
- Hesaplanan yerleşim puanları bir araya getirilir ve softmax fonksiyonundan geçirilerek olasılıksal değerleri (dikkat mekanizmasındaki ağırlıkları) hesaplanır.
- Kodlayıcının çıktısı olan gizli durumlar ile üçüncü adımda hesaplanan dikkat mekanizması ağırlıkları çarpılır ve içerik vektörü elde edilir.
- Önceki adımda elde edilen içerik vektörü ve daha önceden tahmin edilen kelimelerin gömmeleri birleştirilerek YSA'ya girdi olur.

Lokal dikkat mekanizmasının çalışma akışı şekil 3.35’deki gibidir.



Şekil 3.35. Lokal dikkat mekanizması çalışma akışı [104]

3.11.2. Global dikkat mekanizması

Luong dikkat mekanizması diye de adlandırılabilen global dikkat mekanizması [105], Luong ve diğerleri tarafından tasarlanmıştır. Bahdanau tarafından tasarlanan lokal dikkat mekanizmasından ilham alınarak geliştirilen model bu mekanizmanın daha basit bir hali olarak öne sürülmüştür. Global dikkat mekanizmasında yerleşim puanı hesaplanmadan önce YSA çalışır ve gizli durum üretir. Devamında yerleşim puanı lokal dikkat mekanizmasındaki gibi kod çözücünün bir önceki gizli durumu yerine yeni hesaplanan güncel gizli durum ile hesaplanır. Global dikkat mekanizmasında lokal dikkat mekanizmasından farklı olarak üç farklı yerleşim puanı hesaplama yöntemi mevcuttur. Sonraki adımlarda lokal dikkat mekanizması gibi ilerleyen global dikkat mekanizmasında, en son aşamada, YSA'nın çıktısıyla hesaplanan içerik vektörü birleştirilir ve tam bağlı katmana iletilir.

3.12. Derin Öğrenme Hiper Parametreleri

Derin öğrenme hiper parametreleri ağı nasıl eğitileceği, ağı yapısındaki değişkenler gibi birçok parametrenin genel çerçeve adıdır. Hiper parametreler veri kümesinden veri

kümesine, modeli tasarlayan kişiden modelin eğitileceği makinenin işlem gücüne göre değişebilir. Tasarlanan modelin başarımlar oranını yükseltmede hiper parametre seçimlerinin önemli bir rolü vardır. Yaygın kullanılan hiper parametreler şunlardır.

3.12.1. Öğrenme hızı

Geri yayılım ile ağırlıkların güncellendiği derin öğrenme modellerinde yeni ağırlık değerlerinin hesaplanmasında öğrenme hızı hiper parametresi kullanılır. Öğrenme hızı sabit olabileceği gibi değişken bir değerde olabilir. Eğer öğrenme hızı çok küçük olursa modelin ideal noktaya ulaşması çok uzun sürecektir. Fakat öğrenme hızı eğer çok büyük olursa da ideal noktaya ulaşmaktan ziyade modelin ağırlık değerleri sürekli bir salınım halinde olacaktır. Öğrenme hızının seçiminde yaygın kanı önce 0.001 gibi büyük bir değer seçip iterasyon sayısı arttıkça öğrenme hızını azaltma yönündedir [106].

3.12.2. Mini grup boyutu

Mini grup (mini batch) boyutu, veri kümesinin alt kümelerine ayrılması ve her bir alt kümede bulunan veri sayısını ifade etmek için kullanılan bir terimdir. Bütün veri kümesinin aynı anda modele girerek modelin eğitilmesi hem zaman hem de kullanılan hafıza açısından oldukça zorlayıcı bir işlemdir. Zaten birçok veri kümesi boyutu sebebiyle hafızaya sığmayacak kadar büyüktür. Bu sebeple veri kümesi birden çok parça halinde modele verilir. Mini grup boyutu için denenmesi önerilen değerler 4, 8, 16, 32, 64, 128 ve 256'dır [107].

3.12.3. Tur sayısı

Modelin eğitimi sırasında model iteratif bir şekilde eğitilir. Buradaki iterasyon sayısı tur (epoch) sayısı olarak adlandırılır. Eğitimin başlarında modelin çıkardığı sonuçlar başarısız olmakla birlikte tur sayısına bağlı olarak modelin başarımlar oranı da yükselir. Fakat belli bir noktadan sonra tur sayısının artması başarımlar oranını artırmaz. Yani belli bir noktadan sonra başarımlar oranı belli bir değere yakınsar.

3.12.4. Katman sayısı ve nöron sayısı

Derin öğrenme kavramındaki derin kavramı yapay sinir ağlarındaki katman sayısının fazlalığından gelmektedir. Yani derin öğrenmenin ayırıcı özelliklerinden biri katman sayısının fazlalığıdır. Yapılan çalışmalar neticesinde katman sayısının artmasının başarı oranını arttırdığı gözlemlenmiştir [66]. Fakat bu noktada sadece katman sayısını artırmanın başarı oranını belli bir noktadan sonra artırmadığı gözlemlenmiştir. Bunun en temel sebebi geri beslemeli ağlarda katman sayısı arttıkça ilk katmanlardaki ağırlık değişimi oldukça sınırlı kalmasından kaynaklıdır. Nöron sayısı temel olarak modelin hafızasında tutulan bilgi sayısı olarak tanımlanabilir. Girdi boyutunun küçük olduğu çalışmalar için az nöron sayısı belirlemek daha başarılı sonuçlar verebilirken, büyük girdiler için nöron sayısının artırmanın başarı oranını arttırdığı deneyimlenmiştir [108].

3.12.5. Optimizasyon algoritması

Derin öğrenme modellerinde optimizasyon algoritmaları modelin kayıplarını azaltmak için sinir ağındaki ağırlıkları değiştirme işlevine sahiptirler. Optimizasyon algoritmaları ile modelin en optimal sonucu vermesi için ağırlıkların optimum değeri bulunmaya çalışılmaktadır. Literatürdeki çalışmalarda yaygın olarak stokastik gradyan inişi, adaGrad, adaDelta, RMSProp yöntemleri kullanılmaktadır [109].

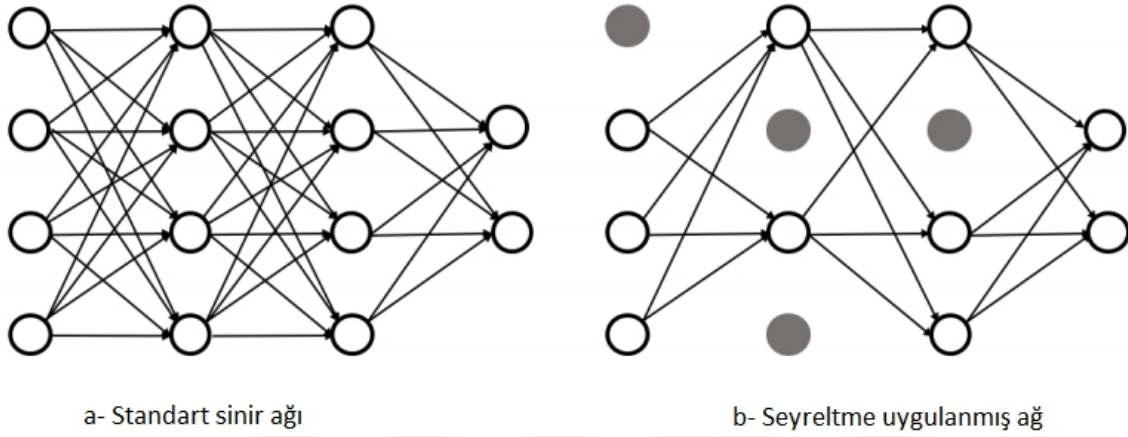
3.12.6. Başlangıç ağırlıklarının belirlenmesi

Modelin eğitime başlamadan önce ağırlıklar belirli değerler ile başlatılabilir. Ağırlıkların başlatılması için bütün ağırlıklara 0 verilmesi, bütün ağırlıklara 1 verilmesi, ağırlıklara 0 ile 1 aralığındaki rastgele sayılar verilmesi, sabit bir değerle ağı başlatmak gibi birçok seçenek vardır. Bu seçenekler arasından ağırlıkları küçük fakat rastgele sayılarla başlatmanın daha iyi sonuçlar verdiği gözlemlenmiştir [110].

3.12.7. Seyreltme

Yapay sinir ağlarında aşırı uyum (overfitting) önemli bir problemdir. Aşırı uyum eğitim verisinin ezberlenmesi ile olur ve böyle bir modelin eğitim verisinden farklı bir veri ile karşılaşması durumunda başarı oranı oldukça düşük olmaktadır. Aşırı uyumun önüne

geçmek için önerilen yöntemlerden bir tanesi seyreltme (dropout) uygulamaktır. Ağdaki belli bir eşik değerinin altındaki nöronların seyreltilmesinin ağın genel başarısını arttırdığı deneyimlenmiştir. Bu işlem ile ağdaki çok önemli olmayan bilgilerin unutulması sağlanmaktadır. Seyreltme değeri olarak genelde 0.5 seçilmektedir. Fakat bu ve diğer bütün hiper parametre değerleri veri kümesine göre, tasarlanan modele göre ve probleme göre değişiklik gösterebilir. Şekil 3.36'da örnek bir seyreltme uygulaması gösterilmektedir.

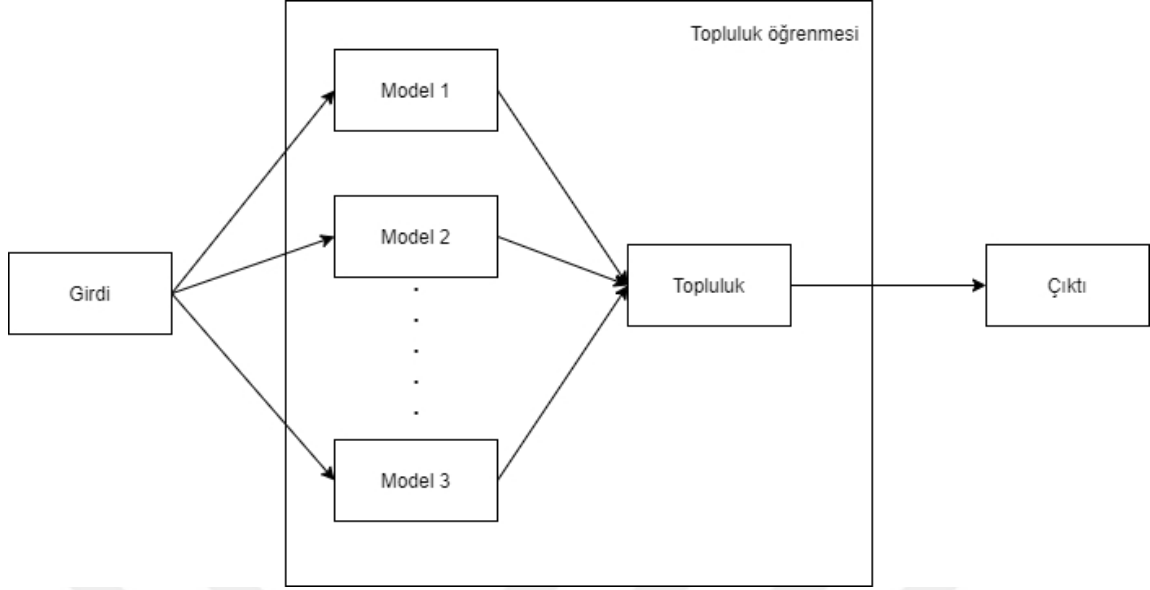


Şekil 3.36. (a) 2 gizli katmanlı standart sinir ağı. (b) Seyreltme uygulanarak üretilmiş sinir ağı [111]

3.13. Topluluk Öğrenmesi

Makine öğrenmesi alanı ve alt alanlarında mevcut problemleri çözmek için farklı yaklaşımlar mevcuttur. Bu yaklaşımlardan biri de topluluk öğrenmesi (ensemble learning) yaklaşımıdır. Genel olarak derin öğrenme problemlerinde modeller değişkenliklerin yüksek olduğu veri kümelerinde çok iyi sonuçlar alamazlar. Veri kümesindeki gürültü veya uç değerler gibi yüksek değişkenlikler modelin ağırlıklarını yanlış yönlendirebilir ve dolayısıyla modelin başarı oranı düşebilir [112].

Topluluk öğrenmesi ise tek bir modelin düşebileceği yüksek değişkenlik problemine çözüm oluşturmaktadır. Topluluk öğrenmesi, problemin çözümünde tek bir model ile çalışmak yerine birden çok modelin beraber kullanılması ile yeni bir model ortaya çıkarılmasını amaçlamaktadır. Bu tarz bir yaklaşımın bazı derin öğrenme problemlerinde tek bir model kullanılarak elde edilen sonuçlara göre daha başarılı sonuçlar verdiği gözlemlenmiştir [113]. Şekil 3.37'de topluluk öğrenmesinin gösterimi verilmiştir.



Şekil 3.37. Topluluk öğrenmesi

3.14. Veri Kümesi

Görüntülere başlık atama konusunda İngilizce dilinde çok sayıda ve büyük veri kümeleri vardır. Fakat Türkçe dilindeki çalışma sayısı oldukça az olduğundan Türkçe veri kümesi sayısı da oldukça sınırlıdır. Özellikle görüntülere başlık atama problemi gibi kompleksliğin yüksek olması sebebiyle ilginin daha az olduğu bir alanda diğer alanlara kıyasla Türkçe veri kümeleri bulmak son derece zordur. Öyle ki bu alanda sadece bir tane Türkçe veri kümesi vardır. TasvirEt isimli Türkçe görüntülere başlık atama veri kümesi oluşturulurken Flickr8K isimli veri kümesindeki görüntüler için Türkçe açıklamalar toplanmıştır. Flickr8K veri kümesi, görüntüleri Flickr web sitesindeki gruplardan alınarak bir araya getirilmiş görüntülerden oluşan bir veri kümesidir.

TasvirEt veri kümesinde Flickr8K veri kümesinden alınan 8091 görüntü mevcuttur ve bu görüntüler için toplamda 16037 adet açıklama vardır. Açıklamalardaki ortalama kelime sayısı 8'dir. TasvirEt veri kümesine ait istatistik bilgileri Çizelge 3.1'de verilmiştir.

Çizelge 3.1. TasvirEt veri kümesi istatistik bilgileri

Görüntü sayısı	8091
Toplam açıklama sayısı	16037
Ortalama kelime sayısı	8

TasvirEt veri kümesindeki en çok kullanılan 15 kelime ve frekansları Çizelge 3.2’de gösterilmektedir.

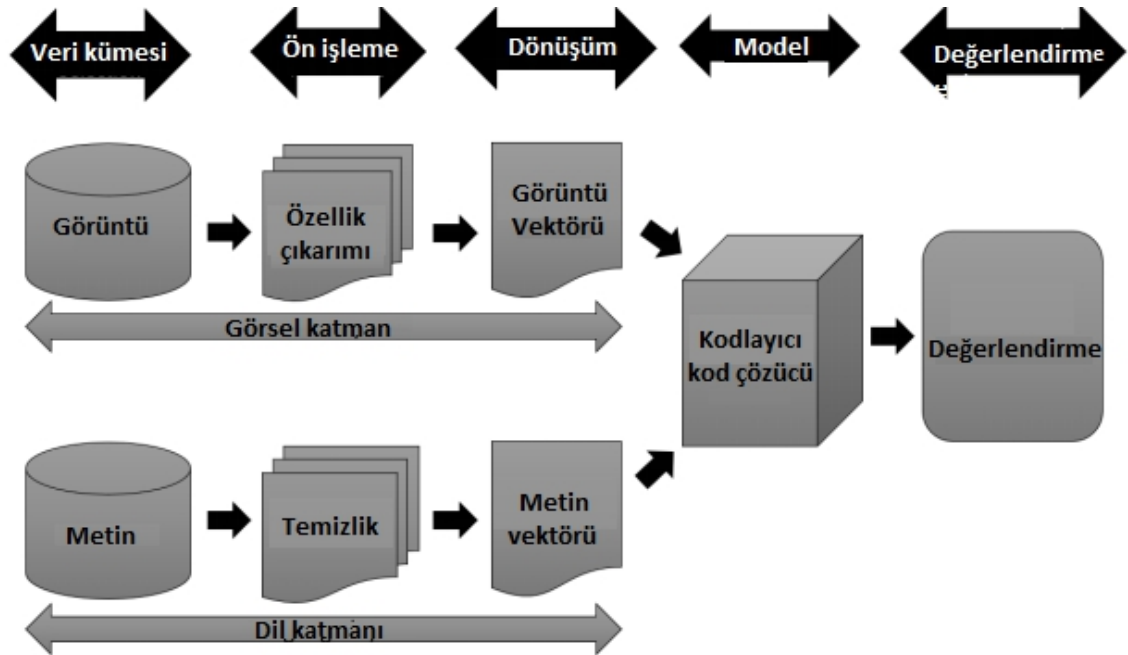
Çizelge 3.2. TasvirEt veri kümesindeki en sık kullanılan 15 kelime ve frekansları

	Kelime	Frekans
1	bir	12331
2	köpek	2610
3	adam	1822
4	iki	1637
5	plan	1608
6	ve	1517
7	çocuk	138
8	kız	1121
9	kadın	1102
10	çocuğu	1050
11	ile	1012
12	küçük	769
13	üzerinde	758
14	siyah	749
15	beyaz	737



4. GÖRÜNTÜLERE BAŞLIK ATAMA İÇİN ÖNERİLEN YAKLAŞIM

Bu bölümde bir önceki bölümde açıklanan kuramsal temeller üstüne görüntülere başlık atama çalışmamıza özgü önerilen yaklaşım aktarılacaktır. Önerilen yaklaşım dört bölümden oluşmaktadır. Bu bölümler sırasıyla veri ön işleme, veri dönüşümü, model ve değerlendirme bölümleridir. Çalışmada kullanılan veya yeni geliştirilen modellere göre değişiklik göstermekle birlikte, Şekil 4.1’de çalışmanın üst seviye genel adımları gösterilmektedir.



Şekil 4.1. Üst seviye çalışma adımları [114]

4.1. Veri Ön İşleme

TasvirEt veri kümesinde görüntüleri tanımlayan tanımlamalar, kitle kaynak yöntemiyle, isteyen herkesin katkı sağlayabileceği şekilde toplanmıştır [39]. Bu sebeple veri kümesindeki tanımlamalar gürültüli'dür. Veri kümesindeki gürültünün etkisini azaltarak elde edilecek sonuçlardaki başarıyı arttırmak için öncelikle veri kümesi üzerine bir dizi ön işleme adımları uygulanmıştır. Bunlar sırasıyla:

- Veri kümesindeki büyük harflerin küçük harflere dönüştürülmesi,
- Noktalama işaretlerinin kaldırılması,

- Özel sembollerin kaldırılması,
- Numaraların temizlenmesi,
- Numara içeren kelimelerin temizlenmesi,
- Tek harften oluşan kelimelerin temizlenmesidir.

Bu işlemlerin ardından temiz hale gelen veri kümesi üzerinde çalışmak için veri kümesi içindeki cümleler sözcüklerine ayrılmış ve bütün sözcüklerden oluşan sözcük hazinesi oluşturulmuştur. Ayırma işlemi için boşluk karakteri kullanılmıştır. Fakat oluşturulan sözcük hazinesindeki kelimelerin çoğu bir kez veya iki kez veya üç kez geçmektedir. Bununla beraber derin öğrenme modellerinin birçoğunda en temel gereksinim, veri sayısının fazlalığıdır. Yani bütün bir veri kümesinde sadece iki kere geçen bir kelimeyi modelin öğrenerek test zamanında doğru tahmin etme olasılığı oldukça zayıftır. Bu sebeple veri kümesinde kök hali dörtten az geçen kelimeler sözcük dağarcığından kaldırılmıştır. Kök hali hakkında ayrıntılı bilgi bir sonraki bölümde verilmiştir.

Veri kümesindeki metinlerin yanı sıra görüntülerin de ön işlemeden geçmesi gerekmektedir. Görüntüler modele vektör formunda girmektedir ve kullanılan ESA mimarisine göre değişmekle birlikte her ESA mimarisinin kabul ettiği kendine özgü bir girdi boyutu vardır. ESA modellerine ilerde değinilecek olmakla beraber, örneğin VGG19 mimarisi girdi boyutu olarak 224x224x3'ü kabul etmektedir. Inception v3 ise girdi boyutu olarak 299x299x3'ü kabul etmektedir. Bu sebeple kullanılan ESA mimarisine göre girdi görüntülerinin yeniden boyutlandırılmıştır. Ön işlemenin son aşamasında görüntü vektörleri ESA'ya girmeden evvel normalize edilmiştir. Normalize etme işlemi de ESA mimarisine göre değişiklik göstermektedir. Örneğin VGG16 mimarisinde pikseller 0 merkezli olarak $[-1,1]$ arasında değerlendirilecek şekilde normalize edilirken, Inception v3 mimarisinde her bir piksel $[-2,0]$ arasında değerlendirilecek şekilde normalize edilmiştir.

4.2. Veri Dönüşümü

Temizlenmiş veriler modele beslenmeden önce verileri modelin işleyebileceği bir formata dönüştürmek gereklidir. Geliştirilen modellerde başlık oluşturulurken başlığı oluşturan kelimeler sıralı bir şekilde tahmin edilir. Yani, model eğitilirken girdi görüntüsünü tanımlayan başlığın t_x anında $k_1, k_2, \dots, k_{x-2}, k_{x-1}$ kelimeleri modelin girdisi olur ve k_x .

kelimenin modelin çıktısı olması beklenir. Bunun yanında geliştirilen modelin, başlığın ne zaman başladığını ve ne zaman bittiğini öğrenebilmesi için benzersiz birer başlangıç ve bitiş kelimelerini her başlığa eklemek gereklidir. Bu sebeple veri kümesindeki her başlığın başına ve sonuna benzersiz "<basKelime>" ve "<bitKelime>" simge kelimeleri eklenmiştir.

Bununla birlikte giriş bölümünde bahsedildiği gibi Türkçe dilinin sondan eklemeli yapısından dolayı, Türkçe dili üzerinde yapılan derin öğrenme çalışmalarında başarı oranını artırmak için, diğer dillere göre ekstra işlemler gereklidir. Bu bağlamda, veri kümesindeki kelimelerin köklerine ayrılarak modellere girdi olması sağlanmıştır. Kelimeleri köküne ayırma işleminde Türkçe Kelime Kök Ayırıcı [115] uygulaması kullanılmıştır. Kelimeleri köklerine ayırmanın iki amacı vardır. İlk olarak aldıkları eklerden dolayı farklı formlarda bulunan, fakat aslında aynı kökten türeyen kelimeleri bulmak ve sözcük hazinesini buna göre oluşturmak amaçlanmıştır. Aksi halde veri kümesinde kök hali dörtten fazla kez bulunan fakat aldığı eklerden dolayı farklı formlarda gözüken kelimeler sözcük hazinesine girmemektedir. Bu durumun örneği Çizelge 4.1'te gösterilmektedir. Eğer kelimeleri köklerine ayırma işlemi gerçekleştirilmeden dörtten az kere veri kümesinde bulunan kelimeleri sözcük hazinesine dahil etmeyerek sözcük hazinesi oluşturulsa idi “afiş” kelimesi kökünden türeyen hiçbir kelime sözcük hazinesinde bulunamayacak ve dolayısıyla geliştirilen model “afiş” kelimesi kökünden türeyen hiçbir kelimeyi öğrenemeyecekti.

Çizelge 4.1. Veri kümesindeki “afiş” kelimesi kökünden türeyen kelimelerin miktarı

Kelime	Miktar
Afiş	1
Afişi	1
Afişin	1
Afişinin	1
Afişlerin	1

Fakat çalışma dahilindeki bütün deneylerde kelimeler köklerine ayrılmamıştır. Köke ayırma işleminin başarıya etkisini ölçümlemek amacıyla bazı çalışmalarda sonuçlar kelimelerin köklerine ayrılmış haliyle elde edilmiş, bazı çalışmalarda da köke ayırma

işlemi yapılmadan sonuçlar elde edilmiştir. Aynı şekilde kelimelerin köklerine ayrılmış formlarının veri kümesindeki toplam frekansını belirleme çalışması da bazı deneylerde yer alırken bazı deneylerde yer almamıştır. Elde edilen sonuçların hepsi karşılaştırmalı olarak deneysel sonuçlar bölümünde sunulmuştur. Kelimeleri köklerine ayırmanın ikinci sebebi ise kelime gömmelerinde değeri bulunamayan kelimelerin kök formlarının kelime gömmelerinin değerini kullanarak çalışmaktır. Bu konu hakkında daha detaylı bilgi bu bölümün sonunda bulunmaktadır.

Bir sonraki aşamada veriler, modelin eğitime girecek hale dönüştürülür. Daha önce de bahsedildiği gibi görüntülere başlık atama probleminin çözümünde görüntüye ait başlık kelime kelime tahmin edilir. Örnekleme gerekirse veri kümesinden bir başlık: “Arabanın hemen yakınında bir kriket sopasıyla topa vurmaya çalışan bir kız çocuğu” şeklindedir. Bu tanımlamanın tahmini Çizelge 4.2’deki gibi gerçekleştirilir.

Çizelge 4.2. Görüntülere başlık atama probleminde modelin girdileri ve çıktısı

Zaman	Girdiler		Çıktı
	Görüntü vektörü	Kısmi başlık	Hedef kelime
1	Görüntü 1	<basKelime>	arabanın
2	Görüntü 1	<basKelime> arabanın	hemen
3	Görüntü 1	<basKelime> arabanın hemen	yakınında
..	Görüntü 1	<basKelime> arabanın hemen yakınında	
13	Görüntü 1	<basKelime> arabanın hemen yakınında bir kriket sopasıyla topa vurmaya çalışan bir kız çocuğu	<bitKelime>

Bunun yanında çalışma dahilinde geliştirilmiş modellerin hiçbiri doğrudan metin üzerinde çalışmamaktadır. Bu sebeple sözcük hazinesindeki tüm kelimeler tam sayıya dönüştürülür. Örneğin;

- [Çimlerin üstünden koşarak gelen bir köpek]
- [64, 78, 249, 864, 1,2]

şeklinde dönüştürülür. Çalışma dahilinde modelin eğitimine giren tanımlamaların uzunluklarını eşitlemek adına veri kümesindeki en uzun cümle uzunluğuna eşit olacak şekilde her cümle sonuna sıfır ekleme (padding) işlemi yapılmıştır. Yani, üstteki örnekteki tanımlamanın sıfır eklendikten sonraki görünümü şu şekildedir:

- [64, 78, 249, 864, 1,2, 0, 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0]

Son işlem olarak veri kümesindeki her kelime, gömme (embedding) katmanından geçirilerek kelimelerin vektörel temsilleri ile işlem yapılması sağlanmıştır. Bu işlem için önceden eğitilmiş Word2vec modeli kullanılmıştır. Köksal tarafından geliştirilen [116] modelin oluşturulması için Vikipedi'deki Türkçe makaleler taranmış ve Gensim [117] kütüphanesi kullanılarak makalelerde yer alan kelimeler vektörize edilmiştir [116]. Bu modelde 150.000'e yakın kelime bulunmasına karşın dilimizin sonradan işlemeli yapısı gereği TasvirEt veri kümesinde yer alan bütün kelimelerin vektörel gösterimi kullanılan Word2vec modelinde bulunmamaktadır. Bu sebeple bir kelime Word2vec modelinde bulunmadığı zaman, bulunamayan kelime köküne ayrıldıktan sonra vektörel gösterimi elde edilmektedir. Fakat kelimeleri köke ayıran uygulama her zaman en doğru şekilde ayıramayabilmektedir. Bu gibi durumlarda kelimenin vektörel gösterimi bulunamamaktadır. Diğer bir durum ise kelimenin köke ayrılmış hali doğru olsa dahi Word2vec modelinde kök yer almayabilmektedir. Fakat bu iki durumun görülmesi oldukça kısıtlıdır. Bu durumlarla karşılaşıldığında ilgili kelime vektörü olarak 0 ayarlanmaktadır.

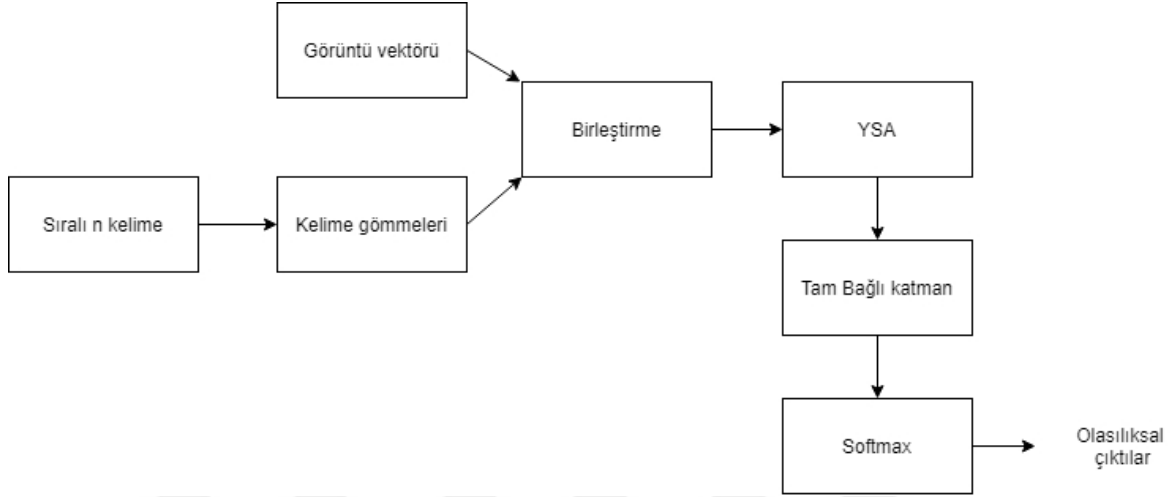
4.3. Model

Bu tez çalışması kapsamında 10 farklı model kullanılmıştır. Bu modellerin bazıları hali hazırda geliştirilmiş ve literatürdeki çalışmalarda sıklıkla kullanılan modeller olmakla birlikte bazı modeller mevcut modeller üzerine yeni geliştirmeler yapılarak bu çalışma kapsamında oluşturulmuştur. Bazı modeller ise topluluk öğrenmesi tekniği altında beraber çalıştırılmıştır.

4.3.1. Enjeksiyon modeli

Enjeksiyon modelinde evrişim sinir ağında özellik vektörü çıkarılmış görüntü ile kelime gömmeleri elde edilmiş kelimelerin vektörleri birleştirilerek YSA'ya girer ve elde edilen

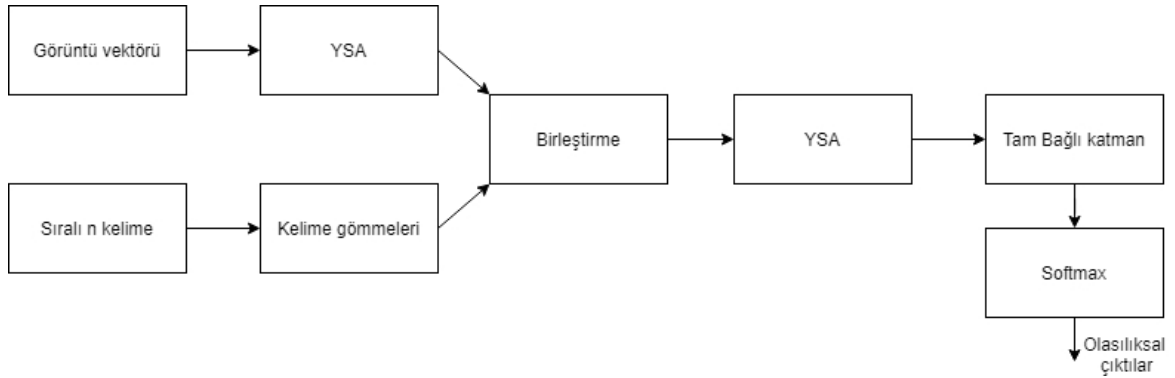
sonuç tam bağı katmandan geçirilerek modelin çıktısı elde edilir. Şekil 4.2’de enjeksiyon modeli gösterilmektedir.



Şekil 4.2. Enjeksiyon modeli

4.3.2. Alternatif enjeksiyon modeli

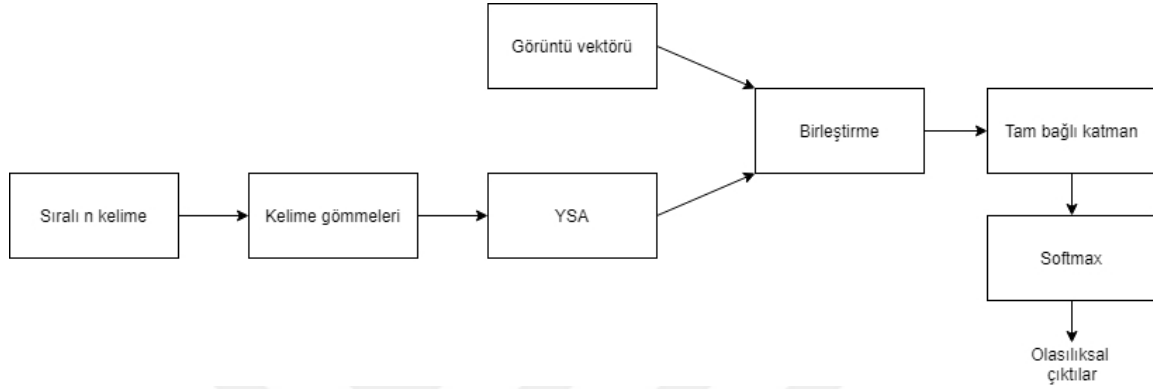
Alternatif enjeksiyon modeli daha önce geliştirilmemiş bir modeldir ve bu çalışma kapsamında enjeksiyon modelinin üzerine geliştirme yapılarak elde edilmiştir. Bu bağlamda enjeksiyon modeline ek olarak ESA’nın çıktısı olarak bulunan görüntülere ait özellik vektörü YSA’dan geçirilir. Elde edilen çıktı kelimeleri gömmeleri ile birleştirilerek oluşturulan vektör YSA’ya girer. YSA’nın çıktısı tam bağı katmandan geçirilerek modelin çıktısı elde edilir. Şekil 4.3’de alternatif enjeksiyon modeli gösterilmektedir.



Şekil 4.3. Alternatif enjeksiyon modeli

4.3.3. Birleştirme modeli

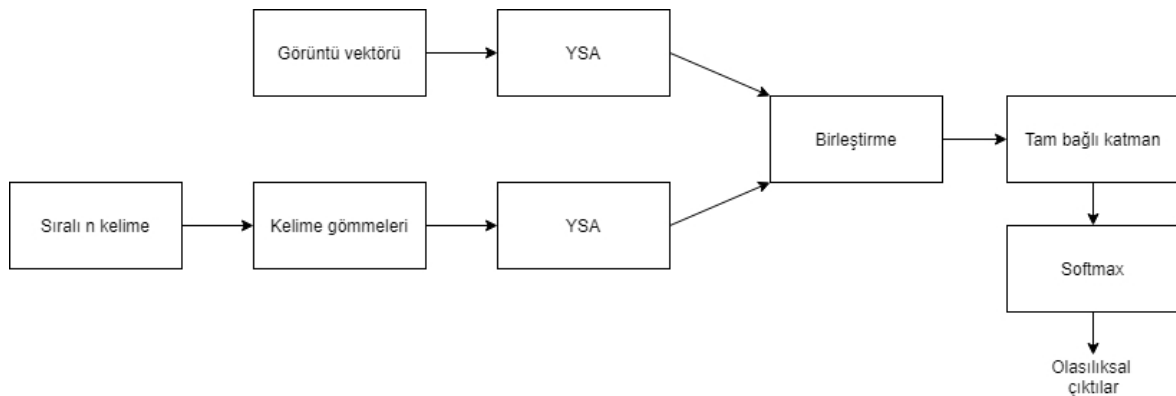
Birleştirme modelinde görüntüler YSA'ya girdi olmaz, bunun yerine kelime gömmeleri YSA'ya girer ve YSA'nın çıktısı ile görüntüye ait özellik vektörü birleştirilerek tam bağlı katmana iletilir ve modelin çıktısı alınır. Şekil 4.4'de birleştirme modeli gösterilmektedir.



Şekil 4.4. Birleştirme modeli

4.3.4. Alternatif birleştirme modeli

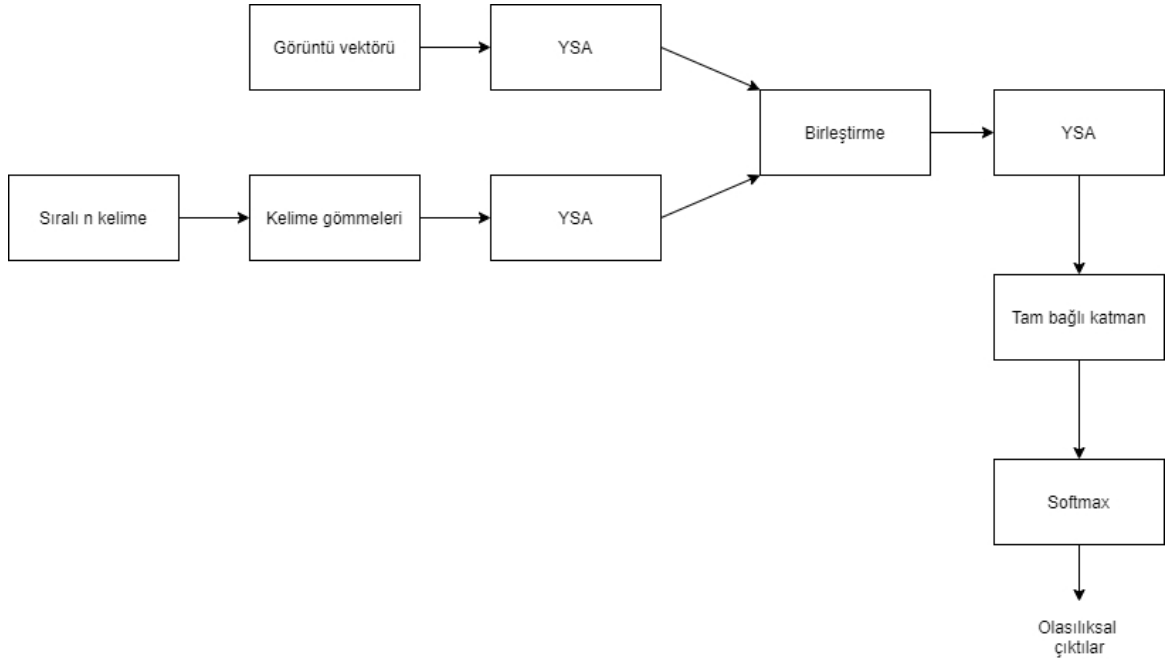
Alternatif birleştirme modeli daha önce geliştirilmemiş bir modeldir ve bu çalışma kapsamında birleştirme modelinin üzerine geliştirme yapılarak elde edilmiştir. Bu bağlamda birleştirme modeline ek olarak hem kelime gömmeleri hem de görüntülere ait özellik vektörü ayrı kanallarda YSA'dan geçirilir. Devamında YSA'ların çıktıları birleştirilerek tam bağlı katmana iletilir ve modelin çıktısı alınır. Şekil 4.5'de alternatif birleştirme modeli gösterilmektedir.



Şekil 4.5. Alternatif birleştirme modeli

4.3.5. BirEn modeli

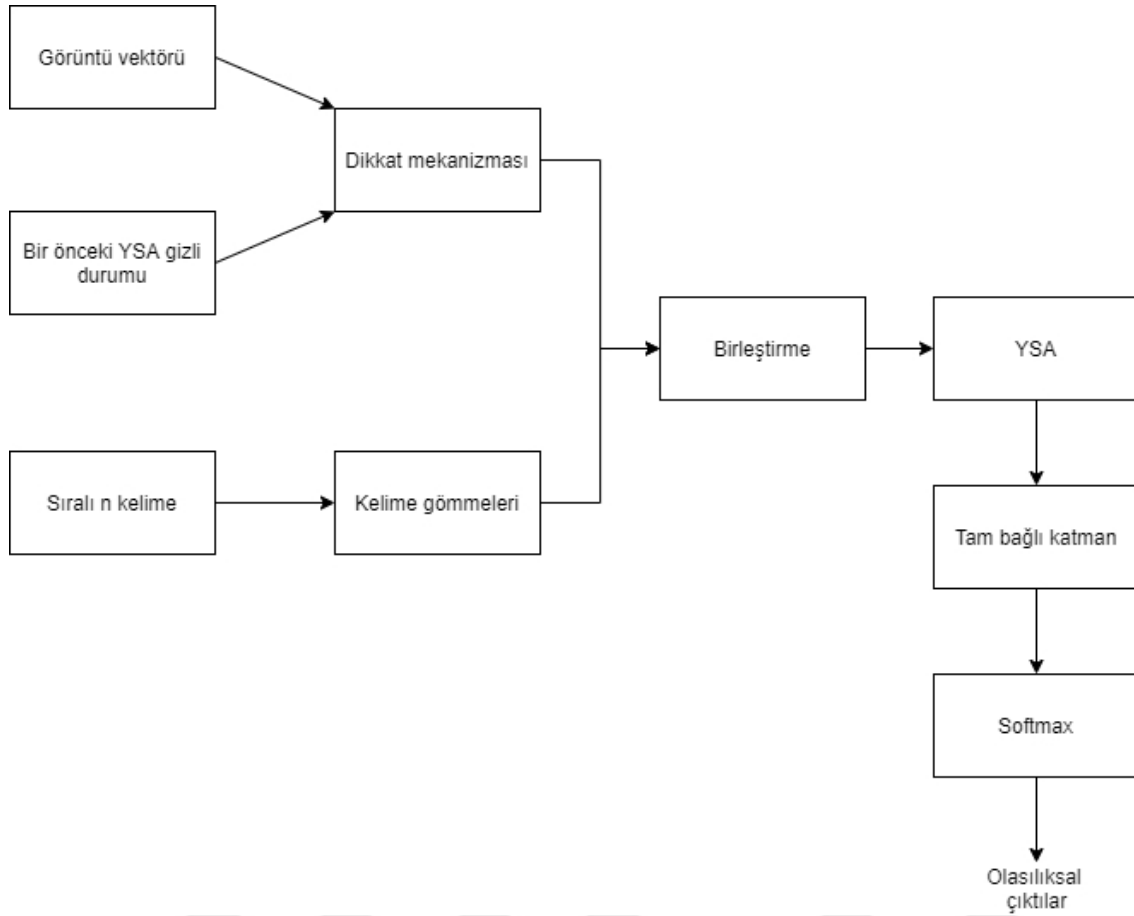
Birleştirme ve enjeksiyon modellerinin isimlerini bir araya getirerek isimlendirdiğimiz BirEn modeli de daha önce geliştirilmemiş bir modeldir ve bu çalışma kapsamında birleştirme ve enjeksiyon modellerinin üzerine geliştirme yapılarak elde edilmiştir. Bu bağlamda her iki modelin özellikleri bir araya getirilmiştir. Modelde görüntüye ait özellik vektörü ile YSA'dan geçirilen kelime gömmeleri birleştirilir ve devamında YSA'ya girerler. YSA'nın çıktısı tam bağlı katmandan geçirilerek modelin çıktısı elde edilir. Şekil 4.6'da BirEn modeli gösterilmektedir.



Şekil 4.6. BirEn modeli

4.3.6. Lokal dikkat mekanizması modeli

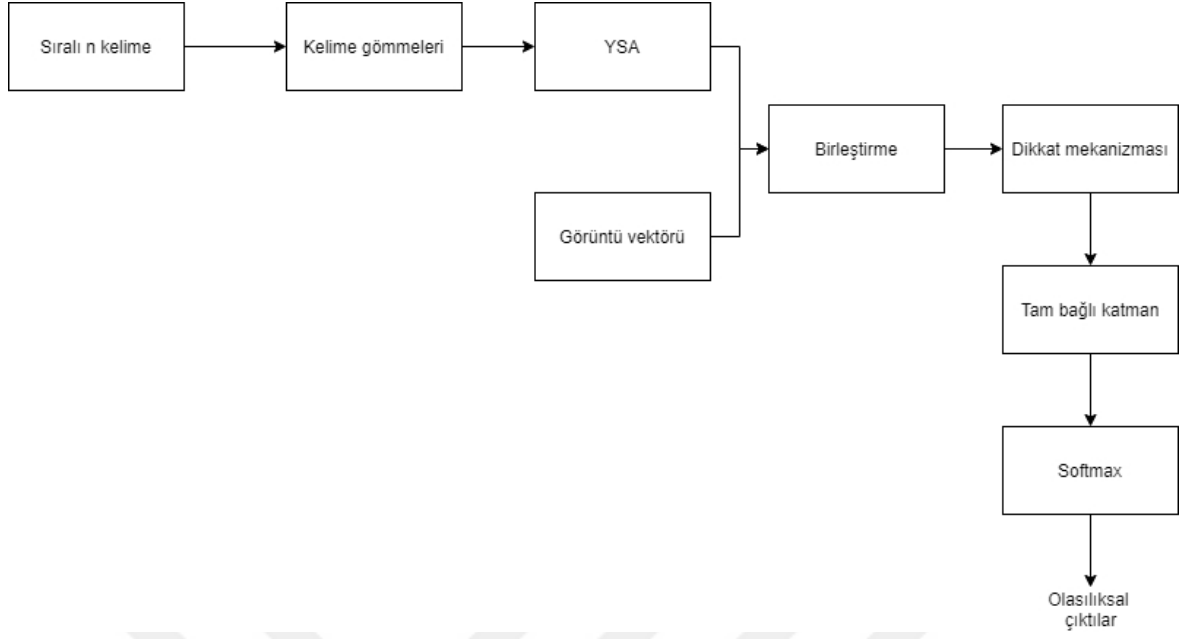
Bölüm 3.11.1'de anlatıldığı gibi görüntülerden elde edilen özellik vektörü ile kod çözücünün bir önceki durumundan hesaplanan yerleşim puanları hesaplanır. Devamında yerleşim puanlarının dikkat mekanizmasındaki ağırlıkları hesaplanır ve elde edilen değer ile görüntülerin özellik vektörü çarpılarak içerik vektörü elde edilir. Elde edilen içerik vektörü ile kelime gömmelerinin birleşimi YSA'ya girdi olur. YSA'nın çıktısı tam bağlı katmandan geçirilerek modelin çıktısı elde edilir. Şekil 4.7'de lokal dikkat mekanizması modeli gösterilmektedir.



Şekil 4.7. Lokal dikkat mekanizması modeli

4.3.7. Global dikkat mekanizması modeli

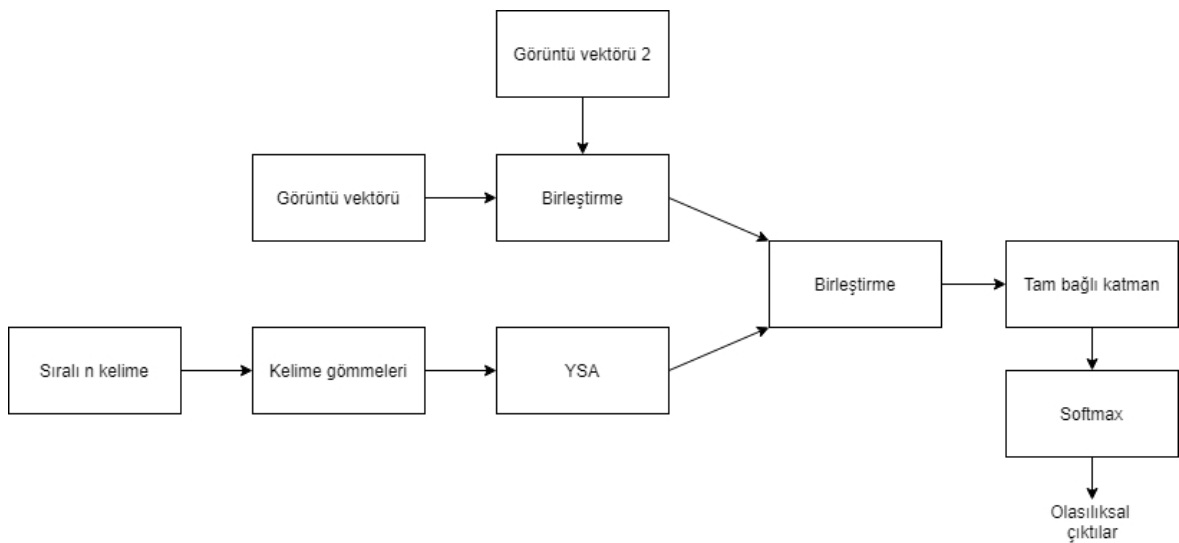
Global dikkat mekanizmasında lokal dikkat mekanizmasından farklı olarak YSA dikkat ağırlıkları üretilmeden evvel çalışır ve güncel gizli durum ile ağırlıklar hesaplanır. Bunun yanında gizli durum ile kodlayıcı çıktılarından elde edilen yerleşim puanı farklı yöntemle hesaplanabilir. Şekil 4.8’de global dikkat mekanizması modeli gösterilmektedir.



Şekil 4.8. Global dikkat mekanizması modeli

4.3.8. Topluluk 1 modeli

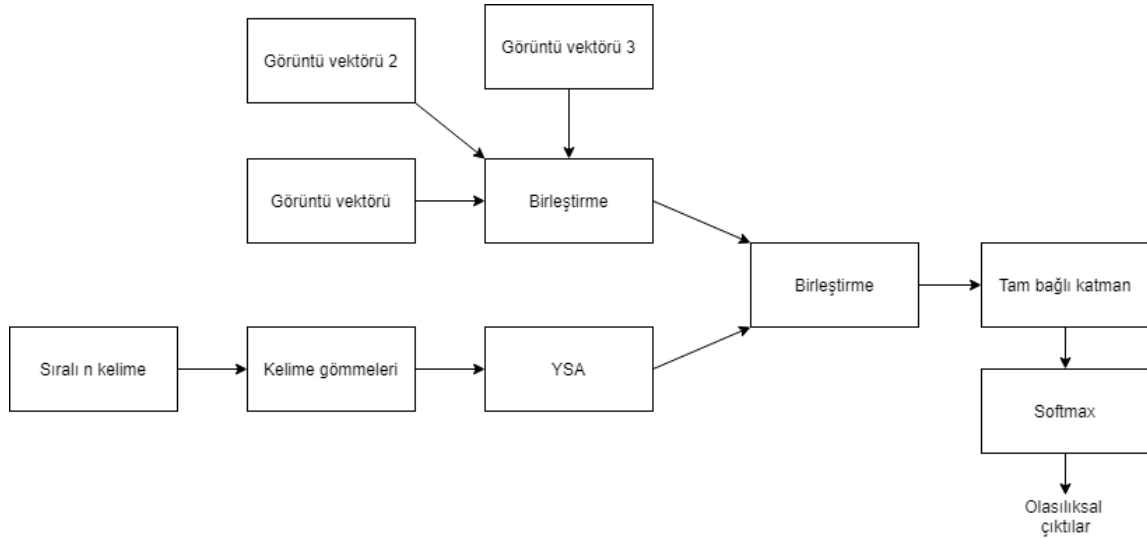
Topluluk modeliyle farklı ESA mimarilerinin beraber kullanımı sağlanmıştır. Bu bağlamda 2 farklı ESA'dan elde edilen özellik vektörü birleştirilmiş ve ardından kelime gömmeleriyle birleştirilerek tam bağlı katmandan geçirilerek modelin çıktısı elde edilmiştir. Şekil 4.9'da Topluluk 1 modeli gösterilmektedir.



Şekil 4.9. Topluluk 1 modeli

4.3.9. Topluluk 2 modeli

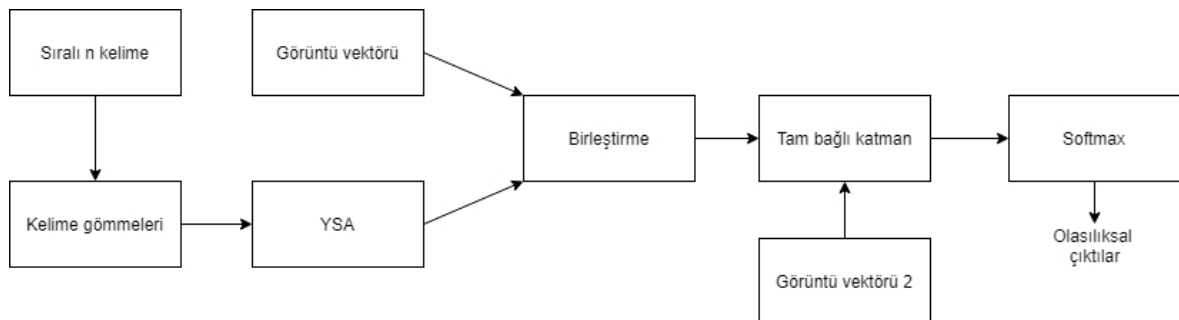
Topluluk 2 modeliyle topluluk 1 modelinden farklı olarak 3 farklı ESA'dan elde edilen özellik vektörü birleştirilmiştir. Devamında elde edilen özellik vektörü ile kelime gömmeleri birleştirilmiş ve tam bağlı katmandan geçirilerek modelin çıktısı elde edilmiştir. Şekil 4.10'da Topluluk 2 modeli gösterilmektedir.



Şekil 4.10. Topluluk 2 modeli

4.3.10. Topluluk 3 modeli

Tez çalışması dahilinde geliştirilen son model Topluluk 3 modelidir. Bu modelde 4.3.3 bölümünde anlatılan birleştirme modeli ile ESA beraber kullanılmıştır. Birleştirme modelinin en son bölümünde tam bağlı katmanda 2. evrişim ağından çıkarılan görüntülere ait özellik vektörü birleştirilmiş ve tam bağlı katmandan geçirilerek modelin çıktısı elde edilmiştir. Şekil 4.11'de Topluluk 3 modeli gösterilmektedir.

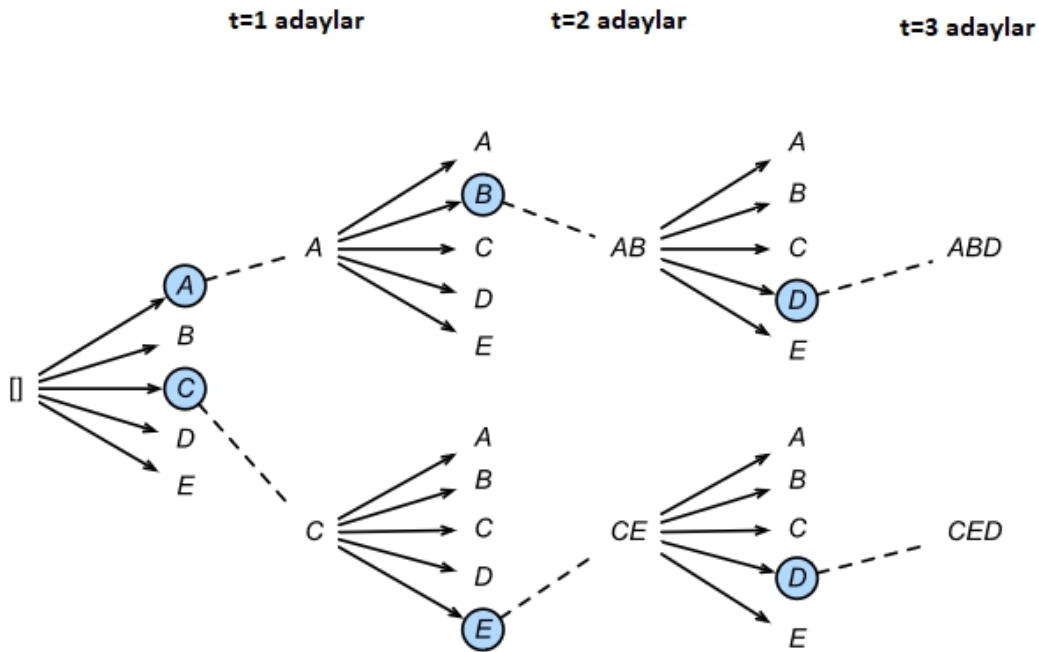


Şekil 4.11. Topluluk 3 modeli

4.4. Değerlendirme

Modelin çıktıları açgözlü arama (greedy search) veya ışın arama (beam search) algoritmaları kullanılarak tam bir cümle haline getirilir ve ardından değerlendirilir. Modelin en son katmanı olan softmax katmanı çıktı olarak bütün kelimelerin doğru kelime olma ihtimallerinin değerlerini döner. Açgözlü aramada en yüksek olasılığa sahip kelime çıktı olarak seçilir ve bir sonraki kelimeyi tahmin etmek için yeni girdiyi oluşturur. Bu süreç cümle tamamlanana kadar tekrarlanır. Açgözlü arama hızlıdır fakat elde edilen sonuç en iyi sonuç olmayabilir [118].

Işın arama algoritmasında ise açgözlü arama gibi en yüksek olasılığa sahip kelime seçilmez, bunun yerine en yüksek ihtimale sahip n adet kelime potansiyel cümleyi oluşturan kelime olarak değerlendirilir ve hafızada tutulur. Bir sonraki aşamada hafızadaki n adet kelime ayrı ayrı yeni girdi olur ve her bir kelime için model koşturulur. Sonuç olarak yine potansiyel n adet kelime her bir girdi için elde edilir. Yani, ikinci aşama sonunda $n*n$ adet 2 kelimeden oluşan aday kelime dizisi elde edilmiştir. Devamında 2 kelimeden oluşan dizilerin ayrı ayrı kümülatif olasılık değerleri hesaplanır ve en yüksek olasılık değerine sahip n tane kelime dizisi potansiyel olarak seçilir. Bu işlem cümle tamamlanıncaya kadar devam eder. Şekil 4.12’de ışın araması örneği gösterilmektedir.



Şekil 4.12. Işın araması [119]

Elde edilen başlıkların doğruluğunu ölçmek için BLEU skorlandırma yöntemi kullanılmıştır. BLEU [43] Papineni ve diğerleri tarafından 2002 yılında iki dil arasındaki otomatik çeviricilerin performanslarını ölçmek için geliştirilmiştir. BLEU algoritması tahmin edilen başlık ile gerçek başlık arasındaki yakınlığı hesaplar. Eğer tahmin ile gerçek cümle birebir aynı ise BLEU skoru 1 olurken, hiç benzerlik yoksa BLEU skoru 0 olur. BLEU skoru ile değerlendirme yöntemi son dönemde sıkça tartışılmaktadır ve yeni arayışlar vardır fakat mevcut çalışmaların ekseriyeti değerlendirmesini BLEU skoru ile yapmıştır. Bu bağlamda anlamlı bir karşılaştırma için BLEU skorlandırma yöntemi ile çıktıları değerlendirmek tercih edilmiştir. Bunun yanında BLEU algoritması, hızlı hesaplama yeteneği ve dilden bağımsız çalışabilmesi nedeniyle de tercih edilmiştir. BLEU algoritması skorlandırmayı aday başlık ile referans başlığın n-gramlarını sayarak yapmaktadır.



5. ARAŞTIRMA BULGULARI

5.1. Seçilen Teknolojiler ve Test Ortamı

Derin öğrenme uygulamalarının gelişimiyle birlikte modellerin oluşturulması ve koşturulmasını sağlayan birçok farklı araç ve teknik geliştirilmiştir. Geliştirilen teknolojiler arasından bu tez çalışmasında kullanılan araç, yapı ve kütüphaneler hakkında temel bilgi verilecektir.

5.1.1. Python

Kodlama kolaylığı ve yazılan kodun anlaşılabilirliği bakımından diğer dillere göre daha kolay olan Python programlama dili seçiminin temel motivasyonu, makine öğrenmesi ve derin öğrenme alanlarında oldukça yaygın kullanılan dil olmasındandır. Yaygın kullanım beraberinde daha çok geliştirilen yapılar ve kütüphaneler getirmektedir. Bununla birlikte yaygın kullanımdan kaynaklı olarak sorun yaşandığında çözüm bulmak adına topluluk desteği oldukça yüksek seviyededir.

5.1.2. Tensorflow

Tensorflow, Google araştırmacıları tarafından tasarlanmış, açık kaynak kodlu bir makine öğrenmesi platformudur [58]. Derin öğrenme modellerinin tasarımında kolaylık sağlayan Tensorflow'un seçiminin temel motivasyonu Keras kütüphanesi ile beraber çalışabilir olmasıdır.

5.1.3. Keras

Python dilinde yazılmış bir derin öğrenme kütüphanesi olan Keras [57], sunduğu üst seviye fonksiyonlar ve efektif kullanımı ile tercih edilmiştir. Bunun yanında derin öğrenme alanında son dönemde geliştirilmiş mimarileri ve yaklaşımları hızlı bir şekilde bünyesine katması ve bu yaklaşımları kolaylıkla kullanma olanağı sağlaması bakımından da Keras kütüphanesi tercih sebebi olmuştur. Bununla birlikte Keras topluluğunun oldukça aktif olması sebebiyle karşılaşılan sorunlara hızla yanıt alınabilmektedir. Örneğin tez çalışması

kapsamında karşılaştığım ve çözümünü bulamadığım Keras ile ilgili problemleri Stackoverflow isimli geliştirici platformunda [120] sorduğumda, oldukça kısa sürelerde yanıtlandığı gözlemlenmiştir. Keras kütüphanesinin yanında gelen bu gibi özellikler de Keras'ın seçiminde etkili olmuştur.

5.1.4. Test ortamı

Python dilinde, Keras kütüphanesi kullanılarak hazırlanan modellerin eğitimi ve testi, 16 GB RAM ve nVidia GTX 1060 ekran kartının olduğu bir sistemde tamamlanmıştır ve GPU'nun gücünden faydalanılmıştır.

5.2. Kullanılan Hiper Parametreler

Modellerin eğitimi sırasında kullanılan hiper parametreler Çizelge 5.1'de sunulmuştur. Öğrenme hızı ve mini grup boyutu ilk 10 turda ilk değerleri, ikinci 10 turda ise ikinci değerleri almışlardır. Kelime gömmelerinin boyutu ise kullanılan Word2vec modelinden elde edilen kelime gömmelerinin kaç boyutlu olarak temsil edileceğini ifade etmektedir.

Çizelge 5.1. Model eğitimi sırasında kullanılan hiper parametreler

Parametre	Değer
Öğrenme hızı	0.001, 0.0001
Mini grup boyutu	64, 128
Tur sayısı	10,10
Nöron sayısı	256
Optimizasyon algoritması	Adam
Kelime gömmelerinin boyutu	400
Seyreltme oranı	0.5
Aktivasyon fonksiyonu	ReLU

Hiper parametreler belirlenirken hiper parametre arama yöntemlerinden sezgisel arama ve ızgara araması beraber kullanılmıştır. Sezgisel arama, daha önceden benzer problemlerde kullanılan hiper parametrelerden ilham alarak bu hiper parametrelerin mevcut probleme dair öngörülere göre uyarlanıp uygulanmasıdır. Izgara araması (grid search) ise belirli aralıklar konularak hiper parametrenin aday değerleri ile modelin eğitilmesidir. Örneğin sözcük hazinesi oluşturmak için kullandığımız kelime sayısı eşik değeri için [0,10] aralığında, 2 birim kaydırarak 0,2,4,6,8,10 kelime eşik değerleri için sözcük hazineleri oluşturulmuş ve model eğitilmiştir.

Görüntülerin modele rastgele sıra ile girmesi ve sıralı girmesi arasında skorlarda farklılık olduğu gözlemlendiği için modellerin eğitiminde görüntüler rastgele sıra ile girdi olmuşlardır. Bunun yanında önceki bölümde bahsedildiği gibi kelimeleri köklerine ayırarak ortak köklere sahip kelimeler üzerinden kelimenin veri kümesinde kaç kere geçtiğini belirlemek ile hiç köklere ayırmadan kaç kere geçtiğini belirlemek arasında skorlarda farklılık olduğu gözlemlendiği için modellerin eğitimi her iki şekilde de gerçekleştirilmiştir. Son olarak yine önceki bölümde bahsedildiği gibi kelimelerin vektörel gösterimleri bulunamadığında kelimeleri köklerine ayırarak vektörel gösterimlerini çıkarmak ile köklere ayırmadan sadece vektörel gösterimleri olan kelimelerle çalışmak arasında skorlarda farklılık olduğu gözlemlendiği için modellerin eğitimi her iki şekilde de gerçekleştirilmiştir.

5.3. Sonuçlar

Birleştirme modeli ile elde edilen BLEU skorları Çizelge 5.2’de gösterilmektedir. Çizelge 5.2, 5.3, 5.4, 5.5’de köke ayırma bölümünde 1 sayısı hiç bir şekilde köke ayırma işlemi uygulanmadığını, 2 sayısı kelime gömmeleri bulunamadığında köke ayırma işleminin uygulandığını, 3 sayısı kelimenin orjinal halinin kelime gömmesi olduğu halde, sözcük hazinesinde eşik değerin altında yer aldığını fakat kelimenin kökünden türeyen kelimelerin toplam sayısının eşik değerin üstünde yer aldığı için kelimenin sözcük hazinesine dahil edildiğini, 4 sayısı ise 3’de yapılan işlemin kelime kökü kullanılarak yapıldığını ifade etmektedir.

Çizelge 5.2. Birleştirme modeli ile elde edilen sonuçlar

ESA	YSA	Arama	Köke ayırma	BLUE-1 skoru
Inception v3	UKSB	açgözlü	1	0.360
Inception v3	UKSB	açgözlü	2	0.376
Inception v3	UKSB	açgözlü	3	0.362
Inception v3	UKSB	açgözlü	4	0.375
Inception v3	UKSB	ışın	2	0.383
Inception v3	KYB	ışın	2	0.394
Inception v3	YUKSB	açgözlü	2	0.365
VGG16	UKSB	açgözlü	2	0.344
VGG19	UKSB	açgözlü	2	0.347
MobileNet	UKSB	açgözlü	2	0.355
ResNet	UKSB	açgözlü	2	0.365
ResNet v2	UKSB	açgözlü	2	0.345
EfficientNet	UKSB	açgözlü	2	0.287
IncResNetv2	UKSB	açgözlü	2	0.374
Xception	UKSB	açgözlü	2	0.379

Elde edilen sonuçlar incelendiğinde ESA mimarileri arasında Inception modülü kullanan mimarilerin diğerlerine göre daha iyi sonuçlar verdiği gözlemlenmiştir. Bunun yanında YSA'da KYB'nin en iyi sonuçlar verdiği, ardından UKSB'nin geldiği ve iki yönlü UKSB'nin en son sırada yer aldığı gözlemlenmiştir. Kelimeyi köklerine ayırarak modeli eğitmenin ise ufakta olsa bir etkisi olmakla birlikte beklenen seviyede bir artış olmamıştır. Son olarak arama yönteminde ışın aramasının açgözlü aramaya göre daha iyi sonuçlar verdiği gözlemlenmiştir. Fakat burada da beklenen sonuçlar ışın aramasının çok daha iyi sonuçlara sahip olacağı yönündeydi.

Alternatif birleştirme, enjeksiyon, alternatif enjeksiyon ve BirEn modelleri ile elde edilen BLEU skorları Çizelge 5.3’de gösterilmektedir.

Çizelge 5.3. Alternatif birleştirme, enjeksiyon, alternatif enjeksiyon ve BirEn modelleri ile elde edilen BLEU skorları

Model	ESA	YSA	Arama	Kök kullanımı	BLUE-1 skoru
Alternatif birleştirme	Inception v3	UKSB	Açgözlü	2	0.374
Alternatif birleştirme	Xception	KYB	Açgözlü	2	0.372
Alternatif birleştirme	VGG19	İYUKSB	Açgözlü	3	0.355
Alternatif birleştirme	IncResNetv2	KYB	Işın	4	0.378
Enjeksiyon	Inception v3	UKSB	Açgözlü	1	0.346
Enjeksiyon	Inception v3	UKSB	Açgözlü	2	0.355
Enjeksiyon	ResNet	İYUKSB	Açgözlü	1	0.338
Enjeksiyon	MobileNet	KYB	Işın	2	0.256
Alternatif enjeksiyon	Inception v3	UKSB	Açgözlü	2	0.348
Alternatif enjeksiyon	Xception	KYB	Açgözlü	1	0.357
Alternatif enjeksiyon	IncResNetv2	İYUKSB	Açgözlü	3	0.335
Alternatif enjeksiyon	ResNet v2	UKSB	Işın	4	0.338
BirEn	Inception v3	UKSB	Açgözlü	2	0.366
BirEn	VGG19	İYUKSB	Açgözlü	1	0.349
BirEn	ResNet	İYUKSB	Işın	3	0.36

Yapılan çalışmalar sonucunda enjeksiyon modeli ve türevi modellerin genel olarak birleştirme modeli ve türevi modellere göre daha az başarılı olduğu gözlemlenmiştir. Modellerin kıyaslanabilmesi adına bütün modellerde bir kere ESA olarak inception v3,

UKSB, açgözlü arama ve köke ayırma olarak 2. seçenek seçilerek modeller eğitilmiştir. Buradan çıkan sonuçlara göre birleştirme modeli de dahil edildiğinde en başarılı model birleştirme modeli, ardından alternatif birleştirme, BirEn, alternatif enjeksiyon ve enjeksiyon modelleri gelmiştir.

Şekil 5.1’de veri kümesindeki görüntüler ve elde edilen açıklamalar gösterilmektedir. Elde edilen açıklamalar incelendiğinde başarısız açıklamaların dahi cümle yapısı bağlamında ve açıklamaların kelime sayısı bağlamında başarısız olmadığı söylenebilir. Bunun yanında yine başarısız çıktılar dahi mantıksız sayılabilecek tarzda değildir. Buradan çıkarılabilecek sonuç veri kümesinin küçüklüğü sebebiyle modellerin yeterince eğitilememiştir.





iki köpek karlar üzerinde
oynuyor



kafasına koruyucu
kask giymiş bir



spor salonunda basketbol
oynayan gençler



kaya tırmanışı yapan bir
adam



sokakta dans eden
afrika asıllı çocuklar



iki köpek çimler üzerinde
koşuyor



beyaz tişörtlü bir adam bir
kayanın önünde duruyor



bir adam ve bir kadın
sokakta yürüyor



kaykay ile artistik
hareketler yapan bir genç

Şekil 5.1. TasvirEt veri kümesi görüntülerinden oluşturulan açıklamalar (üst satır başarılı çıktıları, orta satır orta seviye çıktıları, alt satır ise başarısız çıktıları göstermektedir)

Dikkat mekanizması kullanılarak geliştirilen modeller elde edilen sonuçlar Çizelge 5.4'de gösterilmektedir.

Çizelge 5.4. Dikkat mekanizması kullanılarak geliştirilen modeller elde edilen sonuçlar

Model	ESA	YSA	Arama	Kök kullanımı	BLUE-1 skoru
Lokal dikkat mekanizması	Inception v3	UKSB	Açgözlü	2	0.410
Lokal dikkat mekanizması	VGG16	KYB	Açgözlü	1	0.383
Lokal dikkat mekanizması	ResNet	İYUKSB	Işın	3	0.394
Global dikkat mekanizması	Inception v3	UKSB	Açgözlü	2	0.407
Global dikkat mekanizması	Inception v3	KYB	Işın	2	0.424
Global dikkat mekanizması	VGG19	İYUKSB	Açgözlü	1	0.372

Bütün modeller arasında dikkat mekanizması kullanarak geliştirilen modeller en yüksek skorları elde etmiştir. Aynı zamanda bu skorlar tüm zamanlarda görüntülere başlık atama alanında Türkçe çalışmalar arasındaki en yüksek skorlar olmuştur. Buradan dikkat mekanizmasının görüntülere başlık atama problemi ile uyumlu olduğu çıkarımı yapılabilir. Bunun yanında diğer modellerdeki sonuçlara paralel olarak en başarılı sonuçlar Inception v3 mimarisi kullanılarak çıkarılan görüntü özellik vektörü ile elde edilmiştir.

Topluluk öğrenmesi kullanılarak geliştirilen modeller ile elde edilen sonuçlar Çizelge 5.5’de gösterilmektedir.

Çizelge 5.5. Topluluk öğrenmesi kullanılarak geliştirilen modeller ile elde edilen sonuçlar

Model	ESA	YSA	Arama	Kök kullanımı	BLUE-1 skoru
Topluluk 1	Inception v3, resNet	UKSB	Açgözlü	2	0.372
Topluluk 1	IncResNetv2, Xception	UKSB	Işın	4	0.401
Topluluk 1	EfficientNet, Inception v3,	İYUK SB	Açgözlü	2	0.369
Topluluk 2	Inception v3, resNet	UKSB	Açgözlü	2	0.360
Topluluk 2	Xception, resNet	KYB	Açgözlü	1	0.356
Topluluk 3	Inception v3, Xception, resNet	UKSB	Açgözlü	2	0.395
Topluluk 3	EfficientNet, IncResNetv2, mobileNet	KYB	Işın	1	0.364

Topluluk öğrenmesi kullanılarak eğitilen modellerden elde edilen skorlar dikkat mekanizması kullanarak eğitilen modeller ile elde edilen skorlar kadar yüksek elde olmasa da Inception modülü kullanılarak elde edilen ESA mimarilerinin diğer mimarilere göre daha yüksek sonuç verdiği gözlemlenmiştir.

Tez çalışması dahilinde elde edilen en iyi sonuçlar ve görüntülere başlık atama konusunda Türkçe dilindeki diğer çalışmalar Çizelge 5.6'da mevcuttur.

Çizelge 5.6. Karşılaştırmalı sonuçlar

Çalışma	Kullanılan Veri Kümesi	BLUE-1 skoru
TasvirEt (Yöntem 2) [39]	TasvirEt	0.260
Oto. Tercüme (M3) [40]	MS-COCO	0.342
Oto. Tercüme (F3) [40]	Flickr30K	0.335
Yılmaz [121]	MS-COCO	0.288
Altsözcük (Yöntem 1) [47]	MS-COCO	0.293
Altsözcük (Yöntem 2) [47]	Flickr30K	0.308
Inception v3+UKSB+ ışın [5]	TasvirEt	0.400
Topluluk 1 (IncResNetv2+ Xception, UKSB, ışın araması)	TasvirEt	0.401
Global dikkat mek. (Inception v3, KYB, ışın araması)	TasvirEt	0.424
Lokal dikkat mek. (Inception v3, KYB, açgözlü arama)	TasvirEt	0.410

Bu çalışmada bir önceki çalışmamızda [5] elde edilen 0.394'lük BLUE-1 skoruna göre daha iyi bir sonuç elde edilmiştir. Literatürdeki Türkçe dilinde görüntülere başlık atama çalışmaları ile karşılaştırma yapıldığında bizim çalışmamızın sahip olduğu handikaplara rağmen elde edilen skorların büyük bir çoğunluğunun literatürdeki çalışmalardaki skarlardan daha yüksek olduğu gözlemlenmiştir. Çalışmamızın en büyük handikapı veri kümesinin boyutunun küçüklüğüdür. Bu sebeple sadece 6000 görüntü ile eğitim tamamlanırken diğer çalışmalarda eğitim aşamasında 30.000 veya 80.000 görüntü kullanılmıştır. Bunun yanında TasvirEt veri kümesinin oluşturulmasında kitle kaynak yöntemi kullanıldığı için veri kümesindeki tanımlamalar belirli bir desene sahip değildir. Fakat diğer çalışmalarda veri kümesi Google Translate API kullanılarak Türkçeleştirilmiştir. Yani veri kümesi önceden eğitilmiş bir ağı çıktısı olduğu için doğal

değildir. Bizim çalışmamızda bahsedilen handikapları aşmak adına MS-COCO ve Flick30K veri kümeleri Translate API ile Türkçeye çevirerek kullanılmamıştır. Bunun temel sebebi önceden eğitilmiş bir ağın çıktısı Türkçe veri kümesi olarak nitelendirilemez.





6. SONUÇ VE ÖNERİLER

Evrişimli ve yinelemeli sinir ağları kullanarak görüntülere başlık atama çalışmaları yeni gelişen bir alan olduğu halde derin öğrenmenin dominasyonu sayesinde geliştirilen teknikler ve bu alana yakın alanlardaki çalışmalarda kullanılan tekniklerin uyarlanması sayesinde başarımlar oranları hızla artmıştır. Fakat 2021'in ilk ayları itibariyle bu alandaki en olgun döneme girildiğini söylemek için halen erken olabilir. Muhtemelen gelecek dönemde yapılacak çalışmalarda daha iyi sonuçlar elde edilebilecektir. Özellikle dikkat mekanizması kullanarak geliştirilen çözümlerin görüntülere başlık atanması problemine uyumu düşünüldüğünde, dikkat mekanizmasının gelişimi ile birlikte bu alanda daha iyi sonuçlar alınabileceği söylenebilir.

Cermen dil ailesindeki dillerde görüntülere başlık oluşturma alanında birçok çalışma vardır ve bu çalışmalarda belirli bir olgunluk seviyesine ulaşılmıştır. Fakat Türkçe gibi sondan eklemeli dillerde hem diğer dillerdeki kadar çalışma yapılmamıştır hem de yapılan çalışmalarda diğer dillerdeki seviyede bir başarı gözlemlenmemiştir. Yapılan tez çalışmasının temel motivasyonu da görüntülere başlık atanması konusunda kendi ana dilimizde, hali hazırda geliştirilmiş modelleri ve bu tez çalışması kapsamında geliştirilmiş modelleri kullanarak bir çalışma yapmak olmuştur. Çalışma kapsamında ESA ve YSA, kodlayıcı-kod çözücü mimarisini kullanan modeller ve dikkat mekanizması kullanarak kurulan modeller içinde kullanılmıştır. Geliştirilen modeller, Türkçe veri kümesi üzerinde Türkçeye özgü ön işleme ve dönüşüm yapıldıktan sonra eğitilmiş ve çıkarılan sonuçlar ağgözlü arama ve ışın araması yoluyla cümle haline getirilmiştir. Modelin başarısı BLEU skor tekniği ile ölçülmüştür.

Elde edilen sonuçlar literatürdeki Türkçe görüntülere başlık ataması çalışmalarından daha iyi sonuçlar vermiş olsa da diğer dillere nazaran daha düşük sonuçlar elde edilmiştir. Bunun ilk sebebi kullanılan veri kümesindeki tanım sayısının az olmasındandır. Diğer dillerde dahi eğitim için 100.000'lerle ifade edilen sayılarda tanımlamalar kullanılırken sondan eklemeli yapıda olan Türkçe dilinde diğer dillere nazaran çok daha büyük boyutlu bir veri kümesi ihtiyacı vardır. Bunun yanında daha önemli bir sebep ise mevcut modellerin tasarımı genel olarak Cermen dil ailesi içindeki dillere uygundur. Bu sebeple, bu diller üzerinde en iyi sonuçları veren modeller ile dahi sondan eklemeli yapıda olan

Türkçe dilinde İngilizce dilindeki gibi bir başarı elde edilmemektedir. Bunu yenmek için ekleri atarak sadece kökler ile çalışma yapıldığında da elde edilen sonuçlar da tatmin edici olmamıştır.

Buradan çıkarılacak sonuç ve gelecek çalışmalarda yapılması gereken, Türkçenin sondan eklemeli yapısına uygun olarak metinlerin biçim bilimsel ve söz dizimsel çözümlemelerini matematiksel bir hale getirip, çıkarılan biçim bilimsel ve söz dizimsel çözümlemelerin girdi olabileceği bir model geliştirerek Türkçe dilinde daha yüksek başarı oranlarına ulaşmaktır.



KAYNAKLAR

1. İnternet: Clement, J. Daily social media usage worldwide. URL: <http://www.statista.com/statistics/433871/daily-social-media-usage-worldwide/>. Son Erişim Tarihi: 15.11.2020.
2. İnternet: What Is Instagram? Instagram Help Center. URL: help.instagram.com/424737657584573. Son Erişim Tarihi: 15.11.2020.
3. İnternet: Mind-Boggling Instagram Stats & Facts for 2018. URL: www.wordstream.com/blog/ws/2017/04/20/instagram-statistics. Son Erişim Tarihi: 15.11.2020.
4. İnternet: Instagram Shopping: Sell Products Online. URL: www.facebook.com/business/instagram/shopping. Son Erişim Tarihi: 15.11.2020.
5. Kunt, T. ve Barışçı, N. (2020, 2-4 Ekim). *Image Captioning in Turkish Language*. Proceedings of International Conference on Advanced Materials Science & Engineering and High Tech Device Applications, Ankara, 54-58.
6. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., and Fei-Fei, L. (2015). Imagenet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), 211-252.
7. Gurari, D., Zhao, Y., Zhang, M., and Bhattacharya, N. (2020, 24-28 August). *Captioning Images Taken by People Who Are Blind*. European Conference on Computer Vision, Glasgow, 417-434.
8. İnternet: Image Captioning as an Assistive Technology. URL: www.ibm.com/blogs/research/2020/07/image-captioning-assistive-technology. Son Erişim Tarihi: 18.11.2020.
9. İnternet: Beckett, J. Wearable Device for Blind People Could Be a Life Changer: NVIDIA Blog. URL: blogs.nvidia.com/blog/2016/10/27/wearable-device-for-blind-visually-impaired. Son Erişim Tarihi: 19.11.2020.
10. İnternet: Lamba, H. Image Captioning with Keras-‘Teaching Computers to Describe Pictures. URL: towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-describe-pictures-c88a46a311b8. Son Erişim Tarihi: 20.11.2020.
11. Pervan, N., (2019). *Derin Öğrenme Yaklaşımları Kullanarak Türkçe Metinlerden Anlamsal Çıkarım Yapma*, Doktora Tezi, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 1-5.
12. Yıldırım, E., Çetin, F. S., Eryiğit, G., ve Temel, T. (2015). The Impact of NLP on Turkish Sentiment Analysis. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri Ve Mühendisliği Dergisi*, 7(1).
13. Kvita, B. J. (2016). *Image Captioning with Recurrent Neural Networks*, Master’s Thesis, Brno University of Technology Faculty of Technology, Brno, 15-17.

14. Bai, S., and An, S. (2018). A Survey on Automatic Image Caption Generation. *Neurocomputing*, 311, 291-304.
15. İnternet: Eddy, S. What Is a Hidden Markov Model?. URL: www.nature.com/articles/nbt1004-1315. Son Erişim Tarihi: 21.11.2020.
16. Yang, Y., Teo, C., Daumé III, H., ve Aloimonos, Y. (2011, July). *Corpus-Guided Sentence Generation of Natural Images*. Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, 444-454.
17. Mitchell, M., Dodge, J., Goyal, A., Yamaguchi, K., Stratos, K., Han, X., and Daumé III, H. (2012, April). *Midge: Generating Image Descriptions from Computer Vision Detections*. Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, 747-756.
18. Kuznetsova, P., Ordonez, V., Berg, T. L., and Choi, Y. (2014). Treetalk: Composition and Compression of Trees for Image Descriptions. *Transactions of the Association for Computational Linguistics*, 2, 351-362.
19. Gupta, S., and Malik, J. (2015). Visual Semantic Role Labeling. *Computer Vision and Pattern Recognition*, 2-3.
20. Khojasteh, H., Ensari, E., Razzaghi, P. (2019), *Automatic Image Description Generation Using Deep Multimodal Embeddings*, Master's Thesis, Institute for Advanced Studies in Basic Sciences, Zanjan, Iran, 8-9.
21. Ordonez, V., Kulkarni, G., and Berg, T. (2011). Im2text: Describing Images Using 1 Million Captioned Photographs. *Advances in Neural Information Processing Systems*, 24, 1143-1151.
22. Farhadi, A., Hejrati, M., Sadeghi, M. A., Young, P., Rashtchian, C., Hockenmaier, J., and Forsyth, D. (2010, September). *Every Picture Tells a Story: Generating Sentences from Images*. European Conference on Computer Vision, Berlin, 15-29.
23. Li, S., Kulkarni, G., Berg, T., Berg, A., and Choi, Y. (2011, June). *Composing Simple Image Descriptions Using Web-Scale n-grams*. Proceedings of the Fifteenth Conference on Computational Natural Language Learning, Portland, 220-228.
24. Mason, R., and Charniak, E. (2014, June). *Nonparametric Method For Data-Driven Image Captioning*. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, United States of America, 592-598.
25. Karpathy, A. (2016). *Connecting images and natural language*, Doctoral Dissertation, Stanford University Department of Computer Science, California, 1-8.
26. Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). *Rich feature Hierarchies for Accurate Object Detection and Semantic Segmentation*. Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition, Columbus, United States of America, 580-587.
27. Kiros, R., Salakhutdinov, R., and Zemel, R. (2014, January). *Multimodal Neural Language Models*. International Conference on Machine Learning, Beijing, 595-603.

28. Devlin, J., Cheng, H., Fang, H., Gupta, S., Deng, L., He, X., and Mitchell, M. (2015). *Language Models for Image Captioning: The Quirks and What Works*. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, Beijing, 100-105.
29. Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). *Show And Tell: A Neural Image Caption Generator*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, 3156-3164.
30. Gal, Y., and Ghahramani, Z. (2016, June). *Dropout as a bayesian approximation: Representing model uncertainty in deep learning*. International Conference on Machine Learning, Lille, France, 1050-1059.
31. Saad, D. (1998). Online algorithms and stochastic approximations. *Online Learning*, 5, 6-3.
32. Medress, M. F., Cooper, F. S., Forgie, J. W., Green, C. C., Klatt, D. H., O'Malley, M. H., and Woods, W. A. (1977). Speech understanding systems: Report of a steering committee. *Artificial Intelligence*, 9(3), 307-316.
33. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., and Bengio, Y. (2015, June). *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. International Conference On Machine Learning, Lille, France, 2048-2057.
34. İnternet: Flickr 8k Dataset. URL: www.kaggle.com/adityajn105/flickr8k/activity. Son Erişim Tarihi: 24.11.2020.
35. İnternet: Cs.toronto.edu.tr. URL: cs.toronto.edu/~tijmen/csc321/slides/lecture%5C_slides%5C_lec6.pdf. Son Erişim Tarihi: 24.11.2020.
36. İnternet: Flickr Image Dataset. URL: www.kaggle.com/hsankesara/flickr-image-dataset. Son Erişim Tarihi: 26.11.2020.
37. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., and Zitnick, C. L. (2014, September). *Microsoft Coco: Common Objects in Context*. European Conference on Computer Vision, Zurich, 740-755.
38. Kingma, D. P., and Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. International Conference on Learning Representations (ICLR), San Diego, United States of America, 8-9.
39. Unal, M. E., Citamak, B., Yagcioglu, S., Erdem, A., Erdem, E., Cinbis, N. I., ve Cakici, R. (2016, Mayıs). *Tasviret: Görüntülerden Otomatik Türkçe Açıklama Oluşturma İçin Bir Denektaş Veri Kümesi*. 2016 24th Signal Processing and Communication Application Conference (SIU), Zonguldak, 1977-1980.
40. Samet, N., Hiçsönmez, S., Duygulu, P., ve Akbas, E. (2017). *Görüntü Altyazılama için Otomatik Tercümeyle Eğitim Kümesi Olusturulabilir mi?*. 25th Signal Processing and Communications Applications Conference (SIU), Antalya, 4-6.

41. Simonyan, K., and Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. International Conference on Learning Representations, Banff, Canada, 3-4.
42. İnternet: Cloud Translation Documentation. URL: cloud.google.com/translate/docs. Son Erişim Tarihi: 28.11.2020.
43. Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. (2002, July). *BLEU: a Method for Automatic Evaluation of Machine Translation*. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, 311-318.
44. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). *Distributed Representations of Words and Phrases and Their Compositionality*. Advances in Neural Information Processing Systems, Lake Thae, United States of America, 3111-3119.
45. Lin, C. Y. (2004, June). *Rouge: A Package for Automatic Evaluation of Summaries*. Association for Computational Linguistics, Barcelona, 74-81.
46. Vedantam, Ramakrishna, C. Lawrence Zitnick, and Devi Parikh. (2015). *Cider: Consensus-based Image Description Evaluation*. Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Boston, 4566-4575.
47. Kuyu, M., Erdem, A., ve Erdem, E. (2018, Mayıs). *Image Captioning in Turkish with Subword Units*. 2018 26th Signal Processing and Communications Applications Conference (SIU), İzmir, 1-4.
48. Gage, P. (1994). A New Algorithm for Data Compression. *C Users Journal*, 12(2), 23-38.
49. Alpaydin, E. (2020). *Introduction to Machine Learning*. MIT press, 20-40.
50. Oruganti, R. M. and Ptucha, R. (2016). *Image Description Using Deep Neural Networks*, Master's Thesis, Rochester Institute of Technology , New York, 36-38.
51. Rosenblatt, F. (1960). Perceptron Simulation Experiments. *Proceedings of the IRE*, 48(3), 301-309.
52. Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
53. İnternet: A Quick Introduction to Neural Networks. URL: ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/. Son Erişim Tarihi: 02.12.2020.
54. İnternet: Artificial Neural Network. URL: en.wikipedia.org/w/index.php?title=Artificial_neural_network. Son Erişim Tarihi: 02.12.2020.
55. Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.

56. Ezel, E. (2018). *Derin Öğrenme Yöntemi Kullanılarak Görüntü Tabanlı Türk İşaret Dili Tanıma*, Doktora Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya, 17-21.
57. İnternet: Keras Documentation: About Keras. URL: keras.io/about/. Son Erişim Tarihi: 03.12.2020.
58. İnternet: Why TensorFlow. URL: www.tensorflow.org/about. Son Erişim Tarihi: 03.12.2020.
59. İnternet: Scientific Computing for LuaJIT. URL: torch.ch. Son Erişim Tarihi: 03.12.2020.
60. İnternet: *Caffe*. URL: caffe.berkeleyvision.org. Son Erişim Tarihi: 03.12.2020.
61. Karabaş, A. ve Diri, B. (2019). *Türkçe Mikrobloglarda İroni Tespiti*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 16-19.
62. Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. International Conference on Learning Representations, San Diego, 1-12.
63. Karl, F., Lauri, M., and Biemann, C. (2019, 9-11 October). *Creating Information-maximizing Natural Language Messages Through Image Captioning-Retrieval*. Konferenz zur Verarbeitung Natürlicher Sprache, Nürnberg, Germany, 3-4.
64. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). *Distributed Representations of Words and Phrases and Their Compositionality*. In Neural Information Processing Systems, Lake Tahoe, United States of America, 3111-3119.
65. İnternet: Word2vec Architecture. URL: http://Mccormickml.com/Assets/word2vec/skip_gram_net_arch.Png. Son Erişim Tarihi: 07.12.2020.
66. Ma, L., and Zhang, Y. (2015, October). *Using Word2Vec to Process Big Text Data*. IEEE International Conference on Big Data, Santa Clara, United States of America, 2895-2897.
67. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4), 541-551.
68. Güven, F. (2019). *Using Text Representation And Deep Learning Methods For Turkish Text Classification*, Yüksek Lisans Tezi, Çukurova Üniversitesi Fen Bilimleri Enstitüsü, Adana, 23-24.
69. İnternet: A Comprehensive Guide to Convolutional Neural Networks. URL: towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. Son Erişim Tarihi: 09.12.2020.

70. Kim, Y. (2014, October). *Convolutional Neural Networks For Sentence Classification*. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 1746-1751.
71. İnternet: Convolutional Network. URL: <http://www.deeplearningessentials.science/ConvolutionalNetwork/>. Son Erişim Tarihi: 11.12.2020.
72. İnternet: Derine Daha Derine: Evrişimli Sinir Ağları. URL: ayyucekizrak.medium.com/deri%CC%87ne-daha-deri%CC%87ne-evri%C5%9Fimli-sinir-a%C4%9Flar%C4%B1-2813a2c8b2a9. Son Erişim Tarihi: 12.12.2020.
73. İnternet: What Is Padding in Convolutional Neural Network's(CNN's) Padding. URL: ayeshmanthaperera.medium.com/what-is-padding-in-cnns-71b21fb0dd7. Son Erişim Tarihi: 12.12.2020.
74. İnternet: Understanding of Convolutional Neural Network (CNN) - Deep Learning. URL: medium.com/@RaghavPrabhu/understanding-of-convolutional-neuralnetwork-cnn-deep-learning-99760835f148. Son Erişim Tarihi: 13.12.2020.
75. İnternet: ImageNet. URL: image-net.org/about-overview. Son Erişim Tarihi: 13.12.2020.
76. İnternet: Large Scale Visual Recognition Challenge (ILSVRC). URL: www.image-net.org/challenges/LSVRC. Son Erişim Tarihi: 13.12.2020.
77. Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, 60(6), 84-90.
78. Ulusoy, O., Akgül, C. B., and Anarım, E. (2019). *Improving Image Captioning with Language Modeling Regularizations*. 2019 Innovations in Intelligent Systems and Applications Conference (ASYU), İzmir, 1-6.
79. İnternet: Evrişimli Sinir Ağını (CNN) Anlama - Derin Öğrenme. URL: <https://ichi.pro/tr/evrisimli-sinir-agini-cnn-anlama-derin-ogrenme-168732222943560>. Son Erişim Tarihi: 13.12.2020.
80. İnternet: Building VGG19 with Keras. URL: saicharanars.medium.com/building-vgg19-with-keras-f516101c24cf. Son Erişim Tarihi: 13.12.2020.
81. He, K., Zhang, X., Ren, S., and Sun, J. (2016). *Deep Residual Learning for Image Recognition*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 770-778.
82. İnternet: An Overview of ResNet and Its Variants. URL: towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035. Son Erişim Tarihi: 14.12.2020.
83. Van Hieu, N., and Hien, N. L. H. (2020). Recognition of Plant Species using Deep Convolutional Feature Extraction. *International Journal on Emerging Technologies*, 11(3), 904-910.

84. İnternet: Deep Learning Using Python + Keras (Chapter 3): ResNet. URL: glinksolutions.vn/2018/06/20/deep-learning-using-python-keras-chapter-3-resnet. Son Erişim Tarihi: 15.12.2020.
85. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). *Rethinking the Inception Architecture for Computer Vision*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2818-2826.
86. Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2017). *Inception-v4, Inception-resnet and the Impact of Residual Connections on Learning*. Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, 4278–4284.
87. İnternet: Improving Inception and Image Classification in TensorFlow. URL: ai.googleblog.com/2016/08/improving-inception-and-image.html. Son Erişim Tarihi: 16.12.2020.
88. Chollet, F. (2017). *Xception: Deep Learning with Depthwise Separable Convolutions*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, United States of America, 1251-1258.
89. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., and Adam, H. (2017). Mobilenets: Efficient convolutional Neural Networks for Mobile Vision Applications. *Computing Research Repository*, 2-6.
90. İnternet: Image Classification With MobileNet. URL: medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470 Son Erişim Tarihi: 17.12.2020.
91. Tan, M., and Le, Q. V. (2019). *Efficientnet: Rethinking model scaling for convolutional neural networks*. Proceedings of the 36th International Conference on Machine Learning, Long Beach, United States of America, 6105-6114.
92. Vijayaraju, N. (2019). *Image Retrieval Using Image Captioning*, Master's Thesis, San Jose State University Faculty of the Department of Computer Science, San Jose, United States of America, 13-14.
93. İnternet: DeepAI. Recurrent Neural Network. URL: <https://deepai.org/machine-learning-glossary-and-terms/recurrent-neural-network> Son Erişim Tarihi: 17.12.2020.
94. Ismail, A. A., Wood, T., and Bravo, H. C. (2018). Improving Long-Horizon Forecasts with Expectation-Biased LSTM Networks. *Computing Research Repository*, 4-5.
95. Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. NIPS 2014 Workshop on Deep Learning, New York, 4-5.
96. İnternet: Recurrent Neural Network. URL: sqlml.azurewebsites.net/2017/08/12/recurrent-neural-network Son Erişim Tarihi: 18.12.2020.
97. İnternet: Bi-LSTM. URL: medium.com/@raghavaggarwal0089/bi-lstm-bc3d68da8bd0 Son Erişim Tarihi: 19.12.2020.

98. Harma, S. B. (2020). *Yapay Sinir Ağları ile Makine Çevirisinin Detaylı Başarım Analizi*, Yüksek Lisans Tezi, TOBB ETÜ Fen Bilimleri Enstitüsü, Ankara, 10-16.
99. Tanti, M., Gatt, A., and Camilleri, K. P. (2017). Where to Put the Image in an Image Caption Generator. *Natural Language Engineering*, 24(3), 467-489.
100. Spratling, M. W., and Johnson, M. H. (2004). A Feedback Model of Visual Attention. *Journal of Cognitive Neuroscience*, 16(2), 219-237.
101. Tu, G. (2020). *Image Captioning On General Data And Fashion Data: An Attribute-Image-Combined Attention-Based Network for Image Captioning on Mutli-Object Images and Single-Object Images*, Master's Thesis, KTH Royal Institute of Technology School of Electrical Engineering, Stockholm, 13-19.
102. İnternet: Image Caption Generation with Visual Attention. URL: <https://medium.com/swlh/image-caption-generation-with-visual-attention-c782dfc0634b>. Son Erişim Tarihi: 23.12.2020.
103. Bahdanau, D., Cho, K., and Bengio, Y. (2015). *Neural Machine Translation by Jointly Learning to Align and Translate*. 3rd International Conference on Learning Representations, San Diego, 8-9.
104. İnternet: Attention Mechanism. URL: blog.floydhub.com/attention-mechanism. Son Erişim Tarihi: 21.12.2020.
105. Luong, M. T., Pham, H., and Manning, C. D. (2015). *Effective approaches to attention-based neural machine translation*. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, 1412-1421.
106. Takase, T., Oyama, S., and Kurihara, M. (2018). Effective neural network training with adaptive learning rate based on training loss. *Neural Networks*, 101, 68-78.
107. İnternet: A Gentle Introduction to Mini-Batch Gradient Descent and How to Configure Batch Size. URL: machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size. Son Erişim Tarihi: 22.12.2020.
108. Alvarez, J. M., and Salzmann, M. (2016). *Learning the Number of Neurons in Deep Networks*. Advances in Neural Information Processing Systems, Barcelona, 2270-2278.
109. Ruder, S. (2016). *An overview of gradient descent optimization algorithms*. International Conference on Smart Applications and Data Analysis, Agadir, Morocco, 349-359.
110. Glorot, X., and Bengio, Y. (2010, March). *Understanding the Difficulty of Training Deep Feedforward Neural Networks*. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 249-256.
111. Wang, S., Wang, X., Zhao, P., Wen, W., Kaeli, D., Chin, P., and Lin, X. (2018, November). *Defensive Dropout For Hardening Deep Neural Networks Under Adversarial Attacks*. Proceedings of the International Conference on Computer-Aided Design, San Diego, 1-8.

112. Kohavi, R., and Wolpert, D. H. (1996, June). *Bias Plus Variance Decomposition for Zero-One Loss Functions*. International Conference on Machine Learning, Bari, Italy, 275-83.
113. Lee, I., Kim, D., Kang, S., and Lee, S. (2017). *Ensemble Deep Learning for Skeleton-based Action Recognition Using Temporal Sliding LSTM Networks*. Proceedings of the IEEE International Conference on Computer Vision, Venice, 1012-1020.
114. ABBAS, S. S. I. (2019). *Automatic Text Categorization of Turkish News with Machine Learning and Deep Learning Techniques*, Doktora Tezi, Yıldırım Beyazıt Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 30-40.
115. İnternet: [Deeplearningturkiye. URL: github.com/deeplearningturkiye/kelime_kok_ayirici](https://github.com/deeplearningturkiye/kelime_kok_ayirici). Son Erişim Tarihi: 26.12.2020.
116. İnternet: [Akoksal/Turkish-Word2Vec. URL: github.com/akoksal/Turkish-Word2Vec](https://github.com/akoksal/Turkish-Word2Vec). Son Erişim Tarihi: 26.12.2020.
117. İnternet: Gensim. URL: pypi.org/project/gensim. Son Erişim Tarihi: 26.12.2020.
118. Karakoç, E. (2019). *Derin Öğrenme Yöntemleri Kullanılarak Türkçe Haber Metinlerinden Haber Başlığı Üretilmesi*, Yüksek Lisans Tezi, Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü, Kocaeli, 34-35.
119. İnternet: 9.8. Beam Search Colab [Mxnet] Open the Notebook in Colab Colab [Pytorch] Open the Notebook in Colab Colab [Tensorflow] Open the Notebook in Colab. URL: d2l.ai/chapter_recurrent-modern/beam-search.html. Son Erişim Tarihi: 27.12.2020.
120. İnternet: Stack Overflow Company Page - Learn about Stack Overflow. URL: stackoverflow.com/company. Son Erişim Tarihi: 27.12.2020.
121. Yılmaz, B. D., Demir, A. E., Sönmez, E. B., ve Yıldız, T. (2019) *Image Captioning in Turkish Language*. Innovations in Intelligent Systems and Applications Conference , İzmir, 1-5.



GAZİ GELECEKTİR..